

MIRZO ULUG'BEK NOMIDAGI  
O'ZBEKISTON MILLIY UNIVERSITETI

---



M.X. HAKIMOV

**AXBOROT TIZIMLARINI  
LOYIHALASH  
(TAMOYIL VA AMALIYOT)**

**O'ZBEKISTON RESPUBLIKASI  
OLIV TA'LIM, FAN VA INNOVATSIYA VAZIRLIGI**

**MIRZO ULUG'BEK NOMIDAGI  
O'ZBEKISTON MILLIY UNIVERSITETI**

**M.X. HAKIMOV**

**AXBOROT TIZIMLARINI  
LOYIHALASH  
(TAMOYIL VA AMALIYOT)**

**DARSLIK**

**Toshkent  
"Ma'rifat"  
2024**

**RLIK  
A  
1**

UO\*K: 004.415.25(075.8)

KBK: 32.973-018ya73

H 20

Hakimov M.X. Axborot tizimlarini loyihalash (tamoyil va amaliyot).  
Darslik. – T.: “Ma’rifat”, 2024. 424 bet.

Zamonaviy axborot tizimlarini yaratish o'ta murakkab masala bo'lib, uning yechimi birinchi navbatda loyihalashdan boshlanadi. Ushbu darslikda loyihalashning asosiy tamoyillari va foydalanishga asoslangan axborot tizimlarini modellashtirish uslubiyati yagonalashtirilgan UML tili shaklidagi ochiq standart asosida keltirilgan. Asosiy g'oyalar va usullarning mohiyatini tushunishga yordam beradigan misollar modellashtirish orqali keltirilgan. Asosiy tushunchalar 33ta jadval va 175ta rasmlar bilan to'ldirilgan. Darslikda g'oyaviy tamoyillar va amaliy ko'nikmalar asboblari ketma-ket kiritilgan, asosiy ob'yektlarni modellashtirish turli turdagi diagrammalar va rasmlarda keltirilgan. Amaliyot jarayonida magistrant axborot tizimini loyihalash va dasturiy muhitni o'zlashtirishda ob'yektga yo'naltirilgan uslubiyatga asoslangan CASE vositasi ArgoUMLdan foydalanadi.

UO\*K: 004.415.25(075.8)

KBK: 32.973-018ya73

H 20

**Mas'ul muharrir:**

R.D. Aloyev – Mirzo Ulug'bek nomidagi O'zbekiston Milliy universiteti  
“Hisoblash matematikasi va axborot tizimlari” kafedrası professori, fizika-  
matematika fanlari doktori

**Taqrizchilar:**

H.N. Zayniddinov – Muhammad al-Xorazmiy nomidagi TATU “Syn'iy  
intellect” kafedrası mudiri, professor, texnika fanlari doktori

A.B. Qobulov – Mirzo Ulug'bek nomidagi O'zMU “Axborot xavfsizligi”  
kafedrası professori, texnika fanlari doktori, Xalqaro muxandislar  
akademiyasining Akademigi

Mirzo Ulug'bek nomidagi O'zbekiston Milliy universiteti Kengashining  
2024-yil 28- iyundagi 9- sonli qarori bilan nashr etishga tavsiya etilgan.

ISBN: 978-9910-665-46-2

© Hakimov M.X., 2024

## MUNDARIJA

Kirish .....	10
<b>I- Bob. Tizimlar xususidagi mulohazalar.</b> .....	14
1.1. Tizimning ta'rifi. ....	15
1.2. Tizimning boshqa ta'riflari. ....	17
1.3. Tizimlar turlarini tanishni o'rganish. ....	18
1.4. Tizimlar, mahsulotlar va asboblarni chegaralash. ....	19
1.5. Pretsedent va pretsedent bo'lmagan tizimlar. ....	20
1.6. Tizimning analitik taqdimoti. ....	21
1.7. Mutaxassislikni talab qiladigan tizimlar. ....	22
1.8. Asosiy atamalarni aniqlash. ....	24
1.9. Xulosa. ....	25
1.10. Umumiy mashqlar. ....	26
1.11. Nazorat savollari. ....	27
<b>II-Bob. Tizimning xususiyatlari, xossalari va tavsiflari.</b> ...	28
2.1. Asosiy atamalarning ta'rifi. ....	28
2.2. Xususiyatlar, xossalari va tavsiflarning umumiy tushunchalari. ....	30
2.3. Har bir tizimdagi noyob aniqlovchi ....	31
2.4. Tizim samaradorligini tushunish. ....	41
2.5. Tizimning xususiyatlari. ....	43
2.6. Majburiy shartlar. ....	45
2.7. Xulosa. ....	48
2.8. Umumiy mashqlar. ....	48
2.9. Nazorat savollari. ....	48
<b>III- Bob. Tizimdagi rollar va manfaatdor tomonlar.</b> .....	50
3.1. Asosiy atamalarning ta'riflari. ....	50
3.2. Tashkilot tizimidagi rollar. ....	51
3.3. Tizimdagi rolning mazmuni. ....	53
3.4. Tizimdagi rollarning tizim muhandisligi bilan bog'liqligi. ....	55
3.5. Tizimdan manfaatkor shaxslarning roli nimadan iborat? ..	56
3.6. Tizimdagi manfaatdor tomonlarning rollari. ....	56
3.7. Xulosa .....	56

DENOV. TADBIRKORLIK  
VA PEDAGOGIKA  
INSTITUTI ARM  
№ 33835

3.8. Umumiy mashqlar.....	63
3.9. Nazorat savollari.....	63
<b>IV- Bob. Tizim amallarining modeli.....</b>	<b>65</b>
4.1. Asosiy atamalarning ta'riflari.....	65
4.2. Amaliyotlar tizimi konsepsiyasi.....	66
4.3. Tizim amallarining modeli.....	67
4.4. Tizim amallari modelining tavsifi.....	68
4.5. Muhimligi umumiydashgan tizim amaliyotlari modeli.....	77
4.6. Axborot tizimlari va sinflari.....	77
4.7. Tizim amallari lug'ati.....	80
4.8. Xulosa.....	81
4.9. Nazorat savollari.....	81
<b>V- Bob. Yuqori darajadagi talablar.....</b>	<b>83</b>
5.1. Talablarni aniqlash.....	83
5.2. Talablarni aniqlash loyihalash emas.....	84
5.3. Talablardagi muammolar.....	86
5.4. Ayon va yashirin funksional talablar.....	88
5.5. Funksional bo'lmagan talablar.....	88
5.6. Funksional bo'lmagan talablarning doimiyligi va vaqtinchaligi.....	90
5.7. Majburiy va istalgan talablar.....	91
5.8. Qo'shimcha talablar.....	92
5.9. Nazorat uchun savollar.....	99
<b>VI- Bob. Avtomatizatsiyalashtirilgan tizimlarni ishlab chiqish bosqichlari.....</b>	<b>101</b>
6.1. Asosiy jarayonlar.....	101
6.2. Yordamchi jarayonlar.....	103
6.3. Tashkiliy jarayonlar.....	105
6.4. ATni yaratish bosqichlari.....	105
6.5. Homaki loyiha.....	106
6.6. Texnik loyiha.....	106
6.7. Ish hujjatlari.....	107
6.8. Foydalanishga topshirish.....	108

6.9. Kuzatish.....	109
6.10. Nazorat uchun savollar.....	109
<b>VII- Bob. Dasturiy ta'minot hayot siklining modellari.....</b>	<b>110</b>
7.1. Pog'onali model.....	110
7.2. Pog'onali modelning afzalliklari va kamchiliklari.....	111
7.3. Pog'onali-qaytishli yondashuv.....	112
7.4. Qoplovchi jarayonli pog'onali yondashuv.....	113
7.5. Inkrement (orttirma) model.....	114
7.6. Aylanma (spiral) model.....	116
7.7. Tortishuv modeli.....	119
7.8. RAD uslubiyati.....	120
7.9. XP uslubiyati.....	121
7.10. Nazorat uchun savollar.....	123
<b>VIII- Bob. Axborot tizimlarini yaratishni tashkillashtirish.....</b>	<b>124</b>
8.1. Axborot tizimlarini loyihalash.....	124
8.2. ATni kanonik loyihalash.....	125
8.3. Tashkilot tuzilmasini tekshirish va strategiyani aniqlash.....	127
8.4. Tashkilot faoliyatining tahlili.....	129
8.5. Tizim funksiyalarini sinflash.....	130
8.6. Nazorat ugun savollar.....	132
<b>IX- Bob. Texnik topshiriqning tuzilishi.....</b>	<b>134</b>
9.1. Umumiy qoidalar.....	134
9.2. Texnik topshiriq tarkibi va mazmuni.....	135
9.3. Tizimga qo'yiladigan umumiy talablar.....	137
9.4. TTda ATni loyihalashdagi ta'minot turlariga talablar.....	141
9.5. TTda ATni yaratish bo'yicha ishlarning tarkibi va mazmuni.....	144
9.6. Nazorat ugun savollar.....	146
<b>X- Bob. Axborot ta'minotini modellashtirish.....</b>	<b>147</b>
10.1. Berilganlarni modellashtirish.....	147
10.2. «Mohiyat-aloqa» modeli.....	148
10.3. Binar aloqalarning uch turi.....	153
10.4. Kalitlar.....	156

10.5. Berilganlarni me'yorlashtirish .....	156
10.6. Domenlar .....	157
10.7. Zaif va kuchli mohiyatlar .....	158
10.8. Bir va ko'p qiymatli atributlar .....	161
10.9. Mohiyatlarning ostki va ustki turlari .....	161
10.10. «Mohiyat-aloqa» diagrammalari .....	163
10.11. Triggerlar va saqlangan protseduralar .....	164
10.12. Axborot modeli .....	166
10.13. Predmet sohani modellashtirish darajalari .....	169
10.14. Xulosa .....	172
10.15. Nazorat uchun savollar .....	174
<b>XI- Bob. UML – yagonalashgan modellashtirish tili</b> .....	<b>177</b>
11.1. UML – bu modellashtirish tili .....	177
11.2. Model va uning elementlari .....	180
11.3. Munosabatlar .....	182
11.4. Diagrammalar .....	187
11.5. Umumiy diagrammalar .....	191
11.6. Maxsus diagrammalar .....	200
11.7. Xulosa .....	204
11.8. Nazorat sabollari .....	205
<b>XII- Bob. Tizim tuzilmasini UMLda modellashtirish.</b> .....	<b>206</b>
12.1. Tuzilmani ob'yektga-yo'naltirilgan modellashtirish..	206
12.2. Sinflovchilar va ularning eng muhim xususiyatlari. ...	209
12.3. Nusxaning harakatlanish sohasi va statik faoliyat doirasi. ....	216
12.4. Sinf diagrammasidagi mohiyatlar .....	218
12.5. Sinf diagrammasidagi munosabatlar .....	226
12.6. Xulosa .....	230
12.7. Nazorat sabollari .....	231
<b>XIII- Bob. Uyushmalar.</b> .....	<b>232</b>
13.1. Uyushmalar va ularning qo'shimchalari .....	232
13.2. Uyushma nomi. Uyushma qutbining karraligi. ....	233
13.3. Yig'ish va kompozitsiya .....	234

13.4. Uyushma sinfi va malakachi .....	240
13.5. Amalga oshirish diagrammalari .....	243
13.6. Sinflovchilardagi rollar va nusxalar darajasida modellashtirish .....	249
13.7. Hamkorlik .....	253
13.8. Loyihalash namunasi. ....	255
13.9. Sinflovchilarning nusxalari .....	256
13.10. Ob'yektlar va ob'yektlar diagrammasi .....	258
13.11. Xulosa .....	258
13.12. Nazorat sabollari .....	259
<b>XIV- Bob. IDEF tarkibiy tahlil uslubiyatlari</b> .....	<b>261</b>
14.1. IDEF nima? .....	261
14.2. IDEF0 uslubiyatining asoslari .....	263
14.3. Avlodlar diagrammasi .....	266
14.4. Ajdodlar diagrammasi .....	267
14.5. Diagrammalarni qurish qoidalari. ....	268
14.6. IDEF0 uslubiyati va tilining asosiy tushunchalari ...	270
14.7. IDEF1X uslubiyati .....	275
14.8. Nazorat savollari .....	284

## AMALIY QISM

<b>XV- Bob. ArgoUML muhiti</b> .....	<b>285</b>
15.1. ArgoUML xususuda .....	285
15.2. Ob'yektga yo'naltirilgan tahlil va loyihalash .....	286
15.3. ArgoUMLning rivojlanishi .....	287
15.4. ArgoUMLdan foydalanish .....	288
15.5. Nima uchun ArgoUML boshqacha .....	289
15.6. ArgoUMLda amaliy dastur .....	290
15.7. ArgoUML ochiq manba .....	291
15.8. Loyihalar bilan ishlash .....	291
15.10. Talablarni jamlash .....	292
15.11. Jarayon bosqichlari .....	294

15.12. Namoyishli hujjat .....	295
<b>XVI- Bob. Foydalanish holatlari diagrammasi</b> .....	297
16.1. Nazariy asoslar .....	297
16.2. Foydalanish holatlari va aktyorlar o'rasidagi aloqalar .....	303
16.3. Foydalanish holatlari diagrammasidagi asboblar .....	305
16.4. "Pensiya ishi" ATning foydalanish holatlari diagrammasi .....	310
<b>XVII- Bob. Sinf diagrammasi</b> .....	313
17.1. Nazariy asoslar .....	313
17.2. Sinf diagrammasidagi asboblar .....	322
17.3. Sinf diagrammasini qurish .....	325
17.4. "Pensiya ishi" ATning sinf diagrammasi .....	327
<b>XVIII- Bob. Ketma-ketliklar diagrammasi</b> .....	329
18.1. O'zaro harakat diagrammalari .....	329
18.2. Nazariy asoslar .....	329
18.3. Ketma-ketlik diagrammasining asboblari .....	332
18.4. "Pensiya ishi" ATning ketma-ketliklar diagrammasi .....	333
<b>XIX- Bob. Hamkorlik (aloqa) diagrammasi</b> .....	336
19.1. Nazariy asoslar .....	336
19.2. Hamkorlik diagrammasini qurish .....	337
19.3. "Pensiya ishi" ATning hamkorlik diagrammasi .....	340
<b>XX- Bob. Holat (avtomat) diagrammasi</b> .....	342
20.1. Nazariy asoslar .....	342
20.2. Holat diagrammasini qurish asboblari .....	347
20.3. "Pensiya ishi" ATning holat diagrammasi .....	356
<b>XXI- Bob. Faoliyat diagrammasi</b> .....	358
21.1. Nazariy asoslar .....	358
21.2. Faoliyat diagrammasining asboblari .....	360
21.3. "Pensiya ishi" ATning faoliyat diagrammasi .....	366
<b>XXII- Bob. Joylashtirish diagrammasi</b> .....	369
22.1. Nazariy asoslar .....	370
22.2. Joylashtirish diagrammasining asboblari .....	370
22.3. AKM tizimining joylashtirish diagrammasi .....	373

22.4. "Pensiya ishi" ATning joylashtirish diagrammasi .....	374
<b>XXIII- Bob. Tarkib diagrammasi</b> .....	376
23.1. Nazariy asoslar .....	376
23.2. AKM tizimi uchun tarkib diagrammasi .....	376
<b>XXIV- Bob. Ob'yekt diagrammasi</b> .....	379
24.1. Ob'yekt .....	379
24.2. Nusxa tasnifi .....	380
24.3. Uya .....	382
24.4. Nusxa qiymati .....	384
<b>XXV- Bob. Paket diagrammasi</b> .....	386
25.1. Paket .....	386
25.2. Paketlardagi tobelik .....	389
25.3. Paketni import qilish .....	390
25.4. Element importi .....	392
25.5. Paketlarni birlashtirish .....	395
25.6. Paketlarni birlashtirishning umumiy qoidalari .....	400
25.7. Paketlar uchun qoidalar .....	402
25.8. Sinflar va berilganlar turlari uchun qoidalar .....	402
25.9. Xossalar uchun qoidalar .....	403
25.10. Uyushmalar uchun qoidalar .....	404
25.11. Amallar uchun qoidalar .....	405
25.12. Sanoq turi qoidalari .....	405
25.13. Cheklovlar uchun qoidalar .....	406
<b>Glossariy</b> .....	408
<b>Foydalanilgan adabiyotlar</b> .....	423

## KIRISH

Axborot tizimlari axborot texnologiyalarini amalga oshirish vositasi sifatida ushbu texnologiyalar bilan birgalikda rivojlanmoqda. Ijtimoiy-iqtisodiy xarakterdagi vazifalar birinchi kompyuterlar paydo bo'lgan davrdan boshlab kompyuter texnologiyalari yordamida hal qilindi, lekin ular yetakchi dasturlar emas edi, chunki uzoq vaqt davomida ilmiy va texnikaviy hisobotlarga mo'ljallangan dasturiy mahsulotlar ustunlik qildi. Hozirgi vaqtda vaziyat keskin o'zgargan – mavjud kompyuterlarning aksariyati axborot texnologiyalarini amalga oshirib, kundalik hayotda, o'qishda va eng muhimi, ishlab chiqarish va tijoratda muvaffaqiyatli qulayliklarni ta'minlamoqda.

Yaqin o'tmishda axborot tizimlari va texnologiyalaridan foydalanish tijoratda raqobatdosh ustunlikka erishish uchun zarur shart sifatida qaraldi. Aynan mana shu afsonaviy afzalliklar tijoratni qo'llab-quvvatlashga qaratilgan ko'plab ilmiy tadqiqotlar va amaliy dasturiy ishlanmalarni rag'batlantirdi.

Ham yirik, ham kichik kompaniyalarning axborot tizimlari va texnologiyalariga investitsiyalarni baholash qiyin. Bu sohadagi har qanday faoliyatni, shu jumladan, tijorat jarayonini boshqarish g'oyasi paydo bo'ldi. Ushbu g'oya tijorat jarayonlarini qayta qurish g'oyasining asosini tashkil etdi. Qayta muhandislik yaratuvchilari Tom Davenport, Maykl Hammer va Jeyms Champi - "deyarli har qanday korxonada tijorat jarayonlarini tashkil etish har qanday yopiq tizimda entropiyaning (tartibsizlik) muqarrar ravishda ko'payishi tufayli maqbul emas" degan aniq faktdan kelib chiqdilar. Shuning uchun axborot tizimlari va texnologiyalarini amalga oshirish bosqichidan oldin tashkilotda ishlab chiqarish va tashkiliy munosabatlarni qayta qurish bosqichi bo'lishi kerak. Qayta muhandislik va jarayonni boshqarishni qo'llab-quvvatlash vositalarini yaratish va amalga oshirish jarayonlarida, deyarli barcha axborot tizimlarining yechimlarini ishlab chiquvchilar va yetkazib beruvchilar ishtirok etadilar. Tijorat va

ishlab chiqarishning ko'plab sohalarida ko'plab standart vazifalarning mavjudligi axborot tizimlarini loyihalash va ularning dasturiy ta'minotini ishlab chiqish uchun sanoat uslubiyatlarini yaratishga olib keldi, masalan, namunaviy loyihalash yechimlarini.

Yangi vazifalar va rollarni murakkablik bilan tushunish uchun zarur bo'lgan ko'nikmalar va mas'uliyat va tajribani yillar davomida chiniqtirishni talab qiladi. Tizim bo'yicha mutaxassis bo'lish jiddiy qabul qilinadigan va rasmiy ta'lim beradigan oliy ta'lim tomonidan olib boriladi. Ma'lum muddatli tajribadan so'ng, tashkiliy va shartnomani bajarish talablarini shakllantirish uchun mutaxassislardan boy bilim va tajribani o'zlashtirish va tizimlar qanday ishlashi haqida fikrlarni sintez qilishni talab qiladi. Ushbu talablarning asosiy elementi mijozlar bilan aloqa qilishdir. Aloqa mijozdan tahliliy so'rovlar, kuzatuvlari va ko'nikmalarini tinglash, amaliy ehtiyojlari va ishonchsiz umidsizlini tushunish kabi mahsulotlar orqali amalga oshirilishi kerak.

Foydalanuvchilar o'zlarining qarashlarini yangi, yuqori darajada ishlash va ishonchli texnologiyalarni qo'llashni talab qiladigan yangi turdagi tizimlarga amaliy ehtiyojlar orqali bildiradilar va mutaxassisga tizimni innovatsion loyihalash va yaratish imkoniyatini beradilar. Yaratiladigan har bir tizim, puxta loyihaga ega bo'lsa, u noyob dasturiy mahsulotga aylanadi. Lekin bunday yondashuv yillar talab qilishi mumkinligini unutmazlik kerak.

Axborot tizimlari uchun dasturiy ta'minotni ishlab chiqish sohasi so'nggi o'ttiz yil atrofida rivojlandi yagonalashgan model-lashtirish tili (UML) va Birlashgan jarayon (UP) sifatida qabul qilingan ob'yektlarni boshqarish guruhi (OMG) tomonidan xalqaro standartlar yaratildi. Hatto tezkor rivojlanayotgan usullarda ham hozirgi kunda ob'yektga yo'naltirilgan rivojlanishni yetakchi namuna sifatida foydalanish taklif etilmoqda.

Ob'yektga yo'naltirilgan dasturlash 40 yildan ortiq mavjud bo'lsa-da, texnik xizmat ko'rsatishning yetishmasligi, loyihalar-ni mukammal emasligi va ular ishlab chiqaradigan tizimlarda moslashuvchanlik mavjud emasligi tufayli, hali ham sifatli

dasturiy ta'minotni ishlab chiqarish qiyin bo'lgan davlat va xususiy tashkilotlar mavjud. Ushbu darslikda ushbu kamchilikni minimallashtirish uchun tamoyillar va amaliyotlar keltirilgan.

Dasturiy ta'minot loyihasining boshlang'ich davrida narsalar ko'pincha loyqa bo'ladi, chunki ba'zida manfaatdor tomonlarni kelishib olishi qiyin, chunki talab qilinadigan ma'lumotlar va funksiyalar ko'pincha noaniq, chunki mijozlar qo'rqinchli murakkablikdagi tizimlarni talab qilishadi, hech qanday majburiyatga ega emaslar va biz nima talab qilinishini avvaldan bilamiz, deb juda jozibali taxmin qilishadi.

So'nggi 30 yil ichida biz kompyuter dasturlarini tahlil qilish va loyihalash uchun taklif qilingan ko'plab usullarni ko'rdik, ammo bularning orasidagi, eng tashabbuskor va samaralisi ob'yektga yo'naltirilgan fikrlashni qabul qilishdir.

Tahlil qilish va loyihalash bir vaqtning o'zida dasturiy ta'minot jarayoniga nisbatan erta sodir bo'lsa, ob'yektlar ko'pincha muammoning barcha elementlarini aniq o'z ichiga oladi.

Ob'yektga yo'naltirilgan namunaning muhimligi shundaki, u bizga aniqlash, ishlab chiqish va vakillik bo'yicha muammoni bir jarayon orqali tushunish uchun zarur bo'lgan vositalarni taqdim etadi. Loyihada foydalanish holatlarini ta'kidlaydi va buni tushunishga yordam beradi.

Yaxshi tahlilchi va loyihachilar tobora rivojlanib borish bilan, muammoni yaxshiroq va batafsilroq tushunish imkoniyati bo'ladi, deb ta'kidlamoqdalar. Buni amalga oshirish uchun ob'yektga yo'naltirilgan modellashtirish va bu ishni bajarish uchun UML vositalaridan foydalanishni taklif qilmoqdalar.

Tahlil qilish va loyihalashning samarali usullari, dasturiy ta'minot mutaxassislariga boradigan joyni va u yerga borish vositasini beradi.

Shu sohadagi ba'zi boshqa chet el kitoblaridan farqli o'laroq, UML diagrammalari va ulardan keng foydalanish mumkin bo'lgan, axborot tizimlarining tahlili va loyihalash faoliyatida turli

xildagi diagrammalar ularning yaxshiroq rivojlanishiga qanday yordam berishi mumkinligini ham tushuntiradi.

Ushbu kitob orqali barcha tahlillarga misollar keltirish uchun davomiy amaliy tadqiqotlar usuli va loyihalash texnikasi qo'llaniladi. Asosiy tamoyillar joriy etiladi, ishlaydigan vositalar to'plami kengaytiriladi va UML yordamida tahlil va loyihalash modellarini ishlab chiqish uchun foydalaniladi.

Darslikda tahlilchilar va loyihachilar tomonidan amalga oshiriladigan tadbirlarga ko'proq e'tibor berilgan, shu jumladan tijoratni modellashtirish, talablarni yaratish, tahlil va loyihalash masalalari bo'yicha ko'nikmalar hosil qilishga yordam beradi.

Ushbu darslikning maqsadi axborot tizimlarini loyihalash tushunchalari, uslubiyatlari va texnologiyalarini tizimli ravishda taqdim etishdir, bu talabalarga axborot tizimlari va texnologiyalariga loyihalash bilan bog'liq barcha muammolar bilan tanishish, shuningdek uslubiyatlarning rivojlanish va shakllanish tarixini kanonik loyihalashdan sanoatga, funksional yo'naltirilgan loyihadan ob'yektga yo'naltirilgan va undan keyin jarayonga yo'naltirishdan iborat.

Darslikdan 70610101 – Kompyuter ilmlari va dasturlash texnologiyalari (yo'nalislar bo'yicha), 70610201 – Kompyuter tizimlari va ularning dasturiy ta'minoti (tarmoqlar va sohalar bo'yicha), 70610204 – Axborot tizimlari (tarmoqlar bo'yicha) mutaxassisliklari magistrantlari va talabalari, doktorantlar, tadqiqotchi izlanuvchilar va soha mutaxassislari foydalabishlari mumkin.

Ushbu kitob, oliy o'quv yurtlari magistrantlariga mo'ljallangan axborot tizimlarini loyihalash sohasidagi birinchi darslik bo'lib, uni yozishda Mirzo Ulug'bek nomidagi O'zbekiston Milliy universitetida muallif tomonidan magistrantlarga "Axborot tizimlarini loyihalash" fanini o'qitishdagi ko'p yillik tajribasi hisobga olingan.

## I- BOB. TIZIMLAR XUSUSIDAGI MULOHAZALAR

Tahlil qilish, loyihalash va ishlab chiqish tizimlari, mahsulotlar yoki xizmatlar bir nechta asosiy savollarga javob berishni talab qiladi:

1. Tizim degani nima?
2. Tizim chegaralariga nimalar kiradi?
3. Foydalanuvchi tashkiloti ichida tizim qanday rolni bajaradi?
4. Tizim qanday topshiriq ilovalarini bajaradi?
5. Tizim qanday natijalarga yo'naltirilgan natijalarni beradi?

Ushbu asosiy savollarga javob berish ko'pincha qiyin. Agar siz tizim *nima* ekanligini aniq va qisqacha tasvirlay olmasangiz, sizda katta qiyinchilik mavjud. Endi bir xil muammo ustida ishlaydigan odamlar guruhlarini yaqinlashuv va javoblar bo'yicha konsensusga olib kelishda murakkablik elementini qo'shing. Bu foydalanuvchilar, ekvayerlar va tizimni ishlab chiquvchilar bo'lib, hatto ularning o'z tashkilotlarida ham bu umumiy muammo mavjud. *Ekvayer* – bu savdo-xizmat ko'rsatish tarmog'idagi terminallar va bankomatlardan tashkil topgan kartalarga xizmat ko'rsatish punktlari bilan o'zaro aloqada bo'lgan barcha amallarni amalga oshiradigan bank yoki kompaniya. Tarmoqdagi amallar to'g'risida ma'lumot olgandan so'ng, ekvayer ularni hisob-kitoblarni amalga oshirish uchun tizimga yuboradi. Ekvayer xaridlar amalga oshirilgan yoki xizmatlar kartalar orqali to'langan savdo nuqtalariga mablag'larni qaytarish uchun javobgardir.

Biz tizim nima ekanligini aniqlashdan boshlaymiz va ta'rif doirasidagi tuzulmaviy iboralarning ma'nosini tushuntiramiz. Ta'rifga asoslanib, tizimlarning turli toifalarini kiritamiz va tizimlar, mahsulotlar va asboblar o'rasidagi farqlarni tavsiflaymiz. Biz pretsedent va pretsedent bo'lmagan tizimlar tushunchalari, ular orasi-

dagi farqni o'rganamiz hamda tizimning analitik va grafik tasvirini taqdim etish bilan yakunlaymiz.

### 1.1. Tizimning ta'rifi

"Tizim" atamasi yunoncha "systema" so'zidan kelib chiqqan bo'lib, "birga joylashtirish" degan ma'noni anglatadi. Bir nechta iqtisod, texnika, tibbiyot, tijorat va ishlab chiqarish sohalarida tizimning turlicha ta'riflari mavjud. Lekin Charles S. Wasson tomonidan berilgan tizimning quyidagi ta'rifi, uni mukammal tarzda har tomonlana chuqur ma'noda ifodalaydi:

➤ *Tizim* – bu har biri aniq belgilangan va cheklangan imkoniyatlarga ega bo'lgan o'zaro ishlaydigan elementlarning integratsiyalashgan termasi bo'lib, foydalanuvchiga ma'lum natija va muvaffaqiyat ehtimoli bilan belgilangan amaliy muhitda missiyaga yo'naltirilgan tezkor ehtiyojlarni qondirishga imkon berish uchun qo'shimcha qiymatni qayta ishlashni amalga oshirish uchun sinergik tarzda ishlaydi.

Ushbu ta'rifning mantiqiyligini tushunish uchun, har bir qismni batafsil ko'rib chiqamiz.

Keltirilgan ta'rif tizimlar haqida bir qator asosiy muhokama nuqtalarini qamrab oladi. Ta'rifdagi har bir ibora uchun asosni ko'rib chiqamiz.

"Integratsiyalashgan terma" deganda, tizim ta'rifga ko'ra, jismoniy elementlar, ob'yektlar yoki tarkiblarning shajaraviy darajalaridan tashkil topganligini tushunamiz.

"O'zaro faoliyat yurituvchi elementlar" deganda, masalan, tizim tuzilmasidagi elementlar shakli, mosligi va funksiyasi bo'yicha bir-biriga mos bo'lishi kerakligini tushunamiz. Tizim elementlari uskunalar (masalan, asbob va dasturiy ta'minot, xodimlar, ob'yektlar, amaliy cheklovlar, qo'llab-quvvatlash), texnik xizmat ko'rsatish, ta'minot, ehtiyot qismlar, o'qitish, zaxiralar, protsessual berilganlar, tashqi tizimlar va missiyaga erishishni qo'llab-quvvatlaydigan boshqa har qanday narsalarni o'z ichiga

oladi. Umuman olganda, tizim elementlari bir-birini to'ldirmaydigan vazifalar va maqsadlarga ega bo'lishi kerak. Biroq, tizimlardagi ortiqchalik bir nechta tizim elementlari bo'yicha imkoniyatlarni takrorlashni talab qilishi mumkin. Bundan tashqari, ba'zi tizimlar, masalan, tarmoqlar, bir xil tarkiblarning bir nechta nusxalariga ega bo'lishi mumkin.

Har bir element "*aniq belgilangan va chegaralangan imkoniyatlarga ega*" deganda, biz har bir element qandaydir yuqori darajadagi maqsad yoki maqsadli vazifani bajarish uchun ishlashi kerakligini nazarda tutamiz. Tizim elementining umumiy tizim ishlashiga qo'shgan hissasi aniq ko'rsatilishi kerak. Bu har bir tizim elementi uchun amaliy va funksional ishlash imkoniyatlarini aniqlashni va elementni tahlil qilish, loyihalash, ishlab chiqish, sinovdan o'tkazish, tekshirish va tasdiqlash imkonini beruvchi o'ziga xoslik darajasiga aniq chegaralanishini talab qiladi. Tizim elementi mustaqil ravishda yoki integratsiyalashgan tizimning bir qismidir.

"*Sinergik tarzda ishlash*" deganda, biz elementlar termasini integratsiya qilishdan maqsad, alohida elementlar sifatida erishib bo'lmaydigan yuqori darajadagi qobiliyatni amalga oshirish uchun, alohida elementlarning imkoniyatlaridan foydalanish ekanligini tushunamiz.

"*Qo'shimcha qiymatli qayta ishlash*" deganda, biz amaliy xarajat, foydalilik, yaroqlilik, mavjudlik va samaradorlik kabi omillar, tizimdagi har bir amal va masaladan kiruvchi berilganlarning kirishiga qiymat qo'shishini va tizim umumiy vazifasiga erishishiga hissa qo'shadigan natijalarni ishlab chiqarishni talab qilishini tushunamiz.

"*Foydalanuvchiga missiyaga yo'naltirilgan amaliy ehtiyojlarni bashoratli ravishda qondirish imkoniyatini berish*" deganda, biz har bir tizimning maqsadi (ya'ni, mavjudlik sababi) va foydalanuvchi(lar) uchun qiymati borligini nazarda tutamiz. Uning qiymati operatsion ehtiyojlarni qondirish yoki tizim vazifalari va maqsadlarini qondirishga nisbatan investitsiya daromadi, bo'lishi

mumkin.

"*Belgilangan ish muhitida*" deganda, biz iqtisodiy, natija va omon qolish sabablariga ko'ra har bir tizim belgilangan, ya'ni chegaralangan ish muhitiga ega bo'lishi kerakligini tushunamiz.

"*Muayyan natija bilan*" deganda, biz tizimdan manfaatdor tomonlar (foydalanuvchilar, aksiyadorlar, egalar va boshqalar) tizimlardan natijalar kutishlarini nazarda tutamiz. Kuzatilgan xatti-harakatlar, mahsulotlar, qo'shimcha mahsulotlar yoki xizmatlar, masalan, natijaga yo'naltirilgan, miqdoriy, o'lchanadigan va tekshirilishi mumkin bo'lishi kerak.

"*Muvaffaqiyat ehtimoli*" deganda, biz ma'lum bir natijaga erishish noaniqlik yoki xavf darajasini o'z ichiga olishini nazarda tutamiz. Muvaffaqiyat darajasi ishonchlilik, mavjudlik, turg'unlik, barqarorlik, halokatlilik va o'zgartirish imkoniyatining mavjudligi kabi turli xil ish omillari bilan belgilanadi.

Dasturiy mahsulotlar o'rtasida kadrlar almashinuvi sifatida shartnomalar va tashkilotlar bo'ylab uzluksizlik o'rnatiksa, tashkiliy jihatdan tizimni ishlab chiquvchi guruh a'zolari o'rtasida kelishuvga erishishga erishish oson bo'ladi.

## 1.2. Tizimning boshqa ta'riflari

Milliy va xalqaro standartlar tashkilotlari, shuningdek, turli mualliflar tizim bo'yicha, o'z ta'riflariga ega. Agar biz bularni tahlil qilsak, ularning shaxsiy bilimlari va tajribalari asosida turlicha qarashlarni topasiz. Bundan tashqari, standart tashkilotlari tomonidan "bir o'lcham hammaga mos keladi" konvergensiyasi va konsensusga erishish ko'pincha so'zlarning shunchalik sayozlashishiga olib keladiki, ko'pchilik buni etarli emas, deb hisoblaydi. Standart ta'riflarga ega bo'lgan tashkilotlarga misollar:

- Tizim muhandisligi bo'yicha xalqaro kengash (INCOSE);
- Elektr va elektron muhandislar instituti (IEEE);
- Amerika Milliy Standartlar Instituti (ANSI);
- Elektron Industries Alliance (EIA);



- Xalqaro standartlar tashkiloti (ISO);
- AQSh Milliy Aeronavtika va Koinot boshqarmasi (NASA);
- AQSh Federal Aviatsiya boshqarmasi (FAA).

**Izoh:** Bilimlaringizni kengaytirish maqsadida ushbu tashkilotlarning ta'riflarini o'rganish tavsiya etiladi.

Mutahassislar ta'riflarni ishlab chiqishganda, ular bir vaqtning o'zida tarkib va grammatikani yaratishga harakat qilishadi. Insonlar odatda grammatikaga nomutanosib vaqt sarflashadi va mazmunli tarkibga juda oz vaqt sarflashadi. Biz buni, masalan, tasniflar va rejalarda ko'ramiz. Grammatika muhim ahamiyatga ega, chunki u tilimiz va muloqotimizning ildizidir. Biroq, agar so'zli grammatikada mazmunli mazmun bo'lmasa, hech qanday ahamiyatga ega bo'lmaydi. Siz so'zlarni yozish mashqlari orqali odamlarning qanday jonli va g'ayratli bo'lishiga hayron qolamiz.

### 1.3. Tizimlar turlarini tanishni o'rganish

Tizimlar bir qancha shakllarda yuzaga keladi va tarkibi, shajaraviy tuzilishi va xatti-harakati bilan farqlanadi. Quyida, yuqori darajadagi tizimlarga misollar keltirilgan:

- |                        |                        |
|------------------------|------------------------|
| • Iqtisodiy            | • Ta'lim               |
| • Moliyaviy            | • Ekologik             |
| • Tibbiy               | • Korporativ           |
| • Sug'urta             | • Diniy                |
| • Ijtimoiy             | • Psixologik           |
| • Madaniy              | • Oziq-ovqat tarqatish |
| • Transport            | • Aloqa                |
| • Ko'ngilochar         | • Hukumat              |
| • Qonunchilik          | • Sud                  |
| • Daromad              | • Soliq                |
| • Litsenziyalash       | • Harbiy               |
| • Farovonlik           | • Jamoat xavfsizligi   |
| • Parklar va dam olish | • Atrof-muhit          |
| • Ekologik             | • Turistik             |

Agar biz ushbu tizimlarni tahlil qilsak, ular mahsulotlar, qo'shimcha mahsulotlar yoki xizmatlar kombinatsiyasini ishlab chiqarishini aniqlaymiz. Keyingi tahlillar shuni ko'rsatadiki, ularning aksariyati bir yoki bir nechta sinflarga kiradi, masalan, shaxsiy va tashkiliy; rasmiy va norasmiy; yerga asoslangan, dengizga asoslangan, havoga asoslangan, kosmik yoki gibrid; ochiq halqa va yopiq halqa; statsionar, mobil va ko'chma tizimlar.

### 1.4. Tizimlar, mahsulotlar va asboblarni chegaralash

Ko'pincha tizimlar, mahsulotlar va asboblarni tushunchalari chalkashtirib yuboriladi. Munozarani osonlashtirish uchun, ushbu atamalarning har birini batafsil ko'rib chiqamiz.

**Tizim konteksti.** Tizim ikki yoki undan ortiq integratsiyalashgan elementlardan iborat bo'lishi mumkin, ularning birlashtirilgan – sinergik maqsadi har bir element tomonidan shaxsiy asosda samarali yoki samarali bajarilmasligi mumkin bo'lgan missiya maqsadlariga erishishdir. Ushbu tizimlar odatda turli darajadagi insonlar, mahsulotlar va asboblarni o'z ichiga oladi. Umuman olganda, inson tomonidan yaratilgan tizimlar rejalashtirish, foydalanish, aralashuv yoki qo'llab-quvvatlash uchun ma'lum darajadagi inson zaxiralarini talab qiladi.

**Mahsulot konteksti.** Ba'zi tizimlar boshqa tizimlar tomonidan ishchi mahsulot sifatida yaratiladi. Keling, mahsulot kontekstini aniqlaymiz. Mahsulot, kattaroq tizimning muvofiq elementi sifatida, odatda, ma'lum bir ishlash darajasiga ega bo'lgan o'ziga xos qobiliyatga ega - shakl, moslik va funksiyaga ega bo'lgan jismoniy qurilma yoki ob'yektdir. Mahsulotlar odatda inson yordamisiz o'zini o'zi qo'llash qobiliyatiga, ya'ni aqlga ega emas. Bundan tashqari, mahsulotlar yuqori darajadagi tizim missiyasi maqsadlariga erisha olmaydi, u qandaydir shaklda inson aralashuviga muhtoj. Oddiy qilib aytganda, biz ko'pincha uskunaga

asoslangan mahsulotlarga tegishli bo'lamiz, chunki siz sotuvchidan katalog buyurtma raqami orqali xarid qilishingiz mumkin. Biroq, kontekstga ko'ra, mahsulot foydalanuvchining yuqori darajadagi tizimiga integratsiyalangan sotuvchining "tizimi" bo'lishi mumkin. Samarali ravishda biz tizimlar tizimini yaratamiz.

**1- misol.** Bolg'a, sotib olinadigan mahsulot sifatida, shakli, mosligi va funksiyasiga ega, ammo tirnoqlarni urish yoki uzish uchun uni qo'llash imkoniyati yoq.

**2- misol.** Reaktiv samolyot tizim va sotib olinadigan sotuvchi mahsulot sifatida aviakompaniya tizimiga birlashtirilgan va ma'lum sharoitlarda uchuvchi tomonidan dasturlashtirilgan va faollashtirilganda parvoz qilish qobiliyatiga ega bo'lishi mumkin.

**Asbob konteksti.** Ba'zi tizimlar yoki mahsulotlar yuqori darajadagi tizimlar tomonidan vosita sifatida ishlatiladi. Vosita deganda, biz nimani nazarda tutayotganimizni aniqlaymiz. Asbob foydalanuvchi yoki tizimning shaxsiy imkoniyatlaridan oshib ketadigan missiya maqsadlariga yanada samarali yoki samarali erishish uchun foydalanuvchi yoki tizimga o'z imkoniyatlari va unumdorligidan foydalanish imkonini beruvchi yordamchi mahsulotdir.

**3- misol.** Oddiy tayanch nuqtasi va burilish vositasi, vosita sifatida insonga o'z jismoniy kuchini ishlatishga imkon beradi. Aksinchasi, bir odam tomonidan osonlik bilan harakatlantirilmaydigan tosh.

**4- misol.** Statistik dasturiy ta'minot qo'llab-quvvatlash vositasi sifatida statistika bo'yicha mutaxassisga qisqa vaqt ichida katta hajmdagi berilganlar va farqlarni samarali tahlil qilish imkonini beradi.

### 1.5. Pretsedent va pretsedent bo'lmagan tizimlar

Inson tomonidan yaratilgan tizimlarning aksariyati vaqt o'tishi bilan rivojlanadi. Tizimning har bir yangi evolyutsiyasi

yangi yoki ilg'or texnologiyalar, usullar, vositalar, texnikalar va boshqalarni qo'llash orqali oldingi tizimning imkoniyatlarini kengaytiradi va o'zgartiradi. Shunday bo'lsa-da, tizim amaliy muhitlar yoki ehtiyojlar natijasida shu kabi yangi muammolarni keltirib chiqaradi. Biz ularni pretsedent va pretsedent bo'lmagan tizimlar, deb ataymiz. Garchi biz huquqiy tizim va uning pretsedentlari nuqtayi nazaridan o'ylashga moyil bo'lsak-da, jismoniy tizimlar, mahsulotlar va xizmatlarda ham pretsedentlar mavjud.

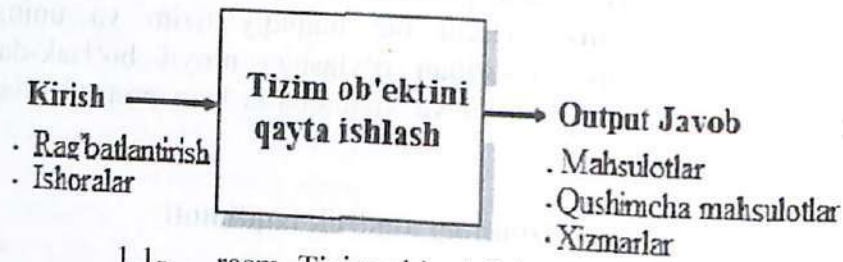
### 1.6. Tizimning analitik taqdimoti

Abstraktsiya sifatida biz 1- rasmda ko'rsatilganidek, to'rtbur-chakli quti yordamida tizimni oddiy ob'yekt sifatida ramziy ravishda ifodalaymiz. Umuman olganda, stimullar va signallar kabi kirishlar, kirishlarni qayta ishlaydigan va chiqishni ishlab chiqaradigan tizimlarga kiritiladi. Qurilish sifatida bu ramziylik qabul qilinadi ammo, so'zlar tizim qanday faoliyat yuritishini aniqlashi kerak. Ya'ni, tizim kerakli mahsulot ishlab chiqarishda kirishga qiymat qo'shiladi.

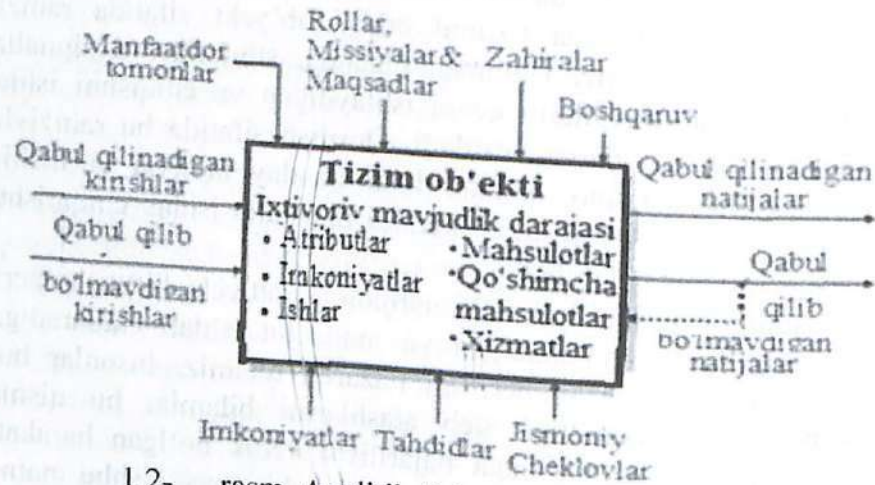
Bunda biz qobiliyat kabi natijani yaratuvchi kiruvchi berilganlarga qiymat qo'shadigan va mahsulot ishlab chiqaradigan transformatsion ishlov berishni nazarda tutamiz. Insonlar buni tizimning funkcionalligi, deb atashlarini bilamiz; bu qisman to'g'ri. Funkcionallik faqat bajarilishi kerak bo'lgan harakatni ifodalaydi, albatta samaradotlik tavsifi kabi emas. Ushbu matnda imkoniyatlar ishchi tizimning funkcionalligi va ishlash atributlarini o'z ichiga olgan tezkor atama sifatida kelishi mumkin.

1.1- rasmda keltirilgan oddiy diagramma tizimni ifodalaydi. Biroq, analitik nuqtayi nazardan, diagrammada tizimning ish muhitida qanday ishlashi bilan bog'liq muhim berilganlar yo'q. Shuning uchun biz ushbu yetishmayotgan elementlarni aniqlash uchun diagrammani kengaytiramiz. Natija 1.2- rasmda ko'rsatilgan. Kerakli/nomaqbul berilganlar, manfaatdor tomonlar va istal-

magan/nomaqbul natijalarni o'z ichiga olgan tuzilmaning atributlari tizimni belgilash, loyihalash va ishlab chiqishda barcha hissa qo'shadigan omillar to'g'ri hisobga olinishini ta'minlash uchun asosiy nazorat ro'yxati bo'lib xizmat qiladi.



1.1- rasm. Tizim ob'yektining asosiy tuzilishi



1.2- rasm. Analitik tizim ob'yektining tuzilishi

### 1.7. Mutaxassislikni talab qiladigan tizimlar

Avvalroq biz har xil turdagi tizimlarning misollarini sanab o'tdik. Ushbu tizimlarning ba'zilar maktablar, shifoxonalar, bank tizimlari va ishlab chiqaruvchilar kabi tizimlar, mahsulotlar yoki xizmatlar ishlab chiqaradigan ish oqimiga asoslangan tizimlardir. Shunday qilib, ular chuqur, qo'llab-quvvatlovchi faollar va ham-

korlikdagi o'zaro ta'sirlarning samarali tashkiliy tuzilmalarini talab qiladi.

Ba'zi tizimlar ixtisoslashtirilgan tuzilmalarni tahlil qilish, loyihalash va ishlab chiqishni, murakkab o'zaro ta'sirlarni va jamoatchilikning xavfsizligi, sog'lig'i va farovonligiga, shuningdek atrof-muhitga ta'sir qilishi mumkin bo'lgan ishlash monitoringini talab qiladi, tizim muhandisligi talab qilinishi mumkin. Ikkala turdagi tizimlarni tahlil qilish, loyihalash va ishlab chiqish uchun nima talab qilinishini o'rganar ekansiz, ularning ikkalasi ham umumiy tushunchalar, tamoyillar va amaliyotlarga ega ekanligini ko'rasiz. Tijorat tizimlari, masalan, optimal kutish liniyasi yoki ob-havo sharoiti uchun rentabellik va investitsiya daromadini va statistik nazariyani aniqlash uchun tijorat modellari va ishlab chiqish uchun turli xildagi analitik va matematik tamoyillarni qo'llashni talab qilishi mumkin. Juda murakkab tizimlarda analitik, matematik va ilmiy tamoyillarni qo'llash kerak bo'lishi mumkin. Biz buni tizim muhandisligi, elektrotexnika, mashinasozlik va dasturiy ta'minot muhandisligi kabi muhandislik fanlari aralashmasini talab qilishi mumkin bo'lgan tizimlar mutaxassisligi, deb ataymiz. Ushbu fanlar faqat tizim, mahsulot yoki xizmatni tahlil qilish, loyihalash va ishlab chiqishning turli bosqichlarida talab qilinishi mumkin.

Biz ikkala turdagi tizimlarni tahlil qilish, loyihalash va ishlab chiqishda qo'llaniladigan tushunchalar, tamoyillar va amaliyotlarni taqdim etamiz. Tashqi tomondan, bu ikki toifa mutaxassislikni talab qiladiganlar va talab qilmaydiganlar o'rtasidagi aniq farqni anglatadi. Xo'sh, tizim mutaxassisligi qachon talab qilinishini qanday bilamiz?

Aslida, bu ikki toifa tizimlar, mahsulotlar yoki xizmatlarning uzluksizligini ifodalaydi, ular murakkab bo'lishi mumkin bo'lgan qog'oz parchasini ishlab chiqarishdan tortib, murakkab bo'lgan tizimni ishlab chiqishgacha bo'lgan masofani o'z ichiga oladi.

*Tizim mutaxassisligi nima?*

Anig'i - tizimlarning ko'p tarmoqli muhandisligi. Biroq, har

qanday ta'rifda bo'lgani kabi, javob ham qo'shimcha aniqlovchi savollarga ehtiyojni yo'q qilishi kerak. Buning o'rniga, tizim javobining muhandisligi ikkita qo'shimcha savol tug'diradi: muhandislik nima? Tizim nima? Ushbu fikrga qarab, keling, ushbu savollarni batafsil ko'rib chiqaylik.

### 1.8. Asosiy atamalarni aniqlash

Muhandislik bo'yicha talabalar ko'pincha rasmiy ta'lim uchun asos bo'lgan asos atamasi bilan tanishmasdan bitiradilar. *Muhandislik* atamasi, lotincha "*ingenerare*" so'zidan kelib chiqqan bo'lib, bu "yaratish" degan ma'noni anglatadi. Masalan, AQShdagi muhandislik maktablarini akkreditatsiya qiluvchi muhandislik va texnologiya bo'yicha akkreditatsiya kengashi bu atamani quyidagicha ta'riflaydi:

**Muhandislik** – bu "To'qish, tajriba va amaliyot natijasida olingan matematika va tabiiy fanlar bo'yicha bilimlarni insoniyat manfaati uchun tabiiy materiallari va kuchlaridan tejamkorlik bilan foydalanish yo'llarini ishlab chiqish uchun oqilona qo'llanadigan kasbdir" (*mamba*: Muhandislik va texnologiya akkreditatsiya kengashi [ABET]).

Tizim muhandisligini aniqlashning bir qancha usullari mavjud, ularning har biri shaxs yoki tashkilotning nuqtayi nazari, tajribasi va shunga o'xshash narsalarga bog'liq. Tizim muhandisligi turli odamlar uchun har xil narsalarni anglatadi.

Vaqt o'tishi bilan tizim muhandisligi haqidagi shaxsiy qarashlaringiz ham rivojlanishini bilib olasiz. Xo'sh, agar sizda turli xil qarashlar va ta'riflar mavjud bo'lsa, nima qilish kerak? Muhimi, siz dasturiy guruhlar rahbari yoki tashkilotchisiz.

1. Konsensus ta'rifni o'rnating.
  2. Ta'rifni barcha uchun qo'llanma bo'lib xizmat qilish uchun tashkiliy yoki dasturiy buyruq vositalarida hujjatlashtiring.
- Tizim muhandisligining asosiy jihatlarini o'z ichiga olgan

qisqa, yuqori darajadagi ta'rifni afzal ko'rganlar uchun quyidagi ta'rifni ko'rib chiqing:

• Tizim muhandisligi analitik, matematik va ilmiy tamoyil-larning maqbul tavakkalchilikka ega bo'lgan, foydalanuvchining amaliy ehtiyojlarini qondiradigan hamda manfaatdor tomonlar manfaatlarini muvozanatlashgan holda ishlab chiqish va hayot sikli xarajatlarini minimallashtiradigan yechimini shakllantirish, tanlash va ishlab chiqish uchun ko'p tarmoqli qo'llanilishini bilgan mutaxassis.

Tizim muhandisligi, biz ko'rib turganimizdek, tizim muhandisligi *nima* qilishini aniqlashdan ko'ra ko'proq narsani talab qiladigan atamalardan biridir; ta'rif, shuningdek, tizim muhandisligidan qanday foyda ko'rishini aniqlashi kerak. Masalan, ABET muhandislik ta'rifi asosiy maqsadni o'z ichiga oladi "materiallardan iqtisodiy jihatdan va tabiat kuchlaridan insoniyat manfaati uchun foydalanish".

Xuddi shu kontekstni tizim muhandisligi ta'rifiga qo'llash tizimlar, mahsulotlar va xizmatlar foydalanuvchisi insoniyatni anglatadi. Biroq, insoniyatning omon qolishi turni qo'llab-quvvatlaydigan yashash muhitiga juda bog'liq. Shuning uchun, tizim muhandisligi shunchaki "insoniyat manfaati uchun" emas, balki kengroq istiqbolga ega bo'lishi kerak. Tizim muhandisligi shuningdek, insoniyat va yashash muhiti o'rasidagi muvozanatni ham qurbon qilmasdan ta'minlashi kerak.

### 1.9. Xulosa

Bu bob tizim nima ekanligini muhokama qilishni yakunlaydi. Biz "tizim" atamasini aniqladik va turli kontekstlarda atamani belgilashdagi qiyinchiliklarni ta'kidladik. Shuningdek, biz tizim turlaridagi misollarni o'rganib chiqdik; misli ko'rilmagan va misli ko'rilmagan tizimlar o'rasi farqlanadi; tizimlar mahsulotlar va asboblarda kontekstida ko'rib chiqiladi.

Biz boshqa tizimlar, mahsulotlar yoki xizmatlarni ishlab chi-

qaradigan ikkita toifadagi tizimlarni aniqlash bilan yakunladik. Ulardan ba'zilar tizim muhandisligini talab qiladi. Shuning uchun biz tizim ta'rifi bilan birgalikda tizim muhandisligini aniqlashga olib keladigan muhandislikni aniqladik.

Ushbu asosiy tushuncha bilan biz endi har bir tizimni noyob qiladigan asosiy atributlar, xossalari va tavsifnomalarini o'rganishga tayyormiz.

### 1.10. Umumiy mashqlar

1. Tizim bo'yicha o'z ta'rifingizni yarating.
2. Ushbu bobda keltirilgan "tizim" ta'riflari asosida:
  - ✓ Ta'riflardagi kamchiliklarga o'z nuqtayi nazaringizni aniqlang.
  - ✓ Sizning ta'rifingiz ushbu kamchiliklarni bartaraf etishiga nima uchun ishonganingiz haqida mantiqiy dalillar keltiring.
  - ✓ Tarixiy nuqtayi nazardan misli ko'rilmagan tizimlar bilan almashtirilgan uchta oldingi tizimni aniqlang.
3. Siz va tashkilotingiz "tizim"ni qanday aniqlaysiz?
4. Siz va sizning ishchi guruhingiz "tizim" tushunchasiga ega-mi? Agar yo'q bo'lsa, a'zolaridan tizim nima ekanligini mustaqil ravishda ishlab chiqishlarini so'rang. Natijalarni umumlashtiring va jamoaga individual nuqtayi nazarni taqdim eting.
5. Natijalarni muhokama qiling va konsensus ta'rifini shakllantiring. Natijalarni kursdoshlarga xabar qiling.
6. Qanday xilma-xil fikrlarni kuzatdingiz?
7. Konsensusga erishish uchun jamoa qanday konsepsiya yoki semantik to'siqlarni yengib o'tishi kerak edi?
8. Quyidagi ro'yxatda keltirilgan tizim muhandisligi ta'riflarini o'rganing. Ushbu ta'riflarni solishtiring va taqqoslang va qaysi biri sizning e'tiqodlaringiz va tajribalaringizga mos kelishini aniqlang.

➤ AFSCM 375-1

- Sobiq FM 770-1
- Sobiq MIL-STD-499A
- EIA/IS-731.1
- Mudofaa tizimlarini boshqarish kolleji (DSMC)
- Tizim muhandisligi bo'yicha xalqaro kengash (INCOSE)
- Xalqaro standartlashtirish tashkiloti (ISO).

9. Tashkilotingiz ishlab chiqaradigan tizim, mahsulot yoki xizmat uchun uni yaratish yoki qo'llab-quvvatlash uchun zarur bo'lgan tarkibiy mahsulotlar va vositalarni (masalan, tashqi tizimlar) aniqlang.

10. Siz kuzatayotgan paradigmalarni aniqlang: (a) tashkilot, (b) mijozlar va (c) tizim yoki mahsulot loyihasiga ta'sir qiluvchi tijorat sohasi. Har bir paradigma uchun manfaatdor tomonlar paradigmanni o'z-o'zidan ravshan qiladigan xarakterli iboralardan foydalanadilar.

11. Sizning tashkilotingiz tizim muhandisligini qanday ko'radi va belgilaydi?

12. Muallifning tizim muhandisligi ta'rifi sizning tajribangiz bilan qanday taqqoslanadi?

13. Tashkilotingiz yoki dasturingiz tizim muhandisligini aniqlashda qanday qiyinchiliklar va paradigmalarga duch keladi?

### 1.11. Nazorat uchun savollar

1. Tizim degani nima?
2. Tizimlar qanday turlarga bo'linadi?
3. Tizim chegaralariga nimalar kiradi?
4. Foydalanuvchi tashkiloti ichida tizim qanday rolni bajaradi?
5. Tizim qanday topshiriq ilovalarini bajaradi?
6. Tizim qanday yo'naltirilgan natijalarni beradi?
7. Tizim turlariga qanday misollar keltirish mumkin?
8. Tizimlar, mahsulotlar va asboblarning o'rtasidagi farqlar qanday?
9. Precedent va precedent bo'lmagan tizimlar o'rtasidagi farq nima?
10. Qanday qilib biz tizimni analitik tarzda ifodalaymiz?

## II- BOB. TIZIMNING XUSUSIYATLARI, XOSSALARI VA TAVSIFLARI

Tizim muhandisligi tizimni, mahsulot yoki xizmatni uning xususiyat (atribut)lari, xossalari va ishlashi nuqtayi nazaridan qanday tavsiflashni tushunish uchun kuchli poydevorni ishlab chiqishni talab qiladi.

Ushbu bobda ko'pchilik tabiiy va inson tomonidan yaratilgan tizimlarda umumiy bo'lgan tizim hususiyatlari keltirilgan. Bizning mulohazalarimiz ushbu atributlarni foydalanuvchilar tizim tasniflari uchun nazorat ro'yxati sifatida foydalanishi va tizimni yaratuvchilar/xizmat provayderlari ushbu tasniflarning muvofiqligini baholash uchun foydalanishi mumkin bo'lgan qolip nuqtayi nazaridan ko'rib chiqiladi. Maqsad, sizga quyidagilarni o'rganishga imkon berishdir – bihinchidan fikrlash, ikkinchidan tashkillashtirish va uchincuidan tizimlarni tavsiflash. Ushbu bilim tizim mutaxassisleri va tizim tahlilchilariga ikki xil tarzda yordam beradi.

Birinchidan, tasniflarni tahlil qilinganida va baholanganida, tez-tez ishlatiladigan xususiyatlar, xossalari va tavsiflarning nazorat ro'yxatlari bizga haqiqatni tekshirish imkonini beradi va tasniflarning talablarida har qanday "tashiklar"ni aniqlaydi.

Ikkinchidan, biz tasniflarni ishlab chiqqanimizda, u asosiy imkoniyatlarni va ularning darajalarini tartibga solish va belgilash uchun mos yozuvlar ro'yxatini taqdim etadi.

### 2.1. Asosiy atamalarning ta'rifi

**Imkoniyat** – bu funksiya (harakat)ni tashqi buyruqlar bilan tugatilgunga qadar, belgilangan muddatda bajarish yoki zaxiralar tugaguniga qadar ishlashning belgilangan darajasida bajarish

uchun tashqi stimuly tomonidan faollashtirilgan yoki qo'zg'atilgan aniq, o'ziga xos xususiyat.

**Moslik.** Ob'yektning boshqa element bilan belgilangan chegaralar to'plamida osonlik bilan va hech qanday aralashuvsiz ulanishi uchun mosligi.

**Shakl.** Bir yoki bir nechta interfeys chegarasi maqsadlarini qo'llab-quvvatlash uchun mo'ljallangan elementning belgilangan shakli.

**Shakl, imkoniyat va funksiya.** Konfiguratsiyani boshqarishda ushbu konfiguratsiya ob'yekt sifatida ob'yektning jismoniy va funksional xususiyatlarini o'z ichiga oladi, lekin elementni tashkil etuvchi elementlarning har qanday xususiyatlarini o'z ichiga olmaydi.

**Funksiya.** Belgilangan samaradorlik chegaralari doirasida muayyan maqsadga erishish uchun tizim elementi tomonidan bajariladigan amal, faoliyat, jarayon yoki harakat. Funksiyalar ishni o'z ichiga oladi - masalan, kuchni masofa bo'ylab harakatlantirish, ma'lumotni tahlil qilish va qayta ishlash, energiya yoki jismoniy xususiyatlarni o'zgartirish, qarorlar qabul qilish, aloqalarni o'tkazish va boshqa ishlab chiqish muhitidagi tizimlar bilan o'zaro hamkorlik qilish.

**Funksionallik.** Atributlari - o'lchanadigan ishlash parameterlari, shu jumladan ishonchlilik, barqarorlik va xavfsizlikdan iborat holat.

**Ishlash darajasi.** Ob'yektiv, o'lchanadigan parametr, tizimning ssenariy farazlari, boshlang'ich shartlari va ish sharoitlari to'plamiga asoslangan funksiyani bajarish qobiliyatini chegaralash uchun xizmat qiladi. Masalan, tizim samaradorligi, "xodimlar" elementining malakasi va tizim samaradorligi.

**Ishlash.** Amal yoki funksiyani bajarish bilan bog'liq jismoniy yoki funktsional atributni tavsiflovchi miqdoriy o'lchov. "Ish" atributlariga quyidagilar kiradi: miqdor (qancha), sifat (qanchalik yaxshi), qamrov (qanchalik maydon, qanchalik uzoq), o'z vaqti-

dalik (qanchalik sezgir, qanchalik tez-tez) va tayyorlik (mavjudlik, missiya/tezkor tayyorgarlik).

**Jismoniy xususiyatlar.** Tarkib, o'lchamlar, tugatish, shakl, moslik va ularning tegishli tolerantliklari kabi moddiy xususiyatlarning miqdoriy va sifatli ifodalari.

## 2.2. Xususiyatlar, xossalari va tavsiflarning umumiy tushunchalari

Biz mutaxassislarining tizimning xususiyatlari, hossalari va tavsiflariga murojaat qilishlarini tez-tez kuzatamiz. Yoqorida keltirilgan atamalarning ta'riflarini o'rganayotgan tasodifiy kuzatuvchidan ko'ra, lug'atlarning ko'pchiligi bu atamalarni boshqasiga havola qilish orqali belgilaydi. Muhokamalarimiz maqsadlari uchun biz atamalar orasidagi farqlarni aniqlash vositasi sifatida quyidagilarni ishlatamiz.

### Atributlar

Atributlar atamasi tizimning funktsional yoki jismoniy xususiyatlarini tasniflaydi. Masalan, ismi; familiyasi; jinsi; ma'lumoti; birlik narxi; fuqaroligi; davlati; yashash shahri; sport turi; lavozimi; reys raqami va sanasi va h.k.

### Xossalari

Xossalari — tizimning o'lchamlik xossalari bildiradi. Masalan, vazn; zichlik; uzunlik, kenglik yoki balandlik kabi o'lcham.

### Tavsiflar

Tavsiflar atamasi har bir tizimni o'ziga xos tarzda aniqlaydigan xulq-atvor va jismoniy fazilatlarni anglatadi. Xulq-atvor xususiyatlariga misollar bashorat qilish va sezgirlikni o'z ichiga oladi. Jismoniy xususiyatlarga misollar uskunani isitish va barqarorlashtirish profillarini o'z ichiga oladi; uskunaning termal imzolari; samolyot raqarlarining kesishishlari; avtomobilning kruiz tezligiga tezlashishi, ishlov berish yoki to'xtash; kit nayzalari belgilari.

Tizimning atributlari, xossalari va tavsiflarining yig'indisi tizimni, mahsulot yoki xizmatni bir xil tasniflar bo'yicha boshqalardan noyob tarzda aniqlaydi va ajratib turadi. Ushbu o'ziga xoslikni ko'rsatish uchun ko'pchilik tizimlar uchun umumiy bo'lgan bir nechta jihatlarni ko'rib chiqamiz.

## 2.3. Har bir tizimdagi noyob aniqlovchi

Barcha tabiiy va inson tomonidan yaratilgan tizimlarning o'ziga xos alomatlari (belgilari) mavjud, masalan, ularning rollari, xatti-harakat modellari, temperamenti va tashqi ko'rinishi, hatto bir tur ichida ham bo'lishi mumkin. Umuman olganda, noyoblikning asosiy atributlari 2.1- jadvalda tasvirlangan quyidagi elementlarni o'z ichiga oladi.

2.1- jadval. Tizim hususiyatlari, xossalari va ularning tavsifi

ID	Atribut	Tavsifi
1	Tizim xayrixohlari	Har bir tizimda kamida bir yoki bir nechta xayrixohlar mavjud, masalan, egalari, uning xatti-harakatlaridan foyda ko'radigan ma'murlar, operatorlar va ta'minotchilar, mahsulotlar, yon mahsulotlar yoki xizmatlar.
2	Tizimning hayot sikli	Har bir tizim, mahsulot va xizmat uning darajasini ko'rsatadigan hayot aylanishiga ega ekanligi.
3	Tizimning tezkor domeni	Har bir tizim o'zini chegaralaydigan tezkor domenga yoki "ta'sir doirasiga" ega qamrovning sohasi, amallari va samaradorligi. Odamlar cho'zishni o'rgandilar. Muayyan tizimga imkon beradigan boshqa aktivlarni qo'llash orqali qamrov sohasidagi uning oraligini "kengaytiring". <i>1- misol.</i> Samolyot ma'lum bir foydala-

		nish ostida ma'lum masofaga ega yoqilg'i, foydali yuk va ob-havo kabi sharoitlar. Yoqilg'i quyish manbalarini joylashtirish - havo-tankerlari - va uning missiyasi parvoz yo'li bo'ylab texnik xizmat ko'rsatish inshootlarining mumkin bo'lgan oraligini kengaytirish.
4	Tizimdagi berilganlar doirasi	Har bir tizimning vaqtning istalgan nuqtasida tizimi vazifasini bajaradi doimiy yoki vaqtinchalik ma'lumotnoma: 1. Uning amaliy domeni uchun amallar bazasi. 2. Navigatsiya uchun asos. 2- <i>misol</i> . Samolyot doimiy bazaga tayinlanishi mumkinligi, faoliyatining markazi bo'lib xizmat qiladi. Samolyotni faoliyatini amalga oshirishi uchun buyurtma berilishi kerak. Masalan, bazadan maxsus (vaqtinchalik) topshiriqlar. Samolyot bazadan o'zining ish faoliyati vaqtida foydalangan ma'lumotnoma.
5	Yuqori tartibli tizimlar	Har bir tizim: 1. Boshqarishi, yo'naltirishi, cheklashi mumkin bo'lgan yuqori tartibli tizimning bir qismi sifatida ishlaydi yoki uning amalda ishlashini nazorat qiladi. 2. Missiyalar uchun zaxiralarni taqdim etadi.
6	Maqsadga asoslangan	Koinotni "tizimlar tizimi" sifatida ko'rib chiqsak, har bir tabiiy va sun'iy tizim o'zining dastlabki qabul qiluvchisi yoki tizim egasi tomonidan taxmin qilinga-

	rol	nidek, uning mavjudligi sababiga asoslangan foydali rolga ega.
7	Tizim vazifalari	Har bir tizim egasi va foydalanuvchilar tomonidan belgilangan natijalarga asoslangan ishlash maqsadlariga erishish uchun o'z maqsadini amalga oshirish uchun vazifalarni bajaradi.
8	Missiya va ishlash maqsadlari	Har bir tizim va missiya bir qator maqsadlar va vazifalar bilan tavsiflanishi hamda ob'yektiv hujjatlashtirilishi kerak. tizim egasi tomonidan zaxira xarajatlari uchun asosiy asos va aktsiyadorlar rejalashtirilgan ko'p qirrali yutuqlar va investitsiyalardan kutilgan daromad asosida, maqsad va vazifalarni belgilaydi. Har bir maqsad miqdoriy, o'lchanadigan, tekshiriladigan yoki tekshirilishi mumkin bo'lgan bir yoki bir nechta aniq maqsadlar bilan qo'llab-quvvatlanishi kerak.
9	Tizimning ishlash shartlari va cheklovlari	Har bir tizim o'z vazifasini bajarishda ma'lum bir to'plamga bo'ysunadi va tizimning yuqori tartibli cheklovlari va shartlari orqali nazorat qilinadi.
10	Tizim yordamchisi	Har bir tizim jismoniy, psixologik, ijtimoiy, moliyaviy va o'z foydalanuvchisi uchun iqtisodiy qo'shimcha qiymat keltiradi. Tizim yordamchi dasturi foydalanish qulayligini o'z ichiga oladi, foydali va boshqa xususiyatlarga ega bo'ladi.
11	Tizimning mosligi	Har bir tizim o'zining rejalashtirilgan qo'llanilishiga mos kelishi va foydalanuvchilarning tashkiliy tizimiga integratsiyalashuvi nuqtayi nazaridan foyda-

		lanuvchi uchun amaliy moslik darajasiga ega bo'ladi. <b>3- misol.</b> Gaz bilan ishlaydigan maysazor mashinasi, daraxtlar va gul bog'lari atrofidagi o'tlarni kesish uchun mos keladi; agar sizda maysa o'roq mashinasi bo'lmasa, u mashina maysa o'rish uchun mos emas.
12	Tizim muvaffaqiyatining mezonlari	Har bir tizim va topshiriq, tizim egasidan bir qator muvaffaqiyat mezonlarini talab qiladi va aksiyadorlar, qanday ob'yektiv mezonlar muvaffaqiyatli ekanligini angulasalar, maqsadlar va natijalarga yo'naltirilgan vazifalar orqali missiyani bajarishga rozi bo'ladilar. Yakuniy muvaffaqiyat, bu foydalanuvchining qabul qilishi va qoniqish darajasidir.
13	Missiya-ning ishonchiligi	Har bir tizim ma'lum bir missiya davomiyligi va amaliy muhit sharoitlari va ssenariylari to'plami uchun missiya maqsadlarini amalga oshirishda muvaffaqiyat qozonish ehtimoli bilan tavsiflanadi.
14	Tizimning samaradorligi	Har bir tizim harajat birligi uchun kutilayotgan muvaffaqiyat ehtimoli bilan tizimning missiyasini bajarish bilan bog'liq ma'lum darajadagi xarajat va texnik samaradorlikka ega. <b>4- misol.</b> Ta'lim tizimi yoki sog'liqni saqlash tizimining tizim samaradorligini ko'rib chiqing. Bundagi qiyinchiligi sunda-ki, qaysi manfaatdor tomonlar nuqtayi nazaridan samaradorlikni baholash, kerakligida.

15	Tizim samarasi	Har bir tizim xom ashyo, berilganlar, rag'batlantirish, ishora va h.k. larni qayta ishlashda samaradorlik darajasiga ega. Muhandislar sifatida biz kirishdagi ma'lum miqdor uchun ishlab chiqarilgan mahsulot miqdori nisbatini ta'minlovchi samaradorlik ko'rsatkichini tayinlaymiz.
16	Tizim integratsiyasi	Har bir tizim amaliy cheklovlar va sharoitlarga qaramasdan tizimlar, mahsulotlar va xizmatlarni talab qilinadigan tarzda yetkazib berish qobiliyatida yaxlitlik darajasiga ega bolishi kerak.
17	Tizimning barqarorligi	Har bir tizim, mahsulot yoki xizmat o'z vazifasini bajarishda muvaffaqiyatga erishish uchun xodimlar, mablag'lar, sarflanadigan materiallar kabi zaxiralarni talab qiladi; tuzatuvchi va profilaktika ishlari; va ehtiyot qismlar, materiallar va o'qitish kabi yordam masalalari hisobga olinadi.
18	Tizimni targ'ib qilish	Ba'zi tizimlar, xususan, korxonalar o'z tizimlarini namoyishlar, reklamalar va hokozalar orqali sotishni kutishadi. Rag'batlantirish faoliyati himoya va xavfsizlikni talab qilishi mumkin. <b>5- misol.</b> Nashriyot yangi kitobni ma'lum bir kun va vaqtda turkumda chiqarishni, kitobni reklama orqali targ'ib qilishni va kitob do'konlari egalariga sotish cheklovlari va shartlarini qo'yishni rejalashtirmoqda. Kitob do'konlari egalari kitobni rasmiy nashrga qadar 24 soatlik kuzatuv (qo'riqlash) ostida qulf va kalit (himoya)

		ostida saqlashlari kerak.
19	Tizim tahdidlari	Har bir tizim va uning vazifalari do'stona, yaxshi yoki dushman niyatlar yoki harakatlarni namoyon qilishi mumkin bo'lgan ish muhitida raqobatchilar yoki raqiblar tomonidan tahdid qilinishi mumkin.
20	Tizimni yashirish	Zaifliklar yoki ajablanish elementiga bo'lgan ehtiyoj tufayli, ba'zi tizimlar o'zlarining shaxsiylikini himoya qilish yoki o'zgartirish uchun kamuflyaj yoki yashirishni talab qiladi.
21	Tizim himoyasi	Har bir tizim tashqi tahdidlarga nisbatan zaifligini minimallashtirish uchun ma'lum darajadagi himoyaga ega bo'lishi kerak.
22	Tizim xavfsizligi	Inson tomonidan yaratilgan tizimlar jismoniy xavfsizlik, aloqa xavfsizligi, amaliy xavfsizlik va axborot xavfsizligi kabi xavfsizlik darajasini saqlab turishi mumkin.
23	Tizim arxitekturasini	Har bir tizim ko'p darajali, mantiqiy (funktional) va jismoniy tuzilish yoki arxitekturadan iborat bo'lib, uning shakli, mosligi va funksiyasi uchun asos yaratadi.
24	Tizim imkoniyatlari	Har bir tizim, ta'rifiga ko'ra, qayta ishlash, kuchli tomonlar, uzatish funksiyalari kabi o'ziga xos qobiliyatlarga ega bo'lib, ular unga xom ashyo, ma'lumot va stimullar kabi kirishlarni qayta ishlashga va xatti-harakatlar namunalarini, mahsulotlar va qo'shimcha mahsulotlar shaklida javob be-

		rishga imkon beradi. Amaliy domenlar kabi tizim imkoniyatlari asboblari yoki boshqa tizimlar yordamida kengaytirilishi mumkin.
25	Tizim amaliy konsepsiyasi	Har bir tizimda tizim egasi, tizim ishlab chiquvchisi va/yoki tizim boshqaruvchisi tomonidan nazarda tutilgan amaliy konsepsiyasi mavjud. Bu konsepsiya amaliy maydonni, tizim imkoniyatlarini, interfeyslarni va amaliy muhitni chegaralash uchun asos bo'lib xizmat qiladi.
26	Tizim fazalari, tartibotlari va ishlash holatlari	Har bir tizim/mahsulot yashash sikli bosqichi uchun har bir tizim, mahsulot yoki xizmat davriy yoki takrorlanmaydigan (bir martalik) bo'lishi mumkin bo'lgan bir qator bosqichlar, tartibotlar va ish holati orqali rivojlanadi.
27	Amaliy me'yorlari standartlari va konvensiyalari	Har bir tizim o'z faoliyati, axloqi, axloqi va bag'rikengliklarini boshqaradigan bir qator amaliy me'yorlar, standartlar va konvensiyalardan foydalanadi.
28	Tizim tavsifi	Har bir tizim tizim arxitekturasini, uning elementlarini, interfeyslarini va h.k. larni tavsiflovchi tizim tavsifiga ega bo'lishi kerak. Bu xususiyatlarning har biri tizimning imkoniyatlari va muhandislik samaradorligi parametrlari bilan ifodalanadi, ular tizimning ishlashi tasnifida talablar sifatida ko'rsatilishi va ifodalanishi kerak.
	Tizimning ishlash	Har bir tizim yuqori darajadagi boshqaruv tomonidan qo'yiladigan jismoniy imkoniyatlar bo'lishi mumkin bo'lgan ishlash

29	shartlari va cheklovlari	cheklovlari va shartlariga ega bo'lishi mumkin. Bular xalqaro, hukumat, ekologik, ijtimoiy, iqtisodiy, moliyaviy, psixologik va boshqalardan iborat bo'lishi mumkin.
30	Tizim datchiklari	Har bir tabiiy va inson tomonidan yaratilgan tizim sensorning qandaydir shakliga ega tashqi ogohlantirishlar va signallarni aniqlashga imkon beradi.
31	Tizimning xatti-harakati, loyihasi	Har bir tizim xatti-harakatlar modellari bilan tavsiflanadi.
32	Tizimning hozirjavobligi va sezgirligi	Har bir tizim xom ashyo yoki stimullarni qayta ishlash va javob berish qobiliyatini tavsiflovchi o'tkazish qobiliyati kabi ishlaganiga asoslangan xatti-harakatlar xususiyatlariga ega. Biz tezkorlikni uning sezgirligi deb ataymiz. 6- <i>misol.</i> Tezlashtiruvchi platalar kompyuter protsessorlariga sezgirlikni oshirish imkonini beradi.
33	Tizim interfeyslari	Har bir tizim ichki va tashqi interfeyslarga ega bo'lib, uning o'zi va uning ish muhiti ichida o'zaro ta'sirini ta'minlaydi.
34	Tizim "ildizi"	Har bir tizimda kamchiliklar, nuqsonlar, xatolar va h.k. larni tuzatish uchun oldingi tizim loyihalari, texnologiyalari va ushbu loyihalarni takomillashtirishdan olingan "ildizi" mavjud.
35	Missiya zaxiralari	Har bir tizim motivatsiya, manevr, jarayon va xatti-harakatlar va jismoniy javoblarni rag'batlantirish uchun zarur bo'lgan aniq

	(tizimga kirishlar)	harakatlarga aylantirilishi mumkin bo'lgan vazifalar, sarflanadigan materiallar, sarf materiallari va operator harakatlari kabi ma'lumotlarni talab qiladi.
36	Tizim mahsulotlari, xizmatlari va qo'shimcha mahsulotlar	Har bir tizim ishlab chiqaradi: 1. Qo'shilgan qiymatli mahsulotlar va/yoki manfaatdor tomonlarga foyda keltiradigan xizmatlarni amalga oshiradi. 2. Tizimning ishlashiga va/yoki uning ish muhitiga ta'sir qilishi mumkin bo'lgan qo'shimcha mahsulotlar. 7- <i>misol.</i> Qo'shimcha mahsulotlar issiqlik, chiqindi mahsulotlar - axlat, chiqindi gazlar, termal belgilar va ranglarni o'z ichiga olishi mumkin.
37	Protsedurali berilganlar	Har bir inson tomonidan yaratilgan tizim uskunalari, xizmatlar va operator interfeyslari va tashqi tizimlar bilan interfeyslari bilan bog'liq xavfsiz ishlash tartib-qoidalarini tavsiflovchi protsessual ma'lumotlarni talab qiladi.
38	Tizimning halokatliligi	Ba'zi mudofaa va hujum tizimlari o'zlarining halokatliligi bilan tavsiflanadi - tahdid yoki nishonni yo'q qilish yoki zarar etkazish, o'chirish, zararsizlantirish yoki boshqa yo'l bilan zarar yetkazish potentsialining mavjudligi.
		Har bir tizim o'zining xulq-atvori va jismoniy xususiyatlarida noaniqlik yoki kamchiliklarni ochib beradigan zaifliklarga ega. Zaiflik jismoniy, psixologik, ijtimoiy, iqtisodiy, xavfsizlik, maxfiylik va boshqa omillarni o'z ichiga

39	Tizim zaifligi	oladi. <i>8- misol.</i> Harbiy tanklar qo'shimcha himoya qatlamlariga ega to'g'ridan - to'g'ri zarbalarning ta'sirini minimallashtirish. Internet saytlarida kompyuter "xakerlari" uchun zaifliklar mavjud.
40	Tizimning saqlanish qobiliyati	Har bir tizim nosozliklarga chidamlilik darajalariga ega bo'lib, ular ma'lum bir ichki yoki tashqi sabab yoki nosozliklar uchun past darajadagi ishlashda missiyalarni bajarish va missiya maqsadlariga erishish imkonini beradi.
41	Tizimning mavjudligi	Talab bo'yicha topshiriqni bajarish uchun tizimning amaliy tayyorligi holati. Mavjudlik tizimning ishonchliligi va barqarorligi funksiyasidir.
42	Tizim estetikasi	Har bir tizim hissiyotlarni jalb qiladigan yoki uning manfaatdor tomonlariga estetik jihatdan yoqimli bo'lgan psixologik yoki tashqi ko'rinish xususiyatlariga ega.
43	Tizimdagi nuqsonlar	Har bir tizim o'zining rivojlanishida o'ziga xosdir. Bunga tizimning ishlashiga ta'sir qilishi yoki tashqi ko'rinishiga ko'ra uning qiymatini kosmetik jihatdan kamaytirishi mumkin bo'lgan dizayndagi kamchiliklar va xatolar, ish sifati va material nuqsonlari, kamchiliklar va boshqalar kiradi.
44	Tavakkalchilik	Har bir tizim, mahsulot yoki xizmat vazifa bilan bog'liq xavf elementiga ega amallar va uning ish muhiti, jumladan: 1. Vujudga kelish ehtimoli. 2. Muvaffaqiyatsizlikning oqibatlari.

45	Tizim atrof-muhit, xavfsizlik va salomatlik	Inson tomonidan yaratilgan har bir tizim tizimning/mahsulotning hayot siklining turli bosqichlarida tizim xodimlari – operatorlar va texnik xizmatchilar, xususiy va jamoat mulki, atrof-muhit va boshqalar uchun ekologik, xavfsizlik yoki sog'liq uchun xavf tug'dirishi mumkin.
46	Tizim salomatligi holati	Har bir tizim amaliy sog'liq holatiga ega bo'lib, u foydalanuvchi topshiriqlarini bajarish yoki qo'llab-quvvatlashga tayyorligining hozirgi holatini ifodalaydi.
47	Tizimga egalik qilishning umumiy qiymati	Har bir inson tomonidan yaratilgan tizim o'z hayot sikli davomida umumiy egalik qiymatiga ega bo'lib, u takrorlanmaydigan va takroriy rivojlanish amaliy xarajatlarini o'z ichiga oladi.

#### 2.4. Tizim samaradorligini tushunish

Umuman olganda, tizimning faoliyati muvaffaqiyatning yakuniy darajasini belgilaydigan asosiy omil hisoblanadi. Tizim funktsionalligi ko'pincha tizimning ishlashi uchun "saralash mezonini" sifatida qaraladi. Foydalanuvchi nuqtayi nazaridan, tizim o'z vazifalarini bajarishda samarali bo'ladimi, missiya va maqsadlari-chi?

Munozarani "samaradorligi"ni aniqlashdan boshlaymiz.

*Samaradorlik toifalari.*

Tizimlarni tadqiq qilganingizda, tez orada ishlashning ikkita asosiy toifasini topasiz: 1) ob'yektiv unumdorlik va 2) subyektiv unumdorlik. Keling, ushbu atamalarni aniqlaymiz:

- O'lchanadigan jismoniy dalillarni keltirib chiqaradigan ob'yektiv samaradorlik oldindan belgilangan mezonlar asosida tizim samaradorligi. Masalan, suvning harorati 84°.

• Subyektiv samaradorlik shaxsiy qadriyatlar, talqinlar yoki istiqbollarga qarab o'zgarib turadigan subyektiv sifat bilan ko'rsatilgan samaradorlik. Masalan, suvning "iliq" yoki "issiq"ligi.

Ushbu ta'riflarni hisobga olgan holda, har bir unumdorlik toifasini batafsilroq ko'rib chiqamiz.

Maqsadli unumdorlik tizim muhandisligi nuqtayi nazaridan, ayniqsa tasniflarni yozishda, tizimning imkoniyatlari va kutilgan unumdorlik darajalari aniq, miqdoriy, o'lchanadigan, sinovdan o'tkazilishi mumkin va subyektiv talqinlarning ta'sirisiz tekshiriladigan parametrlar.

**Ob'yektiv samaradorlikning** misollari quyidagi ko'rsatkichlarni o'z ichiga oladi:

- |                   |                    |                         |
|-------------------|--------------------|-------------------------|
| • Vaqt            | • Burchak          | • Ta'mirlash imkoniyati |
| • Masofa          | • Siqilish         | • Ishonchlilik          |
| • Hajm            | • Tezlik           | • Hosildorlik           |
| • Uzunlik         | • Tezlashtirish    | • Samaradorlik          |
| • Chuqurlik       | • Bosish           | • Harorat               |
| • Qalinlik        | • Qattqlik         | • Bosim                 |
| • Og'irlik        | • Yumshoqlik       | • Namlik                |
| • Ovoz balandligi | • Ot kuchi         | • Xatolar soni          |
| • Zichlik         | • Yopishqoqlik     | • Ko'rish maydoni       |
| • Jismoniy holat  | • Chastota         | • Rezolyutsiya          |
| • Narx            | • Intensivlik      | • Kamchiliklar          |
| • Kuchlanish      | • To'lqin uzunligi | • Tok kuchi             |

Bu ob'yektiv samaradorlik parametrlarining bir nechta misollari. Keling, boshqa turdagi, subyektiv ishlashni tekshiramiz.

### **Subyektiv samaradorlik**

Subyektiv faoliyatni tavsiflash va miqdorni aniqlash qiyinroq. Qizig'i shundaki, biz so'rovlar va intervyular, testlar orqali o'lchanadigan, sinovdan o'tkaziladigan va tekshirilishi mumkin bo'lgan subyektiv ishlash parametrlariga o'zboshimchalik bilan

miqdorlarni belgilashimiz mumkin. Biroq, so'rov yoki intervyu ishtirokchilaridan o'lchanadigan bayonotga o'zlarining afzallik darajasi, roziligi va yoqtirish/yoqtirmasliklarini ko'rsatish so'ralganda, javob baribir sharh, baho, fikr va hokazolarni talab qiladi. Shunday qilib, javob o'tgan tajriba va olingan saboqlar asosida taxallus bo'lishi mumkin. Subyektiv ishlash misollariga quyidagilar kiradi:

- Sifat – tiniqlik, tashqi ko'rinish va rang
- Qarindoshlik
- Yoquvchanlik
- Fikr
- Silliqlik
- Qoniqish – rohatlanish va lazzatlanish.

### **2.5. Tizimning xususiyatlari**

Tizimlarni, ayniqsa marketing yoki tahlil uchun tavsiflaganimizda, to'rtta asosiy tur mavjud.

Biz ko'rib chiqadigan xususiyatlar quyidagilardan iborat:

- ❖ umumiy xususiyatlar,
- ❖ amaliy yoki xatti-harakatlar xususiyatlari,
- ❖ jismoniy xususiyatlar,
- ❖ tizim estetikasi.

#### **Umumiy xususiyatlar**

Tizimning yuqori darajadagi xususiyatlari uning umumiy xususiyatlaridir. Biz ko'pincha marketing broshyuralarida umumiy xususiyatlarni ko'ramiz, bu erda mijoz yoki mijozni jalb qilish uchun asosiy xususiyatlar ta'kidlangan.

Bu ob'yektiv ishlash parametrlarining bir nechta misollarini ko'ramiz.

#### **9- misol.**

• Avtomobilning umumiy xususiyatlari – ikki eshikli yoki to'rt eshikli modellarda mavjud; kabriolet yoki sedan;

konditsionerli qulaylik; mustaqil to'xtatib turish; qoraytirilgan oynalar.

- Samolyotning umumiy xususiyatlari – 100 yo'lovchi, 2000 km. masofasi, boshqaruv imkoniyatlari.

- Oliy ta'lim muassasining umumiy xususiyatlari – 300 nafar xodim; shundan: professor- o'qituvchlar 30ta Dsc, 70ta PhD, 120ta magistr, 80ta texnik xodimlar; yillik daromadi 300 million so'm.

- Tarmoqning umumiy xususiyatlari mijoz-server arxitekturasini, 500 shaxsiy kompyuterlar, Windows va Unix platformalari, xavfsizlik devori, masofaviy dial-up kirish, internet magistrali, tarmoq fayl tuzilishi.

#### *Amaliy yoki xulq-atvor xususiyatlari*

Umumiy xususiyatlardan pastroq tafsilot darajasida tizimlar amaliy xususiyatlarga ega. Foydalanish imkoniyati, omon qolish va belgilangan amaliy muhit uchun ishlash bilan bog'liq tizim xususiyatlarini tavsiflash.

Quyidagi misollarni ko'rib chiqing:

#### *10- misol.*

- Avtomobilni ishlatish xususiyatlari – manevr qobiliyati, burilish radiusi 180 sm., 6 soniyada 0 dan 120 km. gacha tezlik va h.k.

- Samolyotni ishlatish xususiyatlari – har qanday ob-havoda qo'llanilishi, tezligi, o'rindiqlar soni va boshqalar.

- Tarmoqning ishlash xususiyatlari – avtorizatsiya, kirish vaqti, kechikish va h.k.

#### *Jismoniy xususiyatlar*

Har bir tizim funksional bo'lmagan atributlar bilan bog'liq bo'lgan jismoniy xususiyatlar bilan tavsiflanadi – hajmi, vazni, rangi, sig'imi va interfeys atributlari sifatida.

Quyidagi misollarni ko'rib chiqing:

#### *11- misol.*

- Avtomobilning jismoniy xususiyatlari – 3000 kg og'irligi; 17,0 kub sm yuk hajmi; 50 litr yoqilg'i sig'imi; 6250 aylanish tezligida 240 ot kuchiga ega dvigateli, mavjud 7 ta rangi.

- Korxonada yoki tashkilotning jismoniy xususiyatlari – 5000 kv. m ofis maydoni, 15 ta tarmoqqa ulangan kompyuter, 100 000 kvadrat metrlik ombor.

- Tarmoqning jismoniy xususiyatlari – 100 Mb internet magistrali, topografiyasi, routerlari, shlyuzlari.

#### *Tizimning estetik xususiyatlari*

Umumiy, amaliy va jismoniy xususiyatlar ob'yektiv ishlash parametrlari hisoblanadi. Biroq, subyektiv xususiyatlar haqida nima deyish mumkin? Biz ularni tizimli estetik xususiyatlari, deb ataymiz, chunki ular tizimning "ko'rinishi va hissiyoti" bilan bog'liq. Shubhasiz, bunga psixologik, sotsiologik, va foydalanuvchi, ekvayer yoki tizim egasining afzalliklariga murojaat qilish bilan bog'liq madaniy istiqbollar. Shunday qilib, ba'zi xaridorlar mustaqil qaror qabul qiladilar, boshqalari esa tashqi ta'sirga ega tizimlar (ya'ni, boshqa xaridorlar) jamiyat yoki korporativ maqom, imidj va shu kabilar.

### **2.6. Majburiy shartlar**

Tizim barqarorligi, yaxlitligi va ishlashning izchilligi tizimlar orasidagi o'tishni talab qiladi. Bosqichlar, amallar va vazifalar hech qanday ta'sirsiz aniq bir ma'noli o'tishlarga ega. Shunday qilib, tizimlar loyihachilar tomonidan zarur bo'lishi kerak bo'lgan dastlabki shartlar yoki mezonlarga ega, deb taxmin qilinadi. Keyingi bosqichga, amalga yoki vazifaga kirishdan oldin bajarilishi kerak. Ta'rifga ko'ra, tizimdagi berilganlar integratsiyalangan elementlar to'plamidan iborat bo'lib, bu barcha elementlarni ta'minlash uchun muhimdir, bunda tizim sinxronlashtiriladi va uyg'unlashadi.

#### *Faoliyatning dastlabki shartlari va holati.*

Tizimning dastlabki ish sharoitlari tizimning jismoniy va amaliy holatlaridan iborat va tizim missiyasi bosqichining boshida uning atrofidagi ish muhiti, ishlashi, yoki vazifasi hisobga olinadi. Chunki tahlillar ko'pincha ba'zilarini tekshirish uchun asosiy taxminlarni o'rnatishni talab qiladi. Tizim fazalari, amallari yoki vazifalari, dastlabki shartlar "oniy tasvir" yoki boshlang'ich nuqtasi bo'lib xizmat qiladi. Bu bir qancha taxminlarni qamrab oladi.

Ushbu konsepsiyani tasvirlash uchun quyidagi misolni ko'rib chiqing.

### *12- misol.*

Samolyot 35 km tezlikni kesib o'tgan shamolda uchdi; ertalab soatiga 45 km tezlikda shamol bo'roni hudud bo'ylab ko'chib o'tgan.

### *Statika*

Tizimlarni tahlil qilganda, tahlil qilish uchun asosiy asos ko'pincha tizimning jismoniy holati hisoblanadi, ya'ni berilgan "vaqtning oniy tasviri". Statikalar tizimning joriy yo'nalishini tavsiflash uchun ishlatiladi, masalan kattaroq tizim ichida davlat vektori yoki yo'nalishi. Umumiy tizim nuqtayi nazaridan, samolyot ilgichda o'tirish, yo'lakdagi avtomobil, xabar trafigiga ega bo'lmagan tarmoq kompyuter tizimi va yoritish tizimi ON yoki OFF holatida, ularning barchasi statik holatidagi tizimni ifodalaydi. Farqli o'laroq, quyi darajadagi tizim tarkiblar statik holatga ega bo'lishi mumkin, biroq butun tizim esa dinamik holatda bo'ladi.

### *Missiya dinamikasi*

Har bir tabiiy va sun'iy tizim o'z ish muhitida qandaydir shaklda missiyalarni amalga oshiradi. Samolyot km tezlikni kesib o'tgan shamolda uchdi; ertalab 45 km tezlikda shamol bo'roni hudud bo'ylab ko'chib o'tgan bo'ron sifatida boshlandi. Bu dinamik, jismoniy holat. Dinamik - bu tizim statikasining

vaqtga asoslangan tavsifi, uning ish muhitida belgilangan vaqt oralig'i. Dinamika sekin o'zgarishlardan farq qilishi mumkin - tog' yonbag'rida langar o'rnatilgan tosh - mo'tadil o'zgarishlarga, harorat o'zgarishi - shiddatli, o'zgarishlarga, to'satdan o'zgarishlarga esa - zilzilalar yoki vulqonlar kiradi.

Dinamik o'zgarishlar mahalliy yoki global muhitda, kuchlar muvozanatidagi nomuvofiqliklari, tartibsizliklari va beqarorliklari sifatida yuzaga keladi. Insoniyat har doim dinamikani va ularning xulq-atvor modellariga - yer, ob-havo, okeanlar, fond bozori va odamlarga ta'sirini o'rganishga qiziqish bildirgan, ayniqsa, iqtisodiy yoki xavfsizlikka halokatli bo'lishi mumkin bo'lgan dinamik xatti-harakatlar va ta'sirlarni bashorat qilish jarayonida.

Shunday qilib, amaliyot va texnologiya holatining rivojlanishini bashorat qilish katta tijoratdir. Chunki, biz tizim o'zini berilgan dinamik - ish sharoitida qanday tutishi va ishlashini ishonchli tarzda bashorat qila olishimiz kerak.

### *Tizimni barqarorlashtirish*

Barcha tabiiy va sun'iy tizimlar uzoq umr ko'rishni ta'minlash uchun barqarorlik darajasini saqlab turishi kerak. Aks holda, tizim osongina beqaror bo'lib qolishi va o'zi va uning atrofidagi tizimlar uchun xavf tug'dirishi mumkin. Shuning uchun tizimlar o'ziga xos tuzilmaviy xususiyatlariga ega bo'lishi kerak, bu ularga dinamik, tashqi ogohlantirishlarga javoblarini barqarorlashtirish va nazorat qilish imkonini beradi.

Barqarorlik oxir-oqibat, bog'liq va ishonchli bo'lgan kalibrlangan qolipning qandaydir shakliga ega bo'lishiga bog'liq. Sun'iy tizimlar uchun barqarorlik, inertsiyal navigatsiya giroskoplari, global joylashishni aniqlash yo'ldoshlari, elektron soatlar uchun kvarts kristallari va kuchlanish regulyatorlari uchun mos diodlari kabi qurilmalarni qo'llash orqali erishiladi. Ushbu holatlarning har birida tizimni barqarorlashtirish joriy erkin tana dinamikasini o'chash orqali amalga oshiriladi; ularni ma'lum, kalibrlangan qolip manbasi bilan taqqoslash; va har qanday

o'zgarishlarni to'g'irlash uchun tizimning qayta aloqa nazorati harakatlarining tashabbuskori bo'lish kerak.

### **Quvvatni muvozanatlash**

Ushbu elementlarning barchasini hisobga olgan holda, tizim mavjudligi va faoliyatini davom ettirishi uning qobiliyati bilan belgilanadi. Birinchidan, ish muhitining statikasi va dinamikasiga dosh berishi va ikkinchidan, qo'shni tizimlar bilan uyg'unlashadigan qobiliyat va barqarorlik darajasini - kuch muvozanati yoki muvozanat holatini, saqlashi bilan.

### **2.7. Xulosa**

Ushbu bobdagi muhokamamiz tizim atributlari, xossalari va xususiyatlari tushunchasi bilan tanishtirildi. Mavzuga ko'ra, ushbu ma'lumot quyidagilar uchun asos bo'lib xizmat qiladi:

- 1) mavjud tizimning imkoniyatlarini tavsiflash;
- 2) tasniflarni ishlab chiqish yoki baholash uchun dastlabki nazorat ro'yxati tuzish uchun.

### **2.8. Umumiy mashqlar**

1. Kirish qismida ko'rsatilgan ushbu bobdan nimani o'rganishingiz kerak bo'lgan savollarga javob bering.
2. Oldingi bobdagi umumiy mashqlar yoki yangi tizim tanlovi asosida ushbu bobning dolzarb muhokamalaridan olingan bilimlaringizni qo'llang.
3. Tizim missiyasi, maqsadlari va vazifalari orqali tanlangan tizimning noyob aniqlovchisiga misollar keltiring.
4. Missiya amallari atributlarining turlariga misollar keltiring.
5. Umumiy, amaliy, jismoniy va tizimli estetik xususiyatlarga misollar keltiring.
6. Tashkilotingiz taqdim etadigan har qanday tizim, mahsulot yoki xizmatidan foydalanib, inson tizimining rollarini aniqlang.

### **2.9. Nazorat savollari**

1. Tizim atributi nima?
2. Tizim xossasi nima?
3. Tizimning tavsifi nima?
4. Tizim, mahsulot yoki xizmatning o'ziga xos xususiyati nimada?
5. Tizim, mahsulot yoki xizmat ko'rsatish toifalari nimalarni izohlaydi?
6. Tizim xususiyatlarining qanday turlari mavjud?
7. Tizimning muvozanat holati nimadan iborat?

### III- BOB. TIZIMDAGI ROLLAR VA MANFAATDOR TOMONLAR

Faoliyat yuritayotgan har bir tizimning maqsadi yoki vazifasi (missiyasi) bor. Boshqacha qilib aytganda, har bir tizimning mavjudligi boshqa tizimlarga nisbatan mavjudligining sababi bor.

Inson tomonidan yaratilgan tizimlarning aksariyati tashkilot "egasi"ning rollari, missiyalari va maqsadlariga hissa qo'shish va ularga erishishga qaratilgan. Har bir tizim tizim egasining umumiy rolida o'z rolini o'ynaydi. Ushbu bobda tizim rollari va manfaatdor tomonlar tushunchasini tahlil qilamiz. Biz tizimlardan foydalanadigan tashkilotlarning rolini, mahsulotlar va xizmatlardan, hamda jismoniy tizimlarning rollari, vazifalari va maqsadlariga erishish uchun qanday qilib qiymatga ega bo'lgan narsalarni sotib olinishi bo'yicha tadqiqot o'tkazamiz.

Inson tomonidan yaratilgan tizimlarning tashkilotda muvaffaqiyatga erishishi uchun, tizim aniqlashtirilgan, loyihalashtirilgan, ishlab chiqilgan, birlashtirilgan, tekshirilgan, faoliyat yuritayotganligi va qo'llab-quvvatlanishi bilan belgilanadi. Bu to'g'ridan-to'g'ri yoki bilvosita tizimda faoliyat yuritayotgan insonlardan missiya natijalari, erishilayotgan amaliy samaradorligidan manfaatdor bo'lishini talab qiladi. Biz bunday shaxslarni *tizimdan manfaatdor tomonlar* deb ataymiz. Biz ushbu bobni tizimdagi manfaatdor tomonlarning asosiy rollarini va ularning tizim missiyasi va natijalariga ba'zan ijobiy, ba'zan salbiy hissalarini aniqlash orqali yakunlaymiz.

#### 3.1. Asosiy atamalarning ta'riflari

*Yakuniy foydalanuvchi* – natija yoki natijalardan bevosita

foйда ko'radigan shaxs yoki tashkilot tizim, mahsulot yoki xizmat

*Manfaatdor shaxs* – o'ziga yuklangan vazifani bajarishda tizim tomonidan ishlab chiqarilgan natijalardan (masalan, do'stona, raqobatbardosh yoki qarama-qarshi) manfaatdor shaxs yoki tashkilot.

*Tizim foydalanuvchisi* – tizim, mahsulot yoki xizmatdan foydalanadigan jismoniy shaxs yoki tashkilot yoki missiyaga yo'naltirilgan maqsad yoki vazifani bajarish uchun ularning qo'shimcha mahsuloti. Masalan, shahar transport tizimida odamlarni bir joydan boshqasiga tashish uchun avtobus haydovchilari va avtobuslar ishlaydi. Avtobus haydovchilari va yo'lovchilari foydalanuvchilar sifatida tasniflanadi.

#### 3.2. Tashkilot tizimidagi rollar

Tashkilot tizimlari o'zlarining nizomidagi missiyalari va maqsadlarini aks ettiruvchi rollarni bajaruvchi tijorat domenlaridan iborat bo'ladi. 3.1- jadvalda tizim rollari va vazifalariga misollar keltirilgan.

3.1- jadval. Tizimdagi rollar va vazifalariga misol

№	Tizim roli	Rolga asoslangan missiya
1	Qonunchilik	Boshqa shaxslar, tashkilotlar yoki shaharlar yoki viloyatlar kabi davlat tuzilmalarini boshqaradigan qonunlar, nizomlar, qoidalar, qarorlar, siyosatlar va boshqalar ko'rinishidagi ijtimoiy muvofiqlik bo'yicha ko'rsatmalar va cheklovlarni o'rnatish.
2	Sud	Shaxsiy, tashkilot yoki korxonadan tomonidan belgilangan qonunlar, nizomlar, qoidalar, qarorlar, siyosatlar va boshqalarga muvofiqligini baholash.
3	Harbiy	Mamlakat himoyasini ta'minlaydigan va uning konstitutsiyasi, xavfsizligi va

		mustaqilligini himoya qiladigan hamkorlik, favqulodda vaziyatlar, tinchlikni saqlash, oldini olish va urush davridagi rollarni bajarish.
4	Transport	Mijozlarga va mahsulotlarga quruqlik, dengiz, havo yoki kosmos yoki ularning kombinatsiyasi orqali bir joydan ikkinchi joyga xavfsiz va samarali o'tish imkonini beradigan transport xizmatlarini taqdim etish.
5	Fuqarolik	Tashkilotning maqsad va vazifalarini amalga oshiradigan davlat xizmatlarini ko'rsatish.
6	Ta'lim	Odamlarga maxsus bilimlarga ega bo'lishlari va ularni jamiyatga hissa qo'shadigan a'zolar bo'lishga tayyorlash uchun ko'nikmalarini oshirishlari uchun ta'lim imkoniyatlarini ta'minlash.
7	Manba (zaxira)	Investitsiya daromadini kutish bilan jismoniy shaxs, tashkilot yoki korxonaning missiyalari, maqsadlari va vazifalarini qo'llab-quvvatlash uchun ishlash va xavfga mos keladigan zaxiralarni - vaqt, pul, yoqilg'i, elektr energiyasi va boshqalarni taqdim etish.
8	Monitor nazorati	Boshqa tizimlarning ishlashini kuzatib boring, belgilangan standartlarga muvofiq ish faoliyatini baholang, ob'yektiv dalillarni yozing va ish faoliyatini yoki boshqa tizimlarni nazorat qilish.
9	Tadqiqot va ishlanma	Tadqiqot va ishlab chiqish, ishlab chiqarish yoki tizimlar uchun yangi texnologiyalarni qo'llashni o'rganish.

10	Режиссер	Bozor talablari va standartlariga muvofiq tizim tuzilmasini katta yoki ommaviy hajmda ishlab chiqarish.
11	Konstruksiya	Tizim ishlab chiquvchilariga foydalanuvchilarga tizimlarni o'rnatish, ishlatish, qo'llab-quvvatlash, o'qitish va yo'q qilish imkonini beruvchi ob'yektlar, saytlar va boshqalarni amalga oshirish imkonini beruvchi qurilish xizmatlarini taqdim etish.
12	Qishloq xo'jaligi	Bozorga odamlar va hayvonlar iste'moli uchun xavfsiz va atrof-muhit uchun xavfsiz oziq-ovqat va qishloq xo'jaligi mahsulotlarini taqdim etish.
13	Chakana yoki ulgurji savdo	Bozorga iste'mol tovarlari va xizmatlarini yetkazib berish.
14	Tibbiyot	Tibbiy maslahat va davolash xizmatlarini taqdim etish.

### 3.3. Tizimdagi rolning mazmuni

Har bir texnogen tizim o'z mijozlarini, foydalanuvchilar va manfaatdor tomonlarni qo'llab-quvvatlash uchun ishlashga asoslangan aniq maqsadlar va natijalarga erishish uchun mo'ljallangan missiyaga yo'naltirilgan amallar va vazifalarni bajaradi. Ushbu missiyalarni bajarish ikki turdagi tezkor rollarni talab qiladi: missiya tizimi va qo'llab-quvvatlash tizimi.

#### *Tizimdan manfaatdorlikning mohiyati.*

Agar biz inson tomonidan yaratilgan tizimlarni tahlil qilsak, har bir tizim bir vaqtning o'zida ikki xil ma'nodagi rolni bajarishini aniqlaysiz – birinchidan, tizimdagi missiyaning roli va ikkinchidan tizimdagi yordamchining roli. Keling, bu rollarning har birini aniqlaylik.

**Tizimdagi missiyaning roli.** Tizimdagi missiyaning rollari samaradorlikka asoslangan natija maqsadlariga erishadigan mahsulot va xizmatlarni yetkazib berish uchun maxsus vazifalar yuklangan tizimlar tomonidan amalga oshiriladi. Quyidagi misolni ko'rib chiqamiz.

**1- misol.**

Xalqaro kosmik hamjamiyatga tegishli bo'lgan kosmik kemalar, turli ilmiy maqsadlarni amalga oshirish uchun kosmik missiyalarni bajaradi.

Shuni ta'kidlash kerakki, foydalanuvchi tizim missiyalariga ega bo'lishi yoki bo'lmasligi mumkin. Kosmik hamjamiyat ba'zi hollarda shunday qiladi; boshqa hollarda, ular tizimlarni ijaraga berish uchun boshqa tashkilotlar bilan shartnoma tuzishlari mumkin. Masalan, yuk tashuvchi kompaniya tijoratning ko'tarilishi paytida tashkiliy missiyalarni bajarish uchun qo'shimcha transport vositalarini ijaraga olishi mumkin.

**Tizimdagi yordamchining roli.** Tizimdagi yordamchining rollari, tizimdagi missiyalarning tayinlangan vazifalarni bajarishi va qo'llab-quvvatlashi, hamda keyingi texnik xizmat ko'rsatish va o'qitishni amalga oshirishga tezkor tayyor bo'lishini ta'minlashi uchun, boshqa tizimlar tomonidan bajariladi. Quyidagi misollarni ko'rib chiqamiz.

**2- misol.**

Aviakompaniya tizimining missiyasi, shaharlar o'rtasida yo'lovchilar va yuklarni xavfsiz va qulay tashishni maqsad qilgan samolyot va ekipajdan iborat. Samolyotni ta'minlash tizimi yuk tashish xodimlari, mexaniklar, jihozlar, yer bilan aloqa uskunalari va samolyotni xavfsiz parvozga tayyorlash, sarflanadigan narsalarni to'ldirish, yuk va yo'lovchi yuklarini yuklash/tushirishdan iborat.

**3- misol.**

Xalqaro kosmik hamjamiyatga tegishli bo'lgan kosmik missiyalarni bajarish uchun tizimdagi missiya o'z rolini bajaradi. Xalqaro kosmik stansiyani qo'llab-quvvatlash tizimi sifatida Space

Shuttle va Soyuz astronaut va kosmonavtlari, yuklarni esa Kennedi kosmik markazi va Xalqaro kosmik hamjamiyat o'rtasida kosmosga olib chiqiladi va yerga olib tushiladi.

### 3.4. Tizim rollarining tizim muhandisligi bilan bog'liqligi

Qanday tashkiliy rollar tizim muhandisligi bilan bog'liqligini tushunishimiz kerak. Uskuna, dasturiy ta'minot va o'quv dasturlar kabi jismoniy tizimlar mavjud, chunki yuqori darajadagi tashkilotlar foydalanuvchilar sifatida byudjet xarajatlari va jadval cheklolari doirasida tashkiliy maqsadlar, missiyalar va maqsadlarga erishish uchun jismoniy tizim imkoniyatlaridan foydalanadilar. Tizim muhandisi sifatida siz, foydalanuvchi qanday niyat qilayotganini to'liq tushunishingiz va u bilan qadrlashingiz kerak. Tashkilotning maqsadlari, vazifalari va vazifalariga erishish uchun belgilangan amaliy muhitda tizimni joylashtirish va ishlatish kerak.

Masalan, ushbu ko'rsatma tashkilotning transport roliga qanday aloqasi bor? Agar siz aviakompaniya tijoratida ishlasangiz, siz bron qilish va chiptalar sotish, ro'yxatdan o'tish va bagajni qayta ishlash, samolyotlar, darvozalar, maxsus xizmatlar yoki xavfsizlik xizmatlarini taqdim etishingiz yoki shartnoma tuzishingiz kerak. Ushbu ob'yektlarning barchasi tashkilot rolini bajarish uchun jismoniy tizimlarni, shuningdek, asbobiy va dasturiy ta'minotni talab qiladi.

Tashkilot, birinchidan tizim, mahsulot yoki xizmatni ishlab chiqishi; ikkinchidan, tizim, mahsulot yoki xizmatni boshqa sotuvchilardan xarid qilishi; uchinchidan, tizimlar yoki xizmatlar uchun shartnoma tuzishi va ijaraga berishi mumkin.

Har qanday holatda ham tashkilotning transport roliga erishishga hissa qo'shadigan, har bir tizim ob'yektiga maqsadlar, vazifalar yoki ishlash talablari ajratiladi. Tashkilot, masalan, aviakompaniya, amallarni boshlaganida, rejalashtirish, amalga oshirish va foydalanish faoliyatining ajralmas qismini ta'minlash

uchun, ko'p sonli odamlar kerak. Har bir manfaatdor tomon tashkilotning roli va undagi joylangan tizimlarning natijalari va muvaffaqiyatlaridan manfaatdordir.

### 3.5. Tizimdan manfaatkor shaxslarning roli nimadan iborat?

Inson tomonidan yaratilgan tizimlar, konsepsiyadan tortib to yo'q qilishgacha, to'g'ridan-to'g'ri va bilvosita inson yordamini talab qiladi. Tizim, mahsulot yoki xizmatdan manfaatdor odamlar har bir tizimni konsepsiyalash, moliyalashtirish, xarid qilish, loyihalash, ishlab chiqish, integratsiyalashuvi, ishlashi, qo'llab-quvvatlashi va ishdan chiqishiga hissa qo'shishni kutishadi. Biz bu odamlarni *manfaatdor tomonlar*, deb ataymiz. Tizimning hajmi va murakkabligiga, shu jumladan foydalanuvchi uchun xavf va ahamiyatiga qarab, manfaatdor tomonlar rolini jismoniy shaxs, tashkilot yoki korporatsiya, hukumat yoki mamlakat kabi yuqori darajadagi korxonalar bajarishi mumkin. Manfaatdor tomonlar kimligini yaxshiroq tushunish uchun keling, ushbu atamaning qo'llanilishini qamrab olamiz va birlashtiramiz.

*Manfaatdor tomon* – bu tizim va uning natijalaridan manfaatdor bo'lgan har qanday shaxs yoki tashkilotdir.

### 3.6. Tizimdagi manfaatdor tomonlarning rollari

Inson tomonidan yaratilgan har bir tizim turli maqsadlar va kun tartibiga ega bo'lgan inson rollari bilan qo'llab-quvvatlanadi, bu tizimning butun hayot sikli davomida uzoq faoliyat yuritishi va me'yoriy ishlashiga yordam beradi. Tizimdagi rollarga quyidagi misollarni keltiramiz.

#### 4- misol.

- ❖ Tizim himoyachisi yoki tarafdori
- ❖ Tizim aksiyadori
- ❖ Tizim egasi
- ❖ Tizim foydalanuvchilari
- ❖ Tizim arxitektori

- ❖ Tizim buyurtmachisi
- ❖ Tizimni ishlab chiqaruvchi
- ❖ Xizmat ko'rsatuvchi – rejissyor
- ❖ Mustaqil test agentligi
- ❖ Tizim administratori
- ❖ Missiyani rejalashtiruvchi
- ❖ Tizim tahlilchisi
- ❖ Tizimni qo'llab-quvvatlash
- ❖ Tizim ta'minotchisi
- ❖ Tizim instruktorigi
- ❖ Tizim tanqidchisi
- ❖ Tizim raqobatchisi
- ❖ Tizim raqibi
- ❖ Tizim tahdidi

Ushbu rollarning har biri bilan tanishamiz va aniqlaymiz. 3.2-jadvalda har bir manfaatdor tomon rolining qisqacha tavsifi keltirilgan.

3.2- jadval. Tizimdagi manfaatdor tomonlar rolining tavsiflari

№	Rol	Rolning tavsifi
1	Tizim himoyachisi yoki tarafdori	Tizimning sababi, vazifasi yoki mavjudligi sababini himoya qiluvchi shaxs, tashkilot yoki korxonalar. Tizim himoyachilari tizimni qo'llab-quvvatlashdan moddiy yoki nomoddiy foyda olishlari mumkin yoki ular tizim o'zlari qo'llab-quvvatlagan yuqori darajadagi sabablarga hissa qo'shayotganiga ishonishlari mumkin.
2	Tizim aksiyadori	Tizim va uning rivojlanishidagi to'liq yoki aksiyadorlik ulushlariga to'g'ridan-to'g'ri yoki bilvosita "egalik qiluvchi" jismoniy shaxs, tashkilot yoki korxonalar, amal, mahsulotlar va yon mahsulotlar.
3	Tizim egasi	Tizim, uning rivojlanishi, ishlashi, mahsulotlari, qo'shimcha mahsulotlar, natijalar va yo'q qilinishi uchun qonuniy

		va ma'muriy javobgarlik va javobgar bo'lgan jismoniy shaxs, tashkilot yoki korxon.
4	Tizim foydalanuvchilari	Foydalanuvchilar tizimni jismonan boshqarishi yoki tizimga kiruvchi berilganlar, materiallar, xom ashyo, oldindan ishlov berilgan materiallar va boshqalarni taqdim etishi va qo'shimcha qiymat qo'shilgan mahsulot, xizmatlar yoki ma'lumotlarni kutishi mumkin. Foydalanuvchilar to'g'ridan-to'g'ri yoki bilvosita tizim himoyachisi, tizim egasi yoki boshqa foydalanuvchilarni tizim ichiga aralashishiga yol ruxsat berishi mumkin.
5	Tizim arxitektori	Tizim, tizim g'oyalari, vazifalari, maqsadlari va vazifalarini namoyish qilish, konseptuallashtiradigan va shakllantiradigan shaxs, tashkilot yoki korxon. Tizim muhamdisligi ko'p tarmoqli bo'lgani sababli, tizim arxitektori roli asbob arxitektorlari, dasturiy ta'minot arxitektorlari, o'quv arxitektorlar va boshqalar orqali namoyon bo'ladi.
6	Tizim buyortmachisi	Quyidagilar uchun texnik vakil sifatida xizmat qilishi uchun, foydalanuvchi tomonidan tanlangan agent (yoki agentlik) quyidagi vazifalarni amalga oshirishi kerak: 1. Tizimni belgilash. 2. Tizimning yaratuvchisini yoki xiz-

		matlar provayderini tanlash. 3. Texnik yordam ko'rsatish. 4. Shartnomaning bajarilishi va tekshirilgan va tasdiqlangan tizimni foydalanuvchiga etkazib berish bo'yicha shartnoma nazoratini ta'minlash.
7	Tizimni ishlab chiqaruvchi	Tizim samaradorligi tasnifsida chegalangan va belgilangan amaliy imkoniyatlar va samaradorlik asosida tasdiqlangan tizim yechimini ishlab chiqish uchun mas'ul shaxs, tashkilot yoki korxon.
8	Xizmat ko'rsatuvchi – rejissyor	Tizimni boshqarish yoki uning ishlashini qo'llab-quvvatlash bo'yicha xizmatlar ko'rsatish uchun ijaraga olingan yoki shartnoma tuzgan jismoniy shaxs, tashkilot yoki korxon.
9	Mustaqil test agentligi	Tizim foydalanuvchining mo'ljallangan va belgilangan amaliy muhit uchun hujjatlashtirilgan amaliy missiyasi ehtiyojlarini qondirishini tekshirish va/yoki tasdiqlash uchun mas'ul shaxs, tashkilot yoki korxon.
10	Tizim administratori	Tizimning umumiy ishlashi, konfiguratsiyasi, kirish va texnik xizmat ko'rsatish uchun javobgar shaxs, tashkilot yoki korxon.
11	Missiyani rejalashtiruvchi	Jismoniy shaxs, tashkilot yoki korxon: 1. Vaziyat tahlili asosida missiya maqsadlarini batafsil taktik amalga oshirish rejalariga aylantiradi; kuchli, zaif tomonlar, imkoniyatlar va tahdid-

		larga nisbatan tizim imkoniyatlari va ishlashini ta'minlash. 2. Missiya va uning maqsadlariga erishish uchun harakat yo'nalishini, qarshi choralarini va zarur zaxiralarni ishlab chiqish.
12	Tizim tahlilchisi	Missiyani rejalashtiruvchilar, tizim operatorlari va tizim boshqaruvchilari tomonidan asosli qaror qabul qilishni qo'llab-quvvatlash uchun mazmunli ma'lumotlarni taqdim etish uchun tahliliy usullar va usullarni (ilmiy, matematik, statistik, moliyaviy, siyosiy, ijtimoiy, madaniy va boshqalar) qo'llaydigan shaxs yoki tashkilot.
13	Tizimni qo'llab-quvvatlash	Tizimni, uning imkoniyatlarini va/yoki tizimning missiyasi va maqsadlariga muvaffaqiyatli erishishni ta'minlaydigan barqarorlik darajasida ishlashini qo'llab-quvvatlash uchun mas'ul bo'lgan shaxs, tashkilot yoki korxonalar. Tizimni qo'llab-quvvatlash texnik xizmat ko'rsatish, o'qitish, ma'lumotlar, texnik qo'llanmalar, zaxiralar va boshqaruv kabi tadbirlarni o'z ichiga oladi.
14	Tizim ta'minotchisi	Jismoniy shaxs, tashkilot yoki korxonalar profilaktika va tuzatuvchi texnik xizmat ko'rsatish, tizimni yangilash va h.k.lar orqali yuklab tizimni elementining to'g'ri ishlashini ta'minlash uchun javobgardir.
		Tizim elementining barcha a'zolarini tizim vazifasi va uning maqsadlariga

15	Tizim instruktori	erishishda ishlash standartiga asoslangan malaka darajasiga erishish uchun o'rgatish uchun javobgar shaxs yoki tashkilot.
16	Tizim tanqidchisi	Raqobatchi, qarama-qarshilik yoki dushmanlik motivlariga ega bo'lgan shaxs, tashkilot yoki korxonalar o'ziga yuklangan vazifalar, maqsadlar va vazifalarni iqtisodiy jihatdan samarali, qo'shimcha qiymatli tarzda bajarish uchun tizimning kamchiliklarini e'lon qiluvchi va/yoki tizim tahdid ostida, deb hisoblovchi. Tizim tanqidchisi boshqa tizimda tizim himoyachisi bo'lib xizmat qilishi mumkin.
17	Tizim raqobatchisi	Missiyalari, maqsadlari va vazifalari o'xshash missiya natijalarini qo'lga kiritish uchun raqib bilan raqobatlashadigan shaxs, tashkilot yoki korxonalar.
18	Tizim raqibi	Manfaatlari, mafkurasi, maqsadlari va quyidagilardan iborat dushman shaxs, tashkilot yoki korxonalar: 1. Boshqa tizimning vazifalari, maqsadlari va/yoki maqsadlariga qarshi turish. 2. Xulq-atvor namunalari va tahdidli ko'rinadigan harakatlarni namoyish etish.
19	Tizim tahdidi	Raqobatbardosh, raqib yoki dushman shaxs, tashkilot yoki korxonalar boshqa tizimning vazifalari, maqsadlari va/yoki maqsadlariga zid bo'lgan missiyalar,

	maqsadlar va vazifalarni faol ravishda rejalashtiruvchi va/yoki amalga oshiruvchi.
--	--

### **Tizimdagi manfaatdor tomonlarning rollarining ko'pligini tushunish.**

Manfaatdor tomonlarning roli konteksti qiziqishlar tizimiga asoslanadi. Ushbu rollarni bajaradigan shaxs, tashkilotlar yoki korxonalar boshqa tizimlarda manfaatdor tomonda bo'lishi mumkin. Aslida, manfaatdor tomonlarning roli tizimdan tizimga farq qilishi mumkin.

#### **5- misol.**

Tizimingiz uchun tizim advokati ko'plab tizimlar uchun tizim egasi bo'lib xizmat qilishi mumkin. Foydalanuvchilar o'z vazifalari va maqsadlariga erishish uchun bir nechta boshqa tizimlardan foydalanishlari mumkin. Tizim ishlab chiquvchisi ko'plab tizimlarning ishlab chiquvchisi va boshqa tizimlarning tizim boshqaruvchisi bo'lishi mumkin.

### **3.5. Xulosa**

Bizning muhokamamiz tizimdagi manfaatdor tomonlarning rollarini qamrab oldi. Har bir manfaatdor tomon tizim, mahsulot yoki xizmatning natijasi va muvaffaqiyatiga ma'lum darajada qiziqish bildiradi. Ushbu manfaatlar asosida tizim manfaatdor tomonlari tizim talablari uchun asosiy manbalarga aylanadi. Tizimning ishlashidan manfaatdor tomonlarning qoniqishi oxir-oqibat tizimning maqbulligini belgilaydi.

### **3.6. Umumiy mashqlar**

1. Kirish qismida ko'rsatilgan ushbu bobdan nimani o'rganishingiz kerak bo'lgan savollarga javob bering.

2. 2-bobda aniqlangan tizimlar ro'yxatiga qarang. Oldingi bobning Umumiy mashqlari yoki yangi tizim tanlovidan olingan tanlov asosida ushbu bobning dolzarb muhokamalaridan olingan bilimlaringizni qo'llang. Quyidagilarni aniq belgilang:
  - a. Tizimdagi qanday missiya va yordamchi rollari mavjud?
  - b. Har bir tizim uchun manfaatdor tomonlar kimlar?
3. Tashkilot ichida shartnoma dasturini aniqlang.
4. Nazorat ro'yxati sifatida 3.2- jadvaldan foydalanib, tizimdagi rollar va manfaatdor tomonlarini aniqlash uchun dastur va texnik direktorlar bilan suhbatlashing.
5. Tashkilot ichida aloqa, buxgalteriya hisobi yoki shartnomalar kabi xizmatlar guruhini aniqlang.
6. Nazorat ro'yxati sifatida 3.2- jadvaldan foydalangan holda xizmatlar tizimining roli va manfaatdor tomonlarni aniqlang.
7. Tashkilot taqdim etadigan har qanday tizim, mahsulot yoki xizmatdan foydalanib, inson tizimi qanday rol o'ynashini aniqlang.

### **3.7. Nazorat savollari**

1. Tizimdagi rol nima?
2. Tizimdagi missiyasi nima?
3. Tizimdagi yordamchi nima?
4. Manfaatdor shaxs kim?
5. Turli manfaatdor tomonlar kimlar?
6. Tizim rejissyorining vazifasi nimadan iborat?
7. Tizim administratorining vazifasi nimadan iborat?
8. Tizim tahlilchisining vazifasi nimadan iborat?
9. Tizimni qo'llab-quvvatlovchilar, deganda namalar nazarda tutiladi?
10. Tizimni qo'llab-quvvatlashda qanday tadbirlar nazarda tutiladi?
11. Tizim ta'minotchisining vazifasi nimadan iborat?
12. Tizim instruktorining vazifasi nimadan iborat?

13. Tizim tanqidchisining vazifasi nimadan iborat?
14. Tizim raqobatchisi deganda nimani tushunasiz?
15. Tizim tahdidi deganda nimani tushunasiz?

#### IV- BOB. TIZIM AMALLARINING MODELI

Missiya tahlili tizimdan foydalanish holatlari va ssenariylarini aniqlaganligi sababli, eng yaxshi foydalanuvchi bilan hamkorlikda, tizimni qanday joylashtirish, ishlatish, qo'llab-quvvatlash va yo'q qilish niyatida ekanligini konseptualashtirish uchun ishlaydi. Konseptualizatsiyani hujjatlashtirish mexanizmlaridan biri bu amallarning tizim konsepsiyasi yoki amallar konsepsiyasidan iborat. Ushbu bo'lim amallar konsepsiyasini ishlab chiqish uchun tizimli asosni ta'minlovchi tizim amallarining modeli bilan tanishtiriladi.

Tizim amallari modeli yuqori darajadagi amaliy ish jarayonini ta'minlaydi va tizimni quyidagicha tavsiflaydi:

- missiya uchun tuziladi;
- missiyani amalga oshiradi;
- faoliyati davomida qo'llab-quvvatlaydi.

Modelning tuzilishi tafsiri talablariga yoki tizimni loyihalashtirish yechimi uchun ish jarayoniga aylantirilishi mumkin bo'lgan amallar va vazifalardan iborat. Bizning mulozaralarimiz modelning amaliy imkoniyatlari qanday taqsimlangani va tizim elementlariga, masalan, uskuna, xodimlar va ob'ektlarga o'tishi haqida tushuncha beradi. Natijada, ushbu muhokamalar, tizim vazifasi va qo'llab-quvvatlash amallari uchun asos bo'lib xizmat qiladi.

##### 4.1. Asosiy atamalarining ta'riflari

*Amallar g'oyasi* – bu missiyasi oldidan, topshiriqdan va keyingi bosqich natijalariga asoslangan ishlash maqsadlariga erishish uchun zarur bo'lgan tizimning ketma-ket va/yoki parallel amallardagi ish jarayonining tavsifi.

*Nazorat yoki bosqichma-bosqich punkt* – bu, to'g'ri qaror qabul qilish mezonlari to'plami bajarilmaguncha, ish jarayonining borishini keyingi maqsadli amallar to'plamiga o'tkazishni cheklab qo'yadigan asosiy qaror nuqtasi.

Tizim amallari ko'p darajali va o'zaro bog'liq harakatlar yoki vazifalar to'plami bo'lib, ular birgalikda missiya, topshiriq yoki topshiriqdan keyingi bosqich maqsadini qondirishga yordam beradi.

*Tizim amallarining lug'ati* – bu tizim ob'yektining amaliy aloqalari va muayyan bosqich va ish tartibotini qo'llab-quvvatlash uchun zarur bo'lgan o'zaro ta'sirlarni qamrab oladigan va tavsiflovchi hujjatdir. Bunda, amaliy munosabatlar va o'zaro ta'sirlar tahlil qilinadi va zarur amaliy qobiliyatlar to'plamga tarjima qilinadi, keyinchalik ular har bir ish tartibotini qo'llab-quvvatlash uchun tizimning ishlash talablariga aylantiriladi.

*Tizim amallarining modeli* – bu ko'pgina tizimlar uchun ish jarayoni va amallarni aniqlash uchun boshlang'ich nuqta sifatida ishlatilishi mumkin bo'lgan tizim amallarining umumlashtirilishi-dan iborat tuzilma.

#### 4.2. Amaliyotlar tizimi konsepsiyasi

Tizimning muammoli maydoni va yechim bo'shliqlari chegalangandan so'ng, keyingi qadam foydalanuvchining yechim maydoni tizimidan qanday foydalanish niyatida ekanligini tushunishdir. Aksariyat tizimlar oldingi va oddiygina qo'llaniladigan amallar, ob'yektlar va ko'nikmalarning mavjud infratuzilmasi asosida qurish uchun yangi texnologiyadir. Biroq, bu misli ko'rilmagan tizimlar paydo bo'lmaydi, degani emas.

Murojaat oldidan va misli ko'rilmagan tizimlar haqida qo'shimcha ma'lumot olish uchun ushbu tizimlarning ta'rifi bo'yicha 3-bobga qarang. Muammoni hal qilish fazosi konsepsiyasini kengaytirsak, bizning tahlilimiz shuni ko'rsatadiki, yechim fazoviy vaqtga asoslangan o'zaro ta'sirlar, ya'ni ob'yektlar munosabatlari – tizim amallari modeliga umumlashtirilishi mumkin bo'lgan amallar

to'plami bilan tavsiflanishi mumkin. O'z navbatida, model tizimning vazifasini bajarish uchun zarur bo'lgan yuqori darajadagi ketma-ket va bir vaqtning o'zida bajariladigan amallarni tavsiflovchi amallar konsepsiyasini ishlab chiqish uchun asos yaratadi.

*1- misol.* Space Shuttle tizimi kabi tizim uchun amaliy tizim konsepsiyasi foydali yukni koinotga yetkazish, foydali yukni joylashtirish va tajribalar o'tkazish, yuk va kosmonavtlarni yerga xavfsiz tarzda qaytarish uchun zarur bo'lgan amaliy ketma-ketliklarni tavsiflaydi.

Biz 1- misolga tizim/mahsulotning hayot aylanishi doirasidagi siklik amallarni bajaruvchisi sifatida murojaat qilamiz. Tirik organizmlar, masalan, odamlar, oziq-ovqat, suv, dam olish va uyquga bo'lgan kundalik ehtiyojimizdan dalolat beruvchi bu siklik xususiyatni namoyon qiladi. Biz amallar konsepsiyasini foydalanuvchi tizimdan qanday foydalanishni rejalashtirayotganini aks ettiruvchi umumiy maqsadlar to'plami nuqtayi nazaridan umumlashtirishimiz mumkin. Bu maqsadlarga quyidagilar kiradi:

1. Tizimni ishga tushirish.
  2. Tizimni joylashtirish va amaliy foydalanish uchun sozlash.
  3. Tizimning missiyasi oldidan, missiya vaqtida va missiyadan keyingi davrdagi amallarni bajarishga tayyorligini tekshirish.
  4. Tizim faoliyatidan foydalanish.
  5. Tizimni tozalash va keyingi foydalanish uchun saqlash.
  6. Zarur bo'lganda tizimni tashlash.
- Ushbu maqsadlarni umumiy tizim amallari yechimiga qanday integratsiyalash mumkinligini yaxshiroq tushunish uchun, tizimli amallar modeli tavsifini ko'rib chiqamiiz.

#### 4.3. Tizim amallarining modeli

4.1- rasmda tasvirlangan tizim amallarining modeli mutahassislar yaratgan tizimlariga qo'llanilishi mumkin bo'lgan konstruksiyani taqdim etadi. Ushbu grafikda bir nechta variantlar mavjud

bo'lsa-da, uning qanday ishlashini yaxshiroq tushunish uchun ushbu modelni ko'rib chiqamiz.

Birinchidan, grafikaning umumiy mazmuni haqida tushuncha.

4.1- rasmdagi har bir quti umumiy missiya maqsadiga erishish uchun zarur bo'lgan tizimdan foydalanish holatlariga asoslangan imkoniyatlar va tadbirlarning birlashtirilgan, ko'p darajali to'plamini ifodalaydi. Biz ushbu imkoniyatlarning har birini, quyi darajadagi amaliy imkoniyatlarga kengaytirishimiz yoki parchalashimiz mumkin. Oxir-oqibat, bu qobiliyatlar va ularning tegishli ishlash darajalari tizimning bir yoki bir nechta elementlariga, masalan, kodlar, uskuna va ob'ektlarga taqsimlanadi.

Har bir qaror bloki *nazorat yoki bosqich punkti* deb ataladi va oldindan belgilangan chiqish yoki kirish mezonlari to'plamiga asoslangan qaror qabul qiluvchi bo'limdan ruxsat berilmagan qarorni talab qiladi. Har bir amal va nazorat nuqtasi noyob belgilovchi bilan belgilanadi. Belgilovchi tizim ishlashining tafsirini ma'lum bir talablar bo'limiga yoki amaliy konsepsiya tavsifidagi batafsil rivoyatga solishtirish uchun ishlatiladi.

Amallar konsepsiyasi va amaliy konsepsiya tavsifidan foydalanishni kuzating. Turli tashkilotlar u yoki bu atamani ishlatadilar. Ba'zi odamlar uchun amallar konsepsiyasi tizimi qanday ishlashi haqida qisqacha munozarani keltirib chiqaradi, amaliy konsepsiya tavsifi esa amallar konsepsiya uchun qo'llab-quvvatlovchi tafsilotni aniqlaydi.

Izoh: Bitta yoki bir nechta atamani tanlang va uni doimiy ravishda dasturlarda qo'llang.

#### 4.4. Tizim amallari modelining tavsifi

4.1- rasmda ko'pchilik mutaxassislar tomonidan yaratilgan tizimlar uchun qo'llaniladigan tizim amallarining modeli tasvirlangan. Modelga kirish, tizim/mahsulot hayot siklining tizimni yaratish bosqichidan (1) o'tgandan so'ng boshlanadi. Kirish mezonlari (2) tizimning faol xizmatini boshlashga tayyorligini

baholash uchun islatiladi. Biz shu tizimdagi umumlashtirilgan amallar konsepsiyasining modelini tahlil qilib chiqamiz.

#### 3- amal. Tizimni joylash.

Tizimni joylash bo'yicha 3- amal, tizimni foydalanuvchining kerakli manziliga yetkazib berish va o'rnatish uchun zarur bo'lgan tizim imkoniyatlari va faoliyatiga murojaat qiladi. Har bir tizim ishlab chiqarish chizig'idan chiqib ketganda va ishlash talablariga muvofiqligi tekshirilganda, tizim jamlanadi va foydalanuvchiga joylashtirish yoki tarqatish uchun jo'natiladi. Bu faoliyatga – (shartli ravishda) transport, yuklash/tushirish, paketni ochish, dastlabki sozlash, o'rnatish va yig'ish, tizimni tekshirish, yuqori darajadagi tizimlarga integratsiyani tekshirish va ushbu darajadagi o'zaro muvofiqlikni tekshirish amallari kiradi.

Rejalashtirilgan barcha tadbirlar tugagandan so'ng, 4- amal bajariladi.

#### 4- amal. Tizim missiyani o'qitish.

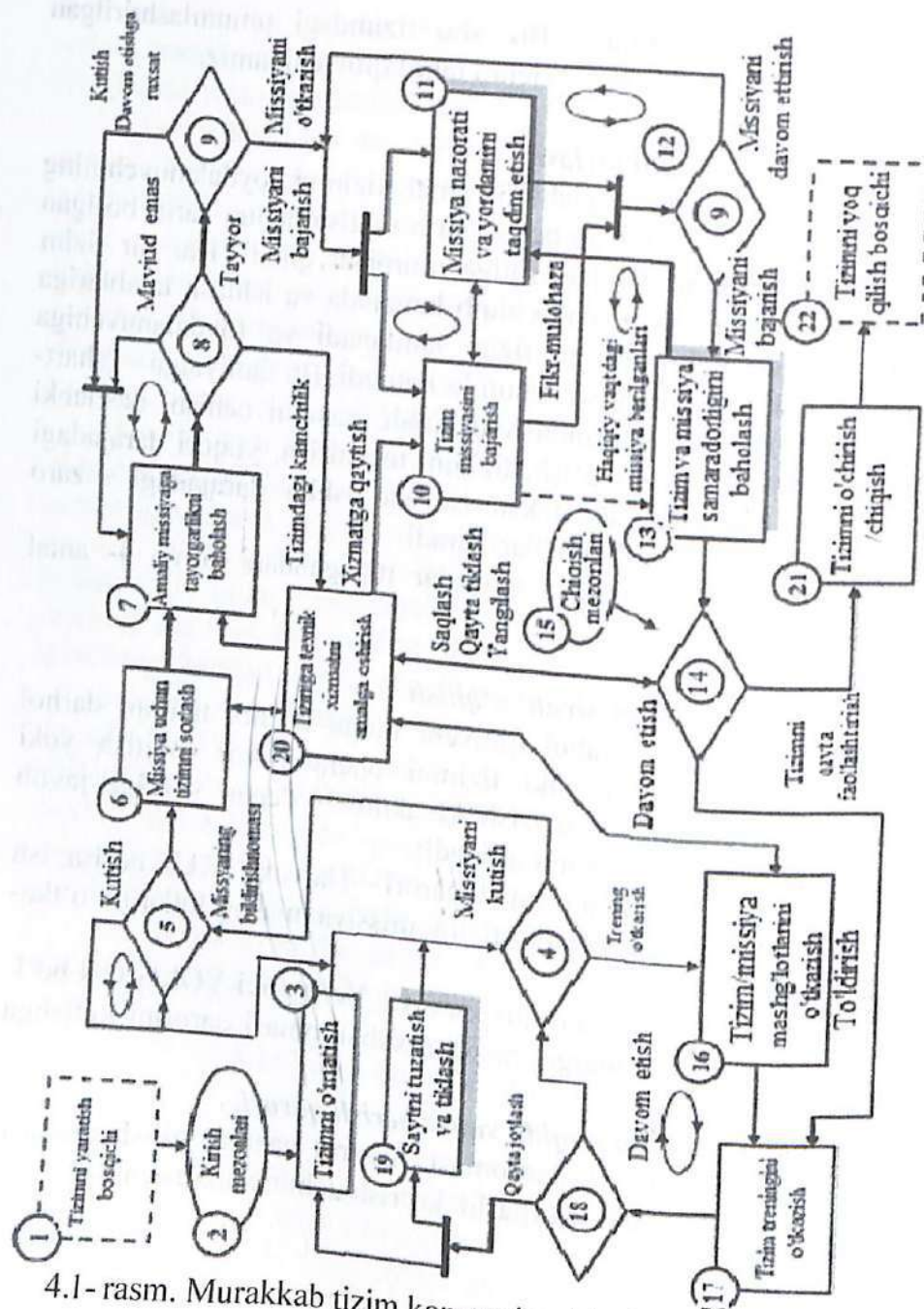
Bu amal, qaror qabul qiluvchi nuqta bo'lib, tizimni darhol faollashtirish kerakmi yoki tizimni operatorlarni o'qitish yoki namoyishlar holatiga o'tkazish kerakmi? - degan savolga javob tekshiriladi va qaror nazorat qilinadi:

- agar tizim/missiya o'qitish qarori – Ha yoki TRUE bo'lsa, ish jarayoni 16- amalga o'tadi (tizim missiya mashg'lotlarini o'tkazish);

- agar tizim/missiya o'quv qarori – Yo'q yoki YOLG'ON bo'lsa, ish jarayoni 5- amalga, missiya xabarnomasi qarorini kutishga o'tadi.

#### 5- amal. Missiya haqida xabar berish qarori.

5- amal missiya xabarnomasi, qarorni nazorat qilish nuqtasi, missiyani bajarishga tayyorgarlik ko'rish uchun xabarnomani



4.1- rasm. Murakkab tizim konsepsiyasidagi amallar modeli

kutishi kerak. Tizim va qo'llanilishiga qarab, 4- va 5- amallarning har biri samarali ravishda tizim uchun siklik kutish nuqtasi bo'lib, yuqori darajadagi bolim, missiyani bajarish to'g'risida buyruq bermaguncha ishlaydi:

- agar missiya xabarnomasi qarori – Ha yoki TRUE bo'lsa, ish jarayoni missiya uchun tizimni sozlash 6- amalga o'tadi;
- agar missiya xabarnomasi qarori – "Yo'q" yoki "Yolg'on" bo'lsa, ish jarayoni 4- amalni o'tkazish tizimi/ missiyani o'qitish qaroriga qaytadi.

**6- amal. Missiya uchun tizimni sozlash.**

6- amal, missiya uchun tizimni sozlash tizimi, kerakli topshiriq uchun tizimni tayyorlash va sozlash uchun zarur bo'lgan tizim imkoniyatlari va tadbirlarini o'z ichiga oladi. Missiya buyurtmalarni olgandan so'ng, tizim missiya uchun sozlanadi. Missiyaning amaliy faoliyatiga, oldindan rejalashtirish, jismoniy asboblardan dasturiy ta'minotni o'zgartirish, xodimlarni o'qitish kiradi. Tizim konfiguratsiyasi - qayta konfiguratsiyasi faoliyati tizim elementlarining moslashtirilgan majmuasini o'z ichiga oladi, masalan:

1. xodimlar – operatorlar, administratorlar va boshqalar;
2. prosedurali berilganlar – ishchi protseduralar, ommaviy axborot vositalari va boshqalar;
3. mutaxassislar tomonidan yaratilgan tizimlar bilan do'stona interfeyslar, kooperativ tizimlar;
4. missiya vazifalari va maqsadlariga ketma-ket yerishishga qaratilgan masala va maqsadlarni izchil hal qilish uchun zarur bo'gan qo'llab-quvvatlash tizimlari – ommaviy axborot vositalari, instruktorlar, ta'minotchilar, xizmat ko'rsatuvchilar va boshqalar.

Faoliyatni tugatgandan so'ng, tizim missiya uchun to'g'ri sozlanganligiga ishonch hosil qilishi uchun tizim tekshiruvi o'tkaziladi:

- agar tizimni tekshirish muvaffaqiyatli bo'lsa, ish jarayoni 7- amal, amaliy missiya tayyorligini baholashga o'tadi;

• agar tizim nuqsonlari yoki kamchiliklari aniqlansa, ish jarayoni 20- amalga, tizimga texnik xizmat ko'rsatishni amalga oshirishga o'tadi.

#### **7- amal. Amaliy missiyaga tayyorgarlikni baholash.**

Amaliy missiyaga tayyorgarlikni baholash amali, tizim imkoniyatlari va belgilangan missiyani bajarishga umumiy tayyorlikni tekshirish uchun zarur bo'lgan tadbirlarni o'z ichiga oladi. Tizim missiya uchun sozlanganidan va barcha tizim elementi zaxiralari to'liq integratsiyalashganidan va ishga tushirilgandan so'ng, missiyaning ishlashga tayyorligi baholanadi. Baholash tizim elementlarining yaxlit to'plamining (masalan, uskunalar, xodimlar va ob'ektlar) o'zlariga yuklangan vazifani bajarishga tayyorligini tahlil qiladi va baholash uchun 8- amal, missiyaga qaror tayyorlashga o'tadi.

#### **8- amal. Missiyaga qaror tayyorlash.**

Agar tayyorlikni baholash natijasi, "Yo'q" bo'lsa, tizim qizil yoki sariq yorliq bilan ishda nuqsoni mavjud, deb belgilanadi. Baholash tizim elementlarining yaxlit to'plamida etishmovchilik bo'lsa, "Mavjud emas" xabari chiqadi va 7- amal "Amaliy topshiriq tayyorligini baholash"ga o'tadi.

Kamchilik mavjudligini aniqlash uchun missiyaga ta'sir qilish xavfini baholash qarori qabul qilinadi, missiyani bekor qilishni yoki tizim/elementni zaxira tizimi bilan almashtirishni kafolatlaydi:

• agar tizimda texnik kamchilik mavjud bo'lsa, ish jarayoni 20- amalga, tizimga texnik xizmat ko'rsatishni amalga oshirishga o'tadi.

• agar tizim missiyani qo'llab-quvvatlash uchun zarur bo'lgan imkoniyatlarni taqdim etishga qaror qilgan bo'lsa, ish jarayoni 9- amalga, missiyani davom ettirish qaroriga o'tadi.

#### **9- amal. Missiyani boshlash qarori.**

Missiyaga ruxsat, qarorni nazorat qilish punkti, missiyani bajarish bo'yicha topshiriqlar berilgan yoki yo'qligini aniqlaydi:

• agar 9- amal, "missiyani kutish" qarori – "Ha" yoki "TO'G'RI" bo'lsa, ish jarayoni 10- amalga, missiyani o'tkazishga o'tadi;

• agarda 9- amalda, missiyani kutish bo'yicha qaror – "Yo'q" yoki "YOLG'ON" bo'lsa, tizimning tayyorligi vaqti-vaqti bilan 7- amal "Amaliy topshiriq tayyorligini baholash"ga qaytish orqali tekshiriladi.

#### **10- amal. Tizim missiyasini bajarish.**

10- amalda, tizim vazifasini bajarishda, tizimning asosiy va ikkinchi darajali vazifalarini bajarish uchun zarur bo'lgan tizim imkoniyatlari va faoliyatini o'z ichiga oladi. Ushbu amal davomida tizim asosiy va ikkinchi darajali missiya maqsadlarini amalga oshirishda tahdid va imkoniyatlarga duch kelishi va ularni jalb qilishi mumkin.

Agar missiyani amalga oshirish davomida tizim texnik xizmat ko'rsatishni talab qilsa, 20- amalga, tizimga texnik xizmat ko'rsatish amaliga o'tishi mumkin.

Fikr-mulohaza asosida missiyani davom ettirish uchun 9- amalga qaytishi yoki "haqiqiy vaqt oralig'idagi missiya berilganlari bilan", 13- amal – "tizim va missiya zaxiralarini baholash"ga o'tishi mumkin.

**2- misol.** Amaliy missiyani bajarish vaqtida samolyotga yonilg'i quyish kerak bo'lishi mumkin.

#### **11- amal. Missiya nazorati va yordamini taqdim etish.**

Ushbu amalda, missiya nazorat funksiyalari orqali, missiyaning bajarilishini tekshiradi va zarur bo'lsa, yordami taqdim etiladi. Missiya bajarilishi bo'yich yakuniy qaror qabul qilish uchun

12- amalga o'tiladi, missiyani davom ettirish uchun 9- amalga, missiyaning yangi funksiyalarini olish uchun 10- amalga o'tiladi.

**12- amal. Missiyaning yakuniy qarori.**

12- amalda missiya bajarilishi bo'yicha yakuniy qaror qabul qilinadi. Missiya davom ettirilsa, 9- amalga, agarda missiyaning nazorati va yordami talab qilinsa 11- amalga qaytiladi.

**13- amaliy. Missiya va tizim ish faoliyatini baholash.**

13- amaliy - missiya va tizim samaradorligini baholash tizimi missiyaning asosiy va ikkinchi darajali maqsadlari va tizimning ushbu muvaffaqiyatga qo'shgan hissasi asosida missiya muvaffaqiyati darajasini ko'rib chiqish uchun zarur bo'lgan tizim imkoniyatlari va tadbirlarini o'z ichiga oladi. Faoliyatlarga missiya keyingi berilganlarini qisqartirish, maqsadli ta'sirni baholash, kuchli va zaif tomonlari, tahdidlar, missiya haqida qisqacha kuza-tishlar, olingan saboqlar va missiya muvaffaqiyati kiradi. Ushbu amallar, shuningdek, missiyani bajarishda insonning ish faoliyatini, kuchli va zaif tomonlarini ko'rib chiqish imkoniyatini beradi. Boshqaruv 14- amalga o'tadi.

**14- amal. Tizimni o'chirish/bosqichma-bosqich o'chirish qarori.**

14- amal – o'chirish/bosqichma-bosqich chiqarish tizimi, qarorni nazorat qilish nuqtasi, tizim joriy amallarni davom ettirishi, yangilanishi yoki ishdan chiqarilishi yoki faol xizmatdan bosqichma-bosqich olib tashlanishi kerakligini aniqlaydi. Qaror tizim uchun belgilangan "chiqish mezonlari" (15)ga asoslanadi:

- tizimni o'chirish/bosqichma-bosqich o'chirish qarori – Ha yoki TRUE bo'lsa, ish jarayoni 21- amalga, "tizimni o'chirish/ bosqichma-bosqich o'chirish"ga o'tadi;

- agar qaror – "Yo'q" yoki "YOLG'ON" bo'lsa, ish jarayoni 17- amalga, "tizim zaxiralarini to'ldirish" amaliga o'tadi.

**16- amal. Tizim/missiya mashg'ulotlarini o'tkazish.**

16- amal – "tizim o'rgatishini o'tkazish" tizim foydalanuvchilarini yoki tizim operatorlarini tizimni qanday to'g'ri ishlatishni o'rgatish uchun zarur bo'lgan imkoniyatlar va tadbirlarni o'z ichiga oladi. Kattaroq, murakkabroq tizimlar uchun boshlang'ich operatorni o'qitish ba'zan tizimni xotira maydonga joylashtirishdan oldin tizim ishlab chiqaruvchisining komputerida amalga oshiriladi. Tuzatish va malaka oshirish treningi tizim xizmatda bo'lganidan keyin amalga oshiriladi. 16- amal davomida, yangi tizim operatorlariga asosiy ko'nikmalarni rivojlantirish uchun tizimdan xavfsiz va to'g'ri foydalanish bo'yicha ko'rsatmalar beriladi.

Tajribali operatorlar, shuningdek, oldingi missiyalardan olingan saboqlar yoki raqib yoki raqobatchi tahdidlar tomonidan qo'llaniladigan yangi taktikalar asosida tuzatuvchi, malaka yoki malaka oshirish treninglarini olishlari mumkin.

O'quv mashg'uloti tugagach, ish jarayoni 17- amalga, ya'ni "tizim zaxiralarini to'ldirish"ga o'tadi. Agar o'qitish davomida tizim texnik xizmat ko'rsatishni talab qilsa, 20- amal, tizimga texnik xizmat ko'rsatish amali faollashtiriladi.

**17- amal. Tizim zaxiralarini to'ldirish.**

17- amalda, tizim zaxiralarini to'ldirish tizim imkoniyatlari va xodimlar, yoqilg'i va materiallar kabi tizim zaxiralarini zaxiralash yoki to'ldirish uchun zarur bo'lgan tadbirlarni o'z ichiga oladi. 17- amal tugagach, ish jarayoni 18 – "tizimni qayta yuklash qarori" amaliga o'tadi.

**18- amal. Tizimni qayta yuklash qarori.**

Tizimni qayta yuklash qarori bo'yicha 18- amal, qarorni nazorat qilish nuqtasi, jismoniy tizimni tashkiliy missiya maqsadlarini qo'llab-quvvatlash uchun yangi foydalanuvchi joylashgan joyga qayta joylashtirish kerakligini aniqlaydi:

- 18- amal – "tizimni qayta yuklash qarori" - Ha yoki TRUE

bo'lsa, ish jarayoni tizimni joylashtirish uchun bo'lgan 3- amalga yoki 19- amal, "saytni tuzatish va tiklash"ga o'tadi;

• agarda "tizimni qayta yuklash qarori" - Yo'q yoki YOLG'ON bo'lsa, ish jarayoni 4- amalga "tizim missiyani o'qitish"ga o'tadi.

#### **19- amal. Saytni tuzatish va tiklash.**

19- amalda tizim faoliyt yuritayotgan saytda xatoliklarni tuzatish va tizimni qayta tiklash jarayonlari amalga oshiriladi. Bu jarayondan keyin ish jarayoni 4- amalga "tizim missiyani o'qitish"ga o'tadi.

#### **20- amal. Tizimga texnik xizmat ko'rsatishni amalga oshirish.**

Ushbu amalni bajarish, tizim imkoniyatlarini va tizim imkoniyatlarini yangilash yoki tizim kamchiliklarini oldini olish yoki tuzatuvchi texnik xizmat ko'rsatish orqali tuzatish uchun zarur bo'lgan tadbirlarni o'z ichiga oladi. Tizimlar missiya muvaffaqiyatiga ta'sir qilishi mumkin bo'lgan har qanday nuqson yoki kamchiliklarni tuzatish uchun zarur bo'lgan tuzatuvchi yoki profilaktika tadbirlarini ifodalash uchun, odatda "qizil" yoki "sariq" kabi osongina tanib olinadigan rang aniqlovchilari bilan belgilanadi.

Tizimga texnik xizmat ko'rsatish muvaffaqiyatli yakunlangandan so'ng, tizim quyidagi amallarning biri orqali faol ishga qaytariladi:

- 6- amal. Missiya uchun tizimni sozlash;
- 7- amal. Amaliy missiyaga tayyorgarlikni baholash;
- 10- amal. Tizim vazifasini bajarish;
- 16- amal. Tizim zaxiralarini to'ldirish;
- 17- amal. Tizim/missiya mashg'ulotlarini o'tkazish.

#### **21- amal. Tizimni o'chirish/bosqichma-bosqich o'chirish.**

21- amal – "tizimni o'chirish/bosqichma-bosqich o'chirish"

tizim imkoniyatlarini va tizimni faol xizmatdan, saqlashdan, ombordan, "konservatsiya"dan o'chirish va olib tashlash uchun zarur bo'lgan tadbirlarni o'z ichiga oladi.

Tizimni o'rnatish va uning barcha tarkiblari va elementlarini to'g'ri yo'q qiling. Ba'zi tizimlar mavjud tizimlar tomonidan qo'llab-quvvatlanmaydigan missiya amallaridagi keskinlikni qo'llab-quvvatlash uchun kelajakda kerak bo'lgunga qadar saqlanishi yoki "konservatsiya" qilinishi mumkin. O'chirishni tugatgandan so'ng, tizim o'z tizimi/mahsulot hayotiy siklining "tizimni yo'q qilish" bosqichiga (22) o'tadi.

#### **4.5. Muhimligi umumiylangan tizim amaliyotlari modeli**

Tizim amallari modeli vaqtga asoslangan jadvalga sinxronlashtirilgan tizimning umumiy yig'indisini tartibga soluvchi yuqori darajadagi qolip bo'lib xizmat qiladi. Modeldagi amallar tizim elementlarining bir nechtasi tomonidan amalga oshiriladi (uskunalar, xodimlar, ob'yektlar va boshqalar). Ushbu amallarni tizim elementlariga taqsimlash bir necha nuqtayi nazardan muhimdir.

#### **4.6. Axborot tizimlari va sinflari**

**Axborot tizimi (AT)** – qo'yilgan maqsadga nisbatan berilganlarni to'plash, saqlash, ishlov va chiqarish uchun vosita, usul va xodimlarning o'zaro bog'langan majmuasidan iborat. AT qo'yilgan maqsadga erishish uchun inson bilan hamkorlikda yuqorida keltirilgan vazifalarni bajarishni ta'minlaydi.

AT turli yondashuvlar asosida har xil sinflarga ajratiladi.

A) An'anaviy AT turlari:

- korporativ;
- guruhli;
- operatsion boshqaruv va buxgalteriya tizimlari;
- tahlilchi;
- ma'lumot va huquqiy;

➤ tranzaksiyani qayta ishlovchi;

➤ qaror qabul qiluvchi.

B) O'zaro harakatlarning yangi standartlari bo'yicha sinflar:

• **G2C** – *government-to-citizen* – davlat xizmatlari va fuqarolar orasidagi aloqalar uchun;

• **G2B** – *government-to-business* - davlat va xususiy kompaniyalar orasidagi aloqalar uchun;

• **G2E** – *government-to-employee* - davlat tashkilotlari va ularning xodimlari orasidagi aloqalar uchun;

• **G2G** – *government-to-government* - turli davlat tashkilotlari va davlat boshqaruv darajalari orasidagi aloqalar uchun.

*Korporativ AT (KAT)* korxonalarini boshqarish muammolarini vertikal (birlamchi berilganlardan tortib yuqori boshqaruv qarorlarini qo'llab-quvvatlashgacha) va gorizontal (faoliyatning barcha sohalari va texnologik amallarni) majmuaviy hal qilishni ta'minlaydi. KATning butun doirasini integratsiya darajasiga ko'ra uch guruhga bo'lish mumkin: yirik, o'rta va kichik. Ular funksiyalar to'plami, narxi va amalga oshirishning murakkabligi bilan farqlanadi.

Yirik korporativ axborot tizimlari ko'pincha tayyor axborot tizimlari emas. Ular dasturiy modullar va berilganlar bazalari to'plamini, shuningdek ularni sozlash va qo'llash texnologiyalarini ifodalaydi. Ushbu tizimlar universal va ko'p tarmoqli tarzda yaratilgan bo'lib, ularning narxi ancha qimmat, o'rnatish va boshqarish esa qiyin.

*Guruhli AT* - mahalliy tarmoqda guruhli ishlash uchun mo'ljallangan. Bu erda quyidagi relyatsion berilganlar bazasini boshqarish tizimlari (BBBT) qo'llaniladi:

Oracle

MS SQL Server

My SQL

PostGreSQL

Sybase

MS Access

*Operatsion AT* tashkilotning operatsion faoliyatini avtomatlashtirish uchun mo'ljallangan.

Operatsion ATdagi asosiy modullar quyidagilardan iborat:

- hujjat aylanishi;
- xaxiralarni hisobga olish;
- hisob-kitob;
- xodimlar hisobi.

*Tahlilchi AT (TAT)*. Asosiy maqsad - tashkilot faoliyati va uning ish muhitini to'liq va har tomonlama tahlil qilish uchun zarur bo'lgan axborotlarni to'plash.

TAT tuzilishi quyidagicha:

- axborotni saqlash moduli;
- axborot qidirish moduli;
- axborotni tahlil qilish moduli;
- tizim konfiguratsiyasi va sozlash moduli.

TAT turlari:

❖ Ko'p o'lchovli berilganlar bazalari (BB) (MDDDB – MultiDimensional Data-base). Ko'p o'lchovli BB dagi berilganlar jadvallarda indekslangan yozuvlar ko'rinishida emas, balki ko'p o'lchovli tartiblangan massivlar yoki kublar shaklida saqlanadi.

❖ Axborot omborlari (AO) (DW - Data Ware-house). Ularda axborotni saqlash ko'p o'lchovli BB dagi berilganlarni saqlashga o'xshaydi. AO dagi qurilish chizmasi "*qor parchasi*", deb ataladi.

❖ Ixtisoslashgan relyatsion BB. Relyatsion BB modeli ob'yektning murakkab berilganlar modeli bilan qidirish va hisoblash tezligiga cheklovlar qo'yadi.

Tahliliy berilganlar uchun ishlatiladigan BBBT bu - ROLAP (Relation OnLine Analytical Processing) tizimidan iborat.

*Huquqiy axborotlarni izlash tizimlari (HAIT)*. HAITdagi asosiy funksiyalar:

- amaldagi qonunchilikning amaldagi holatini aks ettiruvchi huquqiy axborot bazasini saqlash va doimiy yangilab borish;
- norasmiy belgilangan qidiruv mezonlari yordamida berilganlar bazasida hujjatni qidirish.

*Tranzaksiyalarni qayta ishlash tizimlari*, o'z navbatida, berilganlarni qayta ishlash tezligiga qarab ikki turga bo'linadi:

- ✓ paketli;
- ✓ tezkor.

*AT tuzilishi bo'yicha sinflari*. AT tuzilishi bo'yicha, guruh va korporativ AT quyidagi sinflarga bo'linadi:

- "fayl-server" arxitekturasi asoslangan tizimlar;
- mijoz-server arxitekturasi asoslangan tizimlar;
- ko'p darajali arxitekturaga asoslangan tizimlar;
- Internet/Intranet texnologiyasiga asoslangan tizimlar.

*Qarorlarni qabul qilishni qo'llab-quvvatlash tizimlari DSS* (Decision Support System) - murakkab so'rovlar yordamida vaqt, geografik va boshqa parametrlar bo'yicha berilganlarni tanlash, qayta ishlash va tahlil qilishga asoslangan axborot tizimining yana bir turi.

*Texnik tuzilish bo'yicha AT sinflari:*

- avtomatizatsiyalashtirilgan
- avtomatlashtirilgan.

#### 4.7. Tizim amallari lug'ati

Amallar konsepsiyasini grafik tasvirlash bo'yicha jamoaviy kelishuvni olish faqat birinchi qadamdir. Kattaroq, murakkab tizimlar va ishlab chiqish guruhlar bilan ishlashda ushbu darajadagi modellar jamoa a'zolari o'rtasida to'g'ri tushunishni ta'minlash uchun har bir qobiliyat uchun qamrovli ta'riflarni talab qiladi. Misol uchun, siz va sizning jamoangiz tizimning qo'llanilishiga qarab, boshqa tijorat sohasida ishlaydigan jamoadan farqli ravishda muayyan amaliy qobiliyatni belgilashingiz mumkin.

Yechimlardan biri tizim amallari lug'atini yaratishdir. Oldingi Tizim Amallari Modeli tavsiflariga o'xshash har bir qobiliyatni belgilaydigan va qamrab oluvchi lug'at tizimning butun hayoti davomida saqlanishi kerak.

#### 4.8. Xulosa

Tizim amallari modeli tizimlarni, mahsulotlarni, tashkilotlarni, xizmatlarni va hokazolarni aniqlash uchun ishlatiladi. Biz ushbu modelni avtomobillar, kasalxonalar, korxonalar, yong'in va tez yordam xizmatlari, kosmik kemalar, aviakompaniyalar kabi mutaxassislar yaratgan tizimlar uchun, hatto hammasi bo'lmasa ham, boshlang'ich nuqta sifatida qo'llashimiz mumkin. Birgalikda va shaxsiy ravishda modelning har bir amali ko'pgina tizimlar uchun qo'llaniladigan umumlashtirilgan konstruksiyani ifodalaydi. Muhandislik tizimlari sifatida foydalanuvchilar bilan shartnomaviy, qonuniy va tartibga soluvchi talablar doirasida ularning ehtiyojlarini aks ettirish uchun tizim amallari modelini moslashtirish uchun hamkorlik qiling. Har bir amalni ta'minlash uchun tizim amallari lug'ati orqali qamrab olinishi va chegaralanishi kerak.

Ekvayer, foydalanuvchi va tizimni ishlab chiqish guruhlarining barcha a'zolari hech qanday kutilmagan hodisalarsiz, ma'lum amallarga nimalar kiritilgan/kiritilmaganligin aniq tushunishlarini ta'minlash uchun har bir amalni tizim amallari lug'ati orqali qamrab olish va chegaralash kerak.

Tizim amallari modeli juda oddiy ko'rinishi mumkin. Biroq, chuqurroq o'rganib chiqsak, hatto oddiy tizimlar ham amaliy ketma-ketlikni etarli darajada aniqlash uchun oldindan o'ylashni talab qiladi.

#### 4.9. Nazorat savollari

1. Tizim amallari modeli nima?
2. Amallar konsepsiyasi nima?

3. Tizim amallari modelining maqsadi nima?
4. Tizim amallari modelini grafik tasvirlab bering.
5. Modelning har bir amalini yoki vazifalarini tavsiflang.
6. Modellardagi farqlarni uning mustahkam naqlidan ajratib oling.
7. Tizim amallari lug'ati nima?
8. Tizimlar qanday turdagi sinflardan iborat?
9. Yangi standartlar bo'yicha sinflarni ta'riflab bering.
10. Korporativ AT kamchiligi nimadan iborat?

## V- BOB. YUQORI DARAJADAGI TALABLAR

Tizimdan foydalanish holatlarini aniqlash – bu birlashtirilgan talablar intizomining bir qismi bo'lgan faoliyat, ya'ni jarayondir. Talablarni aniqlash va tahlil qilish boshlang'ich bosqichning muhim qismidir. Tizimni loyihalashtiruvchilar jamoasi talablarni aniqlash uchun har bir mavjud manbadan (mutaxassislar, foydalanuvchilar, hujjatlar, interfeyslar, adabiyotlar va h.k.) foydalanishlari mumkin va har bir manba uchun tizim bajarishi kerak bo'lgan funksiyalar to'plami aniqlanishi kerak.

### 5.1. Talablarni aniqlash

Talablarni aniqlash tizim bajarishi kerak bo'lgan funksiyalar va tizim faoliyat yuritishi kerak bo'lgan cheklovlar to'g'risida ma'lumot izlashga mos keladi. Ushbu darslikda keltirilgan talablar, ulardan tizimda foydalanish holatlarida qayd etiladi. Foydalanish holatlarisiz talablarni ro'yxatdan o'tkazishning yana bir usuli bu funksional talablar ro'yxatidan iborat hujjatlardan foydalanishdir, ular cheklovlar ro'yxati bilan birga kelishi ham mumkin. Funksiyalar ro'yxati o'rniga, foydalanish holatlaridan foydalanishning afzalligi shundaki, yaxshi foydalanish holati aniq belgilangan qismlarga ega, bu alohida funksiyalar ro'yxatiga qaraganda ancha tushunarli bo'lgan yuqori darajadagi talablar ro'yxatini yaratishga imkon beradi.

Odatda, yuqori darajadagi foydalanish holatlari varianlarining soni, shaxsiy funksiyalar sonidan ancha past bo'ladi. Shaxsiy funksiyalar yuqori darajadagi foydalanish holatlari o'z tuzilishiga ega bo'lganda, foydalanish holatini kengaytirishda batafsil paydo

bo'ladi. Masalan, Livir tizimi [1] misolida biz, tizim kitoblarni sotib olishi va sotishi, to'lovlarni qabul qilishni nazorat qilishi, lat yegan kitoblardan foydalanishga ruxsat berishi, mijozlarni ro'yxatdan o'tkazishi, savdo hisobotlarini tuzishi, kitoblar mavjudligini tekshirishi va h.k. bo'yicha turli talablarni aniqlashda imkon beradi. Ushbu amallar va boshqalar tizimning funkSIONalligini tashkil qiladi va shuning uchun bu talablar *funksional talablar* deb ataladi. Ushbu funksiyalarni bir yoki bir nechta holatlardan foydalanishga kiritish mumkin.

Boshqa tomondan, talablarni aniqlash paytida tahlilchi tijorat qoidalari yoki tizim tomonidan funksiyalarni qanday bajarish kerakligi bo'yicha cheklovlarga duch kelishi mumkin. Masalan, tijorat qoidasida, kitob do'koni kitoblarni faqat to'lov tasdiqlangandan keyin yuborgan bo'lishi mumkin. Bunday qoida funkSIONal bo'lmagan talab bo'lib, uni foydalanish holatining o'zida izoh yoki sharh sifatida qayd etish mumkin. Foydalanish holati batafsil bo'lishi uchun uni chaqirib olishi va tekshirishi mumkin. Shu bilan bir qatorda, uni raqamlangan ro'yxat yoki foydalanish holatlari noyob raqami orqali elektron jadval sifatida alohida yozib olish mumkin.

## 5.2. Talablarni aniqlash loyihalash emas

Tahlil qilinadigan tizim o'rmonga o'xshaydi. Noma'lum o'rmonni tahlil qilish uchun, har bir o'simlik va hayvonni tekshirishdan boshlash noto'g'ri. Ba'zi odamlar, o'rmonni daraxtlar bo'lgani uchun ko'ra olmaydi, degan gap bor. O'rmon – bu tizim, daraxtlar esa – talablar, deb shartli ravishda qarashimiz mumkin. Faqat jaryonning oxirida tahlil qilayotgan jamoa eng kichik qismlar haqida ham bilimga ega bo'ldik, deb aytishi mumkin. Birinchidan, hammasi xususida tasavvurga ega bo'lish kerak va shundan keyingina tafsilotlarni o'rganish mumkin.

Shunday qilib, boshlang'ich bosqich tasavvurni to'liq ta'minlashi kerak, chunki avval ko'rish va aniqlash uchun nima muhimligini bilish, keyin esa ana shu yaxlitlikni qismlarga ajratish va uni o'rganish va oxirida yechimni ishlab chiqish kerak bo'ladi. Ishlab chiqish va qurilish bosqichlarida yaqinlashuvlarni tashkil etish, har bir sektorni bir vaqtning o'zida ko'rish uchun o'rmonni bo'laklarga bo'lishga to'g'ri keladi va shunday qilib, o'ziga xos murakkablik bilan kurashish imkoniyati paydo bo'ladi. Shunday qilib, boshlang'ich bosqichning yakuniy maqsadlaridan biri foydalanish holatlarida ishni taqsimlashni tashkil qilish davomida bo'lajak yaqinlashishlar orqali ularni o'rganib olishdir.

Boshlang'ich bosqichda talablarni aniqlash tez va umumiy bo'lishi kerak. *Buning to'g'ri usuli – bu talablarning tafsilotlariga emas, balki ularni kengaytirishga e'tibor qaratish kerak.* Tahlilchi tizimni kengaytirishda nima qilish kerakligini tushunishi kerak, buni qanday amalga oshirish haqida batafsil ma'lumotga ega bo'lmagan holda. *Faqat ishlab chiqishdagi yaqinlashuvlar davomida talablarni tahlil qilish chuqurlashtiriladi.*

Talablarni aniqlash uchun ketgan vaqt, ixtiro bilan emas, balki kashfiyot bilan bog'liq bo'lishi kerak. Bu davr davomida tahlilchilar jamoasi mijozlar, foydalanuvchilar va boshqa manfaatdor tomonlar bilan birgalikda, tafsilotlarsiz eng zarur imkoniyatlar va cheklovlar haqida ro'yxat tuzadi. Talablar bo'yicha tafsilotlar keyingi yaqinlashuvlar paytida qulay tarzda joylashtiriladi.

*Shuni ta'kidlash kerakki, talablar – bu mijoz so'ragan narsa ekanligi aniq bo'lishi kerak, aslo jamoa yaratayotgan loyiha emas.* Ba'zi tahlilchilar talablarni to'plashni, ya'ni mijozlarning talablarini saqlashni, tizim loyihasining boshlanishi bilan aralashtiradilar. Bunday chalkashliklarga bitta misol berilganlar bazasini loyihalash bilan bog'liq talablar termasi. Agarda yangi tizim bilan moslashuvchan eski tizim bo'lmasa, relatsion jadvallar termasi mijoz talabi ekanligini qanday asoslash mumkin? Bu holat, oxir-oqibat ba'zi bir tajribali kompyuter ilmlari bo'yicha ilmiy darajaga ega

bo'lgan mijozlar bilan bo'lishi mumkin, lekin bu umumiy qoida emas. Berilganlar bazasidagi jadvallarini, muammoli sohaning emas, balki ma'lum bir sohadagi yechimning bir qismi sifatida qarash mumkin. Tahlilchi talablarni, axborot nuqtayi nazaridan mijozning ehtiyojlari va maqsadlariga mos keladiganlardan izlashi kerak. Keyinchalik, u ushbu ma'lumot, relyatsion berilganlar bazasidami yoki boshqa tuzilmada saqlanishini hal qilishi mumkin.

### 5.3. Talablardagi muammolar

Annotatsiyali va qo'shimcha tasnifli foydalanish variantlari diagrammalari yordamida shakllantirilishi mumkin bo'lgan talablar hujjati, tizimning faoliyati uchun ununhg imkoniyatlari va shartlari ro'yxatga olinishi kerak. Talablar bilan bog'liq muammolar kamida quyidagilardan iborat [1]:

- Talablarni qanday aniqlash mumkin?
- Loyihaning boshqa bosqichlari va jamoalariga talablarni qanday yetkazish kerak?
- Ishlab chiqish davrida talablarning barchasi amalga oshirilganligini tekshirish uchun talablarni qanday eslash kerak?
- Talablar o'zgarishini qanday boshqarish kerak?

Agarda keyinchalik, foydalanish variantlari diagrammasida yaxshi ishlab chiqilgan talablar loyihaga kiritilmagan va keyinchalik buni bilolmasak, diagramma befoyda bo'lar edi. Ushbu tekshiruvni o'tkazish uchun avtomatik mexanizmlarning mavjudligi juda muhimdir. Shuning uchun foydalanish variantlari va loyiha-ning boshqa qismlari o'rasida kuzatuv munosabatlarini qo'llab-quvvatlash muhimdir.

Shuni nazarqa tutish kerakki, loyihani yaratish jarayonida talablar albatta o'zgaradi. Shunday qilib, o'zgarishlarni boshqarish zarur, ular boshlangandan keyin oldini olish kerak emas. Ba'zan, tizim o'rnatilgandan so'ng talablar o'zgaradi. Kontekst shartlari,

qoidalar, kompaniya siyosati yoki ish usullari istalgan vaqtda o'zgarishi mumkin. Garchi tahlilchi bularni oldindan ko'ra olmasa ham, zarur bo'lganda o'zgarishlar jarayonini yengillashtirish uchun o'zgarishlar, ularni joylashtirish mexanizmlari yaratilishi mumkin. Ushbu talab beqarorligini engish uchun maxsus loyihalash andozalari mavjud. O'zgarishlarni hech qachon oldindan aytib bo'lmaydi. Agar tizimda talablardagi o'zgarishlar nazarqa tuzilmagan bo'lsa, ehtimol ularni amalga oshirish qiyin bo'ladi. Bunday vaziyat, talablarga asoslangan tahlil va loyihalash jarayonlarini amalga oshiradigan ko'pgina tizimlar uchun aynan bir xilda bo'lmaydi. Tizim arxitekturasini qo'llab-quvvatlash uchun asos sifatida talablardan foydalanish, harakatlanayotgan qum ustida uy qurishga o'xshaydi; talablar o'zgarganda, tuzilish zarar ko'radi. Biroq Birlashtirilgan Jarayon (Unified Process - UP) tizimi arxitekturasini ancha barqaror bo'lgan ma'lumot va xatti-harakatlarni qamrab oladigan sinflar va tarkiblarga asoslanadi. Bu sinflar, talablarni birgalikda amalga oshirish imkoniyatini beruvchi funksional imkoniyatlarni amalga oshiradi. Agar talablar o'zgarsa, kombinatsiyalar o'zgaradi, lekin asosiy tuzilma emas. Bunday arxitektura Meyer-ning "ochiq-yopiq" tamoyili asosida ishlaydi, ya'ni har doim o'zgartirish uchun yopiq (u ishlaydi), lekin kengaytirish uchun ochiq (u yangi funksional imkoniyatlarni qo'shish mumkin).

Har bir talabning kelib chiqishini kuzatish muhimdir (masalan, ishbilarmon aktyor, ishbilarmon ishchi, mijozning o'zi yoki hatto sohosi mutaxassisi), chunki talablarni shu manbalar bilan taqqoslash kerak, ularning yaxshi yozilgani, to'liq va asosli ekanligini tekshirish zarur.

Ba'zan turli xil odamlar yoki bo'limlar, bir xil talab uchun turli xil tasniflarni taqdim etishlari ham mumkin. Bunday holda, ular orasida shartnoma tuzish kerak, yoki talab uchun maqbul shaklni aniqlash uchun kim eng yuqori vakolatga ega ekanligini aniqlash kerak.

#### 5.4. Ayon va yashirin funksional talablar

Funksional talablarni ayon yoki yashirin, deb aniqlash shart emas. *Ayon funksional talablar* – bu foydalanuvchi tomonidan amalga oshiriladigan funksiyalardan iborat. Tizimning maqsadlari va funksiyalarini va ularni kompyuterda bajarish tamoyillarini belgilash. Bu talablar odatda foydalanuvchi va tizim o'rasidagi ma'lumot almashinuviga mos keladi, masalan tizim interfeysi orqali so'rovlar va berilganlarni kiritish.

*Yashirin funksional talablar* – bu foydalanuvchi bilmagan holda, tizim tomonidan bajariladigan funksiyalar. Odatda bu funksiyalar matematik amallar va berilganlarni yangilashdan iborat bo'lib, foydalanuvchi tomonidan amalga oshirilgan boshqa funksiyalarning oqibatida, foydalanuvchi bilmagan holda, tizim tomonidan bajariladi.

Yashirin talablar tizimning ichida amalga oshiriladi. Shunday qilib, ular foydalanish variantlari diagrammasi kabi oshkor ravishda ko'rinmasa ham, ularni loyihalash va amalga oshirish vaqtida esga tushirish uchun, ular yetarli darajada bir-biriga bog'liq bo'lishi kerak. Shunday qilib, ular foydalanish variantlariga izoh sifatida qo'shilishi mumkin.

Eng ko'p sotilgan kitoblar haqida hisobot ishlab chiqarish ayon talabga misol bo'la oladi va bu narsani ish yurituvchi talab qilishi mumkin.

Yashirin talabga misol, sotish siyosatida chegirma usulini qo'llashdir. Bunda, foydalanuvchi tizimdan amalni bajarishni aniq talab qilmaydi. Chunki, tizim uni avtomatik ravishda amalga oshiradigan faoliyat bo'lgani uchun bu yashirin talab bo'ladi.

#### 5.5. Funksional bo'lmagan talablar

*Funksional bo'lmagan talablar* – bu muayyan tizim funktsiya-

lari bilan bog'liq bo'lishi mumkin bo'lgan cheklovlar yoki sifatlar. Amalga oshirish muhitida dasturiy ta'minot tizimining tashkiliy, ijro etuvchi, operatsion jihatlarini tavsiflash. Masalan, "buyurtmada yuzdan ortiq kitob bo'lishi mumkin emas", "agar aloqa uzilgan bo'lsa, buyurtma tranzaksiyasi saqlanishi kerak" va h.k. Shuning uchun ular tegishli funksiyalarni o'z ichiga olgan foydalanish variantlari diagrammasiga izoh sifatida qaralishi mumkin. Biroq, ba'zida funksional bo'lmagan talablar umumiy bo'lishi mumkin, ya'ni funktsiyaga birlashtirilishi shart emas (masalan, "tizim Java-da amalga oshirilishi kerak") va bu holda ular hujjatning qo'shimcha tasniflarida paydo bo'ladi. Funktsiyaga maxsus birlashtirilgan cheklov va sifatlar *funksional bo'lmagan talablar*, umumiy cheklovlar va sifatlar esa *qo'shimcha talablar* deb ataladi.

Funksional bo'lmagan talablarning ikki turi mavjud:

*Mantiqiy muammolar*: funktsiyaga birlashtirilgan tijorat qoidalari. Masalan, savdoni ro'yxatdan o'tkazish paytida bir qator cheklovlarni ko'rib chiqish mumkin, masalan kredit karta operatori to'lovni tasdiqlamaguncha yoki so'nggi yetkazib berish noto'g'ri manzil tufayli qaytarilgan bo'lsa, sotishni yopmaslik.

*Texnologik muammolar*: funktsiya uchun ishlatiladigan texnologiya bilan bog'liq cheklovlar va sifatlar, masalan, foydalanuvchi interfeysi, aloqa protokolining turi, xavfsizlik cheklovlari, xatolarga bardoshlik va boshqalar.

Masalan, buyurtmani bajarish uchun foydalanuvchi interfeysi ekranlarning ketma-ket oqimiga asoslangan loyiha andozasiga amal qilishi kerakligini aniqlash usuli funktsiyani bajarilishi haqidagi texnologik cheklovdir. Funksional bo'lmagan talabga yana bir misol, "to'lovni tasdiqlash 5 soniyadan oshmasligi kerak". Bu tizimning ishlashi bilan bog'liq texnologik cheklovdir va bu loyiha-zimning kredit karta operatori bilan aloqa mexanizmi haqida o'ylashining kredit karta operatori bilan aloqa mexanizmi haqida o'ylashiga ta'sir qiladi. Bunday holda, tizimni loyihalashda, operatorlar-ga keng yo'lakli ulanishlarni jiddiy ko'rib chiqish kerak bo'ladi.

Talab cheklov qanday amalga oshirilishini belgilamaydi, shunchaki uni talab qiladi.

Tizimni loyihalash va amalga oshirish u yoki bu tarzda talabga javob berishi kerak, aks holda tahlilchi mijoz bilan ba'zi talablarning moslashuvchanligi to'g'risida muzokara olib borishi kerak.

### 5.6. Funktsional bo'lmagan talablarning doimiyligi va vaqtinchaligi

Talabning eng asosiy xususiyatlaridan biri bu funktsional bo'lmagan yoki qo'shimcha talabning doimiy yoki vaqtinchalik bo'lishidir.

Funktsional bo'lmagan va qo'shimcha talablarni, mijoz qaror qabul qilishiga qarab, *doimiy* (ular o'zgarishini kutiladi) yoki *vaqtinchalik* (ular o'zgarishi kutiladi), deb hisoblash mumkin. Doimiylik yoki vaqtinchalik talabning o'ziga xos xususiyati emas, u qulaylik jihatiga qarab aniqlanadi. Dasturiy ta'minotni ishlab chiqish va xizmat ko'rsatishda, bir xildagi talablar vaqt va narxga bog'liq holda, doimiy yoki vaqtinchalik deb hisoblanishi mumkin.

Aksariyat cheklovlar vaqtinchalik bo'lishi uchun kimdir moslashuvchan loyihaga sarmoya kiritsa, xizmat ko'rsatish paytida o'zgarishlarni moslashtirish uchun kamroq kuch sarflanadi. Biroq, bu moslashuvchan loyihaning narxi yaratilish jarayonida ba'zi loyihalar uchun juda katta bo'lishi mumkin. Qaysi cheklovlarni vaqtinchalik talablar sifatida ko'rib chiqish kerakligini, har doim o'ylab ko'rish kerak.

Masalan, qo'shimcha talabni – Livir tizimi yagona valyuta – dollar bilan ishlasin deylik. Agar bu talab *doimiy* deb hisoblansa, tizim yagona valyuta uchun mo'ljallanadi ("dollar" hatto o'zgaruvchilar va atributlarni aniqlash uchun ishlatiladigan berilganlar turi bo'lishi mumkin). Agar talab *vaqtinchalik* deb hisoblansa, boshqa valyuta bo'lmasa ham, bugungi kunda ishlatiladigan butun tizim

kelajakdagi valyutani yoki bir vaqtning o'zida bir nechta valyutalarni ishlatishga tayyor bo'lishi kerak.

Talab doimiy ekanligi haqidagi qabul qilingan qarorning oqibatlarini quyidagicha bo'lishi mumkin:

- Tizimni ishlab chiqish arzon va tez.
- Agarda talablar kelajakda tasodifan o'zgarsa, tizimni o'zgartirish qimmatroq va qiyin bo'ladi.

Boshqa tomondan, talab vaqtinchalik deb qaror qilish quyidagi oqibatlarga olib keladi:

- Tizimni ishlab chiqish qimmatroq va murakkabroq (masalan, valyutani o'zgartirish uchun funktsional imkoniyatlarga ega bo'lishi kerak).
- Tizimga xizmat ko'rsatish oson va tez (agar valyuta o'zgarsa, tizim buni oddiy qayta qiyofalash bilan kutib olishga tayyor).

Shunday qilib, funktsional bo'lmagan talabning tabiati uning doimiy yoki vaqtinchalik ekanligini hal qilmaydi. Mijoz jamoayordamida qaror qabul qilishi kerak. Ideal vaziyatda katta ahamiyatga ega bo'lgan talablarni ajratib olib (haqiqatan ham kutilgan yaqin kelajakda tizimga katta ta'sir ko'rsatadigan o'zgarishlar) ularni vaqtinchalik, boshqalarni esa doimiy deb hisoblash kerak.

### 5.7. Majburiy va istalgan talablar

Talablar, shuningdek, *majburiy* yoki *istalgan* deb hisoblanishi mumkin, ya'ni har qanday yo'l bilan olinishi kerak bo'lganlar (*majburiy*) va rivojlanish jarayoniga katta muammo to'sqinlik qilmasa, olinishi mumkin bo'lgan talablar (*istalgan*).

Funktsional talablariga ko'ra, bu sinflash rivojlanishning ustuvor yo'nalishlarini ko'rsatadi. Agar shartnomada moslashuvchanlik mavjud bo'lib, faqat eng muhim foydalanish variantlari amalga

Nisbiylik	Yetuklik	sonlarning oldini olish uchun juda ehtiyot bo'lish kerakmi? Tizimning nozik qismlari rasmiy tasniflarmi? Tizimda xatolar yo'qligiga ishonch hosil qilish uchun kerakli testlarning intensivligi va turi qanday?
	Mavjudlik	Tizim uchun qaysi bir mavjudlik darajasi talab qilinadi? Qanday qilib bir vaqtning o'zida ko'plab murojaatlarni qo'llab-quvvatlash kerak? Kuniga necha soat? Yiliga necha kun?
	Xatolarga turg'unlik	Tashqi fitna holatlarida, masalan aloqa uzilishlari kabi nome'yoriy holatlarda, tizim qanday javob berishi kerak?
	Qayta tiklash	Nosozlik yuz berganda tizim avtomatik ravishda tiklanishi kerakmi? Qanday holatlarda yo'qolgan berilganlar va uzilgan jarayonlarni tiklash kerak?
Talabchanlik	Muvofiqlik, tanib olish	Potensial foydalanuvchi, dasturiy ta'minot qo'llanilishini tan olishi uchun uni qanday taqdim etish kerak? Dasturiy ta'minot qanday ixchamlangan bo'lishi kerak?
	O'rganish qobiliyati	Foydalanuvchi vakolatli bo'lishi uchun dasturiy ta'minotning mos tushunchalaridan qanday foydalanishi kerak?
	Ish qobiliyati	Dasturiy mahsulotni ishlatish va boshqarish uchun u qay darajada oson bo'lishi kerak? Tizim qanday yordam berishi kerak? Qaysi shakllardagi hujjatlar va qo'llanmalar mavjud bo'lishi kerak? Ular qanday yaratiladi? Ular

		qanday ma'lumotlarni taqdim qilishi kerak?
	Foydalanuvchi xatolaridan himoya	Foydalanuvchi xatolaridan qanday himoya qilish kerak?
	Aloqadan foydalanuvchining estetikasi	Interfeysni ta'minlash uchun loyihalash andozalarining qaysi qiziqarli va qoniqarli o'zaro harakati ta'minlanishi kerak?
	Mavjudlik	Odamlarning maxsus ehtiyojlarini hisobga olgan holda, mahsulotni qay darajada zukkolik bilan ishlab chiqarish kerak?
Ish samaradorligi	Vaqt harakati	Jarayonlar va funksiyalar bilan bog'liq vaqt cheklavlari qanday?
	Zaxiralardan foydalanish	Berilganlar omborining hajmi bo'yicha cheklavlari bormi? Energiya cheklavlari bormi? Aloqa tarmog'ining cheklavlari bormi?
	Hajm	Qayta ishlash samaradorligiga kelsak, kutilayotgan nominal qiymatlar va kritik qiymatlar qanday? Masalan, dasturiy ta'minot bir vaqtning o'zida 2000 tagacha foydalanuvchini qo'llab-quvvatlash uchun ishlab chiqarilishi mumkin, lekin bir vaqtning o'zida 10000 tagacha ishlovchi uchun murojaatlar kafolatlanganmi?
Xavfsizlik	Maxfiylik	Tizim ma'lumotlari va funksiyalari, faqat ularga kirish huquqiga ega bo'lganlar uchun qay darajada bo'lishi kerak?
	Butunlik	Insonlar va tizimlar tomonidan rux-

		satsiz o'zgarishlardan himoyalanish uchun berilganlar va funksiyalar qay darajada bo'lishi kerak?
	Rad etmaslik	Tizim unda saqlangan yozuvlarning haqiqiylikini, ularning mualliflari rad eta olmasligi uchun haqiqatni qay darajada ta'minlashi kerak?
	Hisobdorlik	Tizim tomonidan foydalanuvchi faoliyati qay darajada qayd etiladi? Qaysi bir ma'lumotlar saqlangan?
	Asllik	Tizimga kirgan foydalanuvchi haqiqatan ham o'zi ekanligining tasdig'i qay darajada kafolatlangan bo'lishi kerak?
Moslik	Birgalikda yashash	Dasturiy ta'minot qanday dasturiy mahsulotlar bilan birga bo'lishi kerak? Qaysi dasturlash vositalari va tillaridan qanday foydalanish kerak? Tizimni eski tizimlar bilan birga ishga tushirish shartmi?
	Moslik	Tizim bilan qanday boshqa dasturiy mahsulotlar o'zaro harakatlanishi kerak? Ular qanday tizimlarga berilganlarni yuborishlari kerak? Ular qanday tizimlardan berilganlarni olishi kerak?
Ta'minlanganlik	Modullik	Qaysi arxitekturadan foydalanish kerak? Qatlamlar? Bo'limlar? Tarkiblar? Veb-xizmatlarchi?
	Qayta foydalanish	Tizim boshqa tizimlarning qismlaridan yaratiladimi? Tizim loyihalashda uning qismlari kelajakda qayta ishlatilishini nazarda tutish shartmi?
	Tahlil	Koddagi nosozliklarni tuzatishni oson-

	qilishlik	lashtirish uchun, maxsus usullar yoki vositalardan foydalaniladimi?
	O'zgartirish qobiliyati	Tizimga kiritilgan o'zgarishlar yangi nuqsonlarga olib kelmasligiga kafolat beradigan, maxsus usullar yoki vositalardan foydalaniladimi?
	Tekshirish qobiliyati	Regressiya testlarini soddalashtirish uchun maxsus texnika va vositalardan foydalaniladimi? Avtomatlashtirilgan sinov vositalaridan foydalaniladimi? Rivojlanish vaqtida yaratilgan test zaxiralari qanday saqlanadi?
Ixchamlik	Moslashuvchanlik	Dastlab ishlab chiqilganlardan tashqari, dasturiy ta'minot boshqa kontekstlarga qay darajada moslashishi kerak? Uni qayta kompilyatsiya qilish kerakmi yoki shunchaki qayta sozlashmi? Tizimda nimani sozlash mumkin? Sozlanuvchi elementlarga misollar printerlar, valyuta, kompaniya siyosati, interfeys shriftlari va ranglari, til va boshqalar. Vaqtinchalik talablar, ehtimol sozlash elementlari bilan bog'liq bo'ladi.
	O'rnatish imkoniyati	O'rnatish uchun qanday manbalar taqdim etiladi? O'rnatish avtomatik bo'lishi kerakmi? Berilganlarni ko'chirish avtomatik ravishda bajariladimi?
	O'zgartirish	Bir xil maqsadli tizimlardan birini almashtirishda ikkinchisi qanday zaxiralarni taqdim etishi kerak? Tizim berilganlar va qiyofalarni yaratishda keng foydalaniladigan formatlarda amalga oshirishi shartmi, masalan,

		XMLda?
Samaradorlik	Samaradorlik	Haqiqiy foydalanish muhitida qaysi tijorat maqsadlariga erishish uchun tizim yordam berishi kerak? Ushbu maqsadlarni to'liq va to'g'ri bajarishi uchun tizim qay darajada javobgar bo'lishi kerak?
Samarador	Samarador	Tizim mijozga qanday investitsiya turlarini qaytarishni ta'minlashi kerak?
Qoniqish	Foyda	Foydalanuvchiga undan foydalanish oqibatlarini tushunishga yordam berish uchun, dasturiy ta'minot qanday ishlab chiqilishi kerak?
	Qoniqish	Tizim o'z foydalanuvchilariga qay darajada zavq bag'ishlashi kerak?
	Shinamlilik	Tizim foydalanuvchining jismoniy va psixologik qulayligini qay darajada saqlashi yoki yaxshilashi kerak?
	Ishonch	Manfaatdor tomonlar bu ishni bajarishiga ishonishlari uchun tizim qaysi tarzda ishlab chiqilishi kerak?
Xavfdan ozodlik	Iqtisodiy xavflarni kamaytirish	Tizim qanday moliyaviy xavfni (shu jumladan mulkiy va ma'naviy zararni) kamaytirishi kerak?
	Salomatlik va xavfsizlik xavfni kamaytirish	Tizim odamlar uchun qanday jismoniy xavfni kamaytirishi kerak?
	Ekologik xavfni kamaytirish	Tizim qanday ekologik va mulkiy xavflarni kamaytirishi kerak?
		Tizimdan qanday darajada samarali, natijador, xavf-xatarsiz va undan foy-

Kontekst gamrovi	Kontekstning to'liqligi	dalanish sharoitida foydalanuvchi qoniqish bilan foydalanishi kerak? Ana shu foydalanish sharoitlari qanday? Dasturiy ta'minotdan foydalanish bilan bog'liq huquqiy talablar bormi?
	Moslashuvchanlik	Dastlabki foydalanuvchilar uchun ishlab chiqilganidan farqli ravishda, tizim qay darajada samarali, natijador, xavf-xatarsiz va undan foydalanish sharoitida foydalanuvchi qoniqish bilan foydalanishi mumkin? Ushbu foydalanish sharoitlari qanday?

Ushbu ro'yxat keng bo'lsa-da, qaysi talablar haqiqatan ham muhimligini aniqlash qobiliyatini yaxshilash uchun jamoa, bu faqat tasnif ekanligini yodda tutishi kerak. Mavjud bo'lmagan talablarni izlashning hojati yo'q, masalan, mijozni dasturiy ta'minotni qay tarzda mijozga etkazish usuli uni umuman qayg'urmaydi.

Shuningdek, berilgan talab u yoki bu turga tegishli yoki yo'qligini muhokama qilish uchun vaqtni yo'qotmaslik tavsiya etiladi. Uning turi to'g'risida qaror qabul qilishdan ko'ra muhimroq narsa uning mavjudligini bilishdir, talablarni tasniflash bo'yicha uzoq munozaralar loyihaga muhim bilimlarni qo'shmaydi.

5.1- jadvalda keltirilgan talablarni keltirib chiqaradigan narsa oxir-oqibat ehtiyojlarni topish uchun yaxshi asosdir. Lekin bu savollarning barchasiga javob berish majburiy emas.

### 5.9. Nazorat uchun savollar

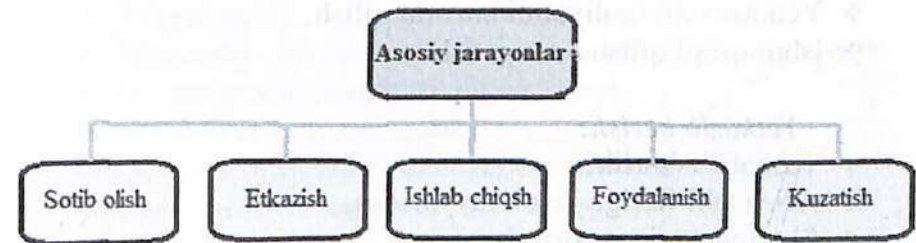
1. Funksional talablarni ta'riflang.
2. Funksional talablarning alohidaligin izohlang.
3. Nima uchun "talablarni aniqlash loyihalash emas" ligini, tushuntiring.

4. Talablarda nechta asosiy muammolar mavjud?
5. Meyerning "ochiq-yopiq" tamoyilini tusuntirib bering.
6. Ayon funksional talablar deganda nimani tushunasiz?
7. Yashirin funksional talablar deganda nimani tushunasiz?
8. Funksional bo'lmagan talablar deganda nimani tushunasiz?
9. Funksional bo'lmagan talablarning nechta turi mavjud?
10. Funksional bo'lmagan talablarning doimiyliigi nimadan iborat?
11. Funksional bo'lmagan talablarning vaqtinchaligi nimadan iborat?
12. Majburiy talablar deganda nimani tushunasiz?
13. Istalgan talabla, deganda nimani tushunasiz?
14. Qo'shimcha talablar, deganda nimani tushunasiz?
15. Qo'shimcha talablarning alohidaligi nimadan iborat?

## VI- BOB. AVTOMATIZATSIYALASHTIRILGAN TIZIMLARNI ISHLAB CHIQISH BOSQICHLARI

*Avtomatizatsiyalashtirilgan tizimlarni yaratish jarayoni* – bu belgilangan talablarga javob beradigan avtomatizatsiyalashtirilgan tizimni yaratish uchun zarur va yetarli bo'lgan ish bosqichlari davrida birlashtirilgan, vaqt o'tishi bilan buyurtma qilingan va o'zaro bog'liq bo'lgan to'plam.

Avtomatizatsiyalashtirilgan tizimlarni yaratish bosqichlari va bosqichlari ushbu natija bilan tugaydigan ishlarni oqilona rejalashtirish va tashkil etish nuqtayi nazaridan yaratilish jarayonining bir qismi sifatida ajralib turadi. 6.1- rasmda hayot siklining asosiy jarayonlari ko'rsatilgan.



6.1- rasm. Tizim dasturiy ta'minotining hayot siklining asosiy bosqichlari

Keyinchalik, hayot sikli jarayonlarining tarkibini batafsilroq ko'rib chiqamiz.

### 6.1. Asosiy jarayonlar

*Sotib olish:*

*Sotib olishga tashabbuskorlik:*

- ❖ Mijozning ehtiyojlarini aniqlash.
- ❖ Tizim talablarini tahlil qilish.
- ❖ Sotib olish to'g'risida qaror qabul qilish.
- ❖ Kerakli hujjatlarni tekshirish.
- ❖ Sotib olish rejasini tayyorlash.

***Ariza takliflarini tayyorlash:***

- ❖ Tizimga qo'yiladigan talablar.
- ❖ Dasturiy mahsulotlar ro'yxati.
- ❖ Shartlar va kelishuvlar.
- ❖ Texnik cheklovlar.

***Shartnomani tayyorlash va sozlash:***

- ❖ Yetkazuvchini tanlash amalini aniqlash.
- ❖ Yetkazuvchini tanlash.
- ❖ Shartnomani tayyorlash.
- ❖ Shartnomaga o'zgartirishlar kiritish.
- ❖ Yetkazuvchi faoliyatini nazorat qilish.
- ❖ Ishni qabul qilish va tugatish.

***Yetkazib berish:***

- ❖ Tashabbuskorlik.
- ❖ Ariza takliflariga javob tayyorlash.
- ❖ Shartnomani tayyorlash.
- ❖ Rejalashtirish.
- ❖ Bajarish va boshqarish.
- ❖ Tekshirish va baholash.
- ❖ Yetkazib berish va tugatish.

***Ishlab chiqish:***

- ❖ Tayyorgarlik.
- ❖ Talablarni tahlil qilish.
- ❖ Arxitekturani loyihalash
- ❖ Dasturiy mahsulot (DM) talablarini tahlil qilish.

- ❖ DM arxitekturasini loyihalash.
- ❖ DM arxitekturasining batafsil loyihasini yaratish.
- ❖ DMni kodlash va sinovdan o'tkazish.
- ❖ DMning integratsiyasi.
- ❖ DMning malaka sinovlari.
- ❖ Tizimning integratsiyasi.
- ❖ DMning malaka sinovlari.
- ❖ DMni o'rnatish.
- ❖ DMni qabul qilish.

***Foydalanish:***

- ❖ Tayyorgarlik.
- ❖ Amaliy sinov.
- ❖ Tizimning ishlashi.
- ❖ Foydalanuvchilarni qo'llab-quvvatlash.

***Kuzatish:***

- ❖ Tayyorgarlik.
- ❖ Muammolarni tahlil qilish va takomillashtirish so'rovlari.
- ❖ Dasturiy ta'minotni takomillashtirish.
- ❖ Tekshirish va qabul qilish.
- ❖ Dasturiy ta'minotni boshqa muhitga o'tkazish.
- ❖ Dasturni foydalanishdan olib tashlash.

**6.2. Yordamchi jarayonlar**

***Hujjatlar:***

- Tayyorgarlik.
- Dizayn va ishlab chiqish.
- Hujjatlarni chiqarish.
- Kuzatish.

***Qiyofani boshqarish:***

- Tayyorgarlik.

- Qiyofani aniqlash.
- Qiyofani nazorat qilish.
- Qiyofaning holatini hisobga olish.
- Qiyofani baholash.
- Chiqarishni boshqarish va yetkazib berish.

**Sifatni ta'minlash:**

- Tayyorgarlik.
- Mahsulot sifatini ta'minlash.
- Jarayon sifatini ta'minlash.
- Tizim sifatining boshqa ko'rsatkichlarini ta'minlash.

**To'g'riligini tekshirish:**

- Tayyorgarlik.
- To'g'riligini tekshirish.

**Sertifikatlash:**

- 2.5.1. Tayyorgarlik.
- 2.5.2. Sertifikatlash.

**Birgalikda baholash:**

- Tayyorgarlik.
- Loyihani boshqarishni baholash.
- Texnik baholash.

**Muammolarni hal qilish:**

- Tayyorgarlik.
- Muammolarni hal qilish.

**Audit:**

- Tayyorgarlik.
- Audit.

### 6.3. Tashkiliy jarayonlar

**Boshqarish jarayoni:**

- ✓ Tashabbuskorlik va boshqarish sohasini aniqlash.
- ✓ Rejalashtirish.
- ✓ Bajarish va nazorat.
- ✓ Tekshirish va baholash.
- ✓ Tugatish.

**Infratuzilmani yaratish jarayoni:**

- ✓ Tayyorgarlik.
- ✓ Infratuzilmani yaratish.
- ✓ Infratuzilmani kuzatish.

**Takomillashtirish jarayoni:**

- ✓ Jarayonni yaratish.
- ✓ Jarayonni baholash.
- ✓ Jarayonni takomillashtirish.

**O'quv jarayoni:**

- ✓ Tayyorgarlik.
- ✓ O'quv materiallarini ishlab chiqish.
- ✓ O'quv rejani amalga oshirish.

### 6.4. ATni yaratish bosqichlari

**AC uchun talablarni shakllantirish:**

- Ob'yektni tekshirish va ATni yaratish zarurligini asoslash.
- AT uchun foydalanuvchi talablarini shakllantirish.
- Bajarilgan ishlar to'g'risida hisobot va ATni ishlab chiqish uchun ariza berish (taktik va texnik topshiriq).

**Hujjatlar:**

- Eskiz loyihasi bayonoti.
- Eskiz loyihasiga tushuntirish xati.

**ATning g'oyasini ishlab chiqish:**

- Ob'jektning o'rganish.
- Kerakli ilmiy-tadqiqot ishlarini olib borish.
- Foydalanuvchi talablariga javob beradigan AT g'oyasining variantlarini ishlab chiqish.
- Bajarilgan ishlar to'g'risida hisobot berish.

**Texnik topshiriq:**

- ATni yaratish uchun texnik topshiriqni ishlab chiqish va tasdiqlash.
- Ixtisoslashtirilgan (yangi) texnik vositalarni ishlab chiqish uchun texnik topshiriqlar.

### 6.5. Xomaki loyiha

Tizim va uning qismlari uchun xomaki loyiha yechimlarini ishlab chiqish.

AT va uning qismlari uchun hujjatlarni ishlab chiqish:

- Tashkiliy tuzilish chizmasi.
- Texnik vositalar tarkibiy majmuasining chizmasi.
- Funktsional tuzilish chizmasi.
- Ixtisoslashtirilgan (yangi) texnik vositalarni ishlab chiqish bo'yicha vazifalar ro'yxati.
- Avtomatizatsiyalashtirish chizmasi.

### 6.6. Texnik loyiha

**Texnik loyiha tarkibi:**

- Texnik loyiha mundarijasi.
- Sotib olinuvchi mahsulotlar ro'yxati.
- Kiruvchi signallar va berilganlar ro'yxati.
- Chiquvchi signallar (hujjatlar) ro'yxati.
- Tizimni yaratish bilan bog'liq bo'lgan loyihadagi qurilish, elektrotexnika, sanitariya-texnik va boshqa bo'limlarini ishlab chiqish bo'yicha vazifalar ro'yxati
- Texnik loyihaga tushuntirish xati.

- Avtomatizatsiyalashtirilgan funksiyalarning tavsifi.
- Vazifalarni belgilash tavsifi (vazifalar to'plami).
- Tizimning axborot ta'minotining tavsifi.
- Axborot bazasini tashkil etish tavsifi.
- Sinflash va kodlash tizimlarining tavsifi.
- Ma'lumotlar massivining tavsifi.
- Texnik vositalar majmuasining tavsifi.
- Dasturiy ta'minot tavsifi.
- Algoritmning tavsifi (loyiha amaliyoti).
- Tashkiliy tuzilmaning tavsifi.
- Joylashuv rejasi.
- Uskunalar va materiallar ro'yxati.
- Mahalliy smeta hisobi.
- Tizimning ishonchliligini loyiha asosida baholash.
- Hujjat shaklini chizish (video kadr).

### Tizim va uning qismlari uchun loyihaviy yechimlarni ishlab chiqish:

- ❖ AT va uning qismlari uchun hujjatlarni ishlab chiqish.
- ❖ AT yaratish bo'yicha va/yoki texnik talablarni ishlab chiqish uchun mahsulotlarni etkazib berish bo'yicha hujjatlarni ishlab chiqish.
- ❖ Avtomatizatsiyalashtirish ob'ekti loyihasining qo'shni qismlarining vazifalarini ishlab chiqish.

### 6.7. Ish hujjatlari

Dasturlarni ishlab chiqish yoki moslashtirish.  
Tizim va uning qismlari uchun ishchi hujjatlarni ishlab chiqish:

- Asl nusxalar egalarining ro'yxati.
- Foydalanish hujjatlari ro'yxati.
- Uskunaning tasnifi.
- Materiallarga bo'lgan ehtiyoj ro'yxati.
- Kompyterdagi axborot tashuvchilarning ro'yxati.

- Kiruvchi berilganlar massivi.
- BBning ro'yxati.
- Chiquvch berilganlar tarkibi.
- Mahalliy smeta.
- Loyihalashni avtomatizatsiyalashtirish texnologiyasi.
- Texnologik yo'riqnomalar.
- Foydalanuvchi qo'llanmasi.
- BBni shakllantirish va saqlash bo'yicha ko'rsatmalar.
- Ulanishlar va ulanishlar jadvali.
- Tizimning tuzilmaviy bo'linish chizmasi.
- Umumiy ko'rinishdagi rasm.
- Texnik vositalarni o'rnatish chizmasi.
- Chizmalı diagramma.
- Texnik vositalarning tarkibiy majmuaviy chizmasi.
- Uskunalarni joylashtirish rejasi.
- Berilganlarni qayta ishlash texnologik jarayonining tavsifi.
- Tizimning umumiy tavsifi.
- Tarkiblar, majmualar, avtomatizatsiyalashtirish vositalari, quyi tizimlar va tizimlarning sinov dasturi va uslubiyati.

### 6.8. Foydalanishga topshirish

- Avtomatizatsiyalashtirish ob'yektini ishga tushirish uchun tayyorlash.
- Xodimlarni tayyorlash.
- Taqdim etilgan mahsulotlar – dasturiy va texnik vositalar, dasturiy va texnik majmualar, axborot mahsulotlari bilan jihozlash.
- Qurilish-montaj ishlari.
- Ishga tushirish ishlari.
- Dastlabki sinovlarni o'tkazish.
- Eksperimental amal.
- Qabul sinovlarini o'tkazish.

### 6.9. Kuzatish

- Kafolat majburiyatlariga muvofiq ishlarni bajarish.

### 6.10. Nazorat uchun savollar

1. Avtomatizatsiyalashtirilgan tizimlarni yaratish jarayonini ta'riflang.
2. Asosiy jarayondagi ishlarni ta'riflang.
3. Yordamchi jarayondagi ishlarni ta'riflang.
4. ATni yaratishdagi yirik bosqichlar nimalardan iborat?
5. Xomaki loyiha davridagi vazifalar nimalardan iborat?
6. Texnik loyiha davridagi vazifalar nimalardan iborat?
7. Ishchi hujjatlar nimalardan iborat?
8. Foydalanishga topshirish davrida qanday ishlar amalga oshiriladi?
9. Kuzatish davrining ahamiyati nimadan iborat?

## VII- BOB. DASTURIY TA'MINOT HAYOT SIKLINING MODELLARI

*Dasturiy ta'minot (DT) hayot siklining modeli* – bu jarayonlar, ishlar va dasturiy mahsulotni ishlab chiqish, ishlatish va qo'llab-quvvatlashni o'z ichiga olgan vazifalardan iborat bo'lib, unda talablarni belgilashdan boshlab, toki foydalanishni to'xtatishgacha bo'lgan tizim hayotini qamrab oladi.

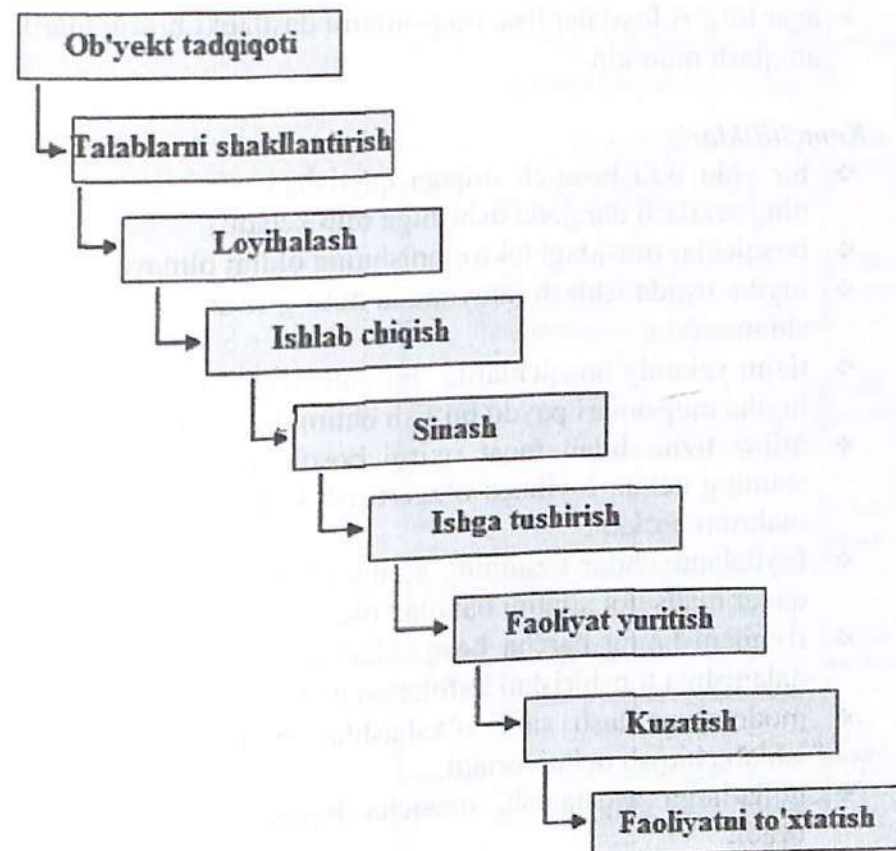
*Dasturiy ta'minot hayot siklining modeli (Software Life Cycle Model, SLCM)* dasturiy ta'minotni ishlab chiqish bo'yicha harakatlar qanday amalga oshirilishini, ushbu harakatlarning "ketma-ketligini" tavsiflash orqali undagi vositalar chizma shaklida tushuntiriladi.

DT hayot siklining asosiy modellari va yodashuvlari:

1. Pog'onali model.
2. Pog'onali-qaytishli yondoshuv.
3. Qoplovchi jarayonli pog'onali yondoshuv.
4. Inkrement model.
5. Aylanma (spiral) model.
6. Tortishuv modeli.

### 7.1. Pog'onali model

*Pog'onali (waterfall)* model dasturiy ta'minotni ishlab chiqish jarayonining modeli bo'lib, unda ishlab chiqish jarayoni talablarini tahlil qilish, loyihalash, amalga oshirish, sinovdan o'tkazish, integratsiya va qo'llab-quvvatlash bosqichlaridan ketma-ket o'tadigan oqimga o'xshaydi (7.1- rasm).



7.1- rasm. Pog'onali model

### 7.2. Pog'onali modelning afzalliklari va kamchiliklari

#### *Afzalliklari:*

- iste'molchilarga yaxshi ma'lum;
- "aniqlik" sharoitida yaxshi ishlaydi;
- tushunish oson;
- qo'llash sodda;
- talablarning barqarorligi bilan ajralib turadi;
- loyihani yaxshi nazorat qilish qulay;

➤ agar to'g'ri foydalanilsa, nuqsonlarni dastlabki bosqichlarda aniqlash mumkin.

**Kamchiliklari:**

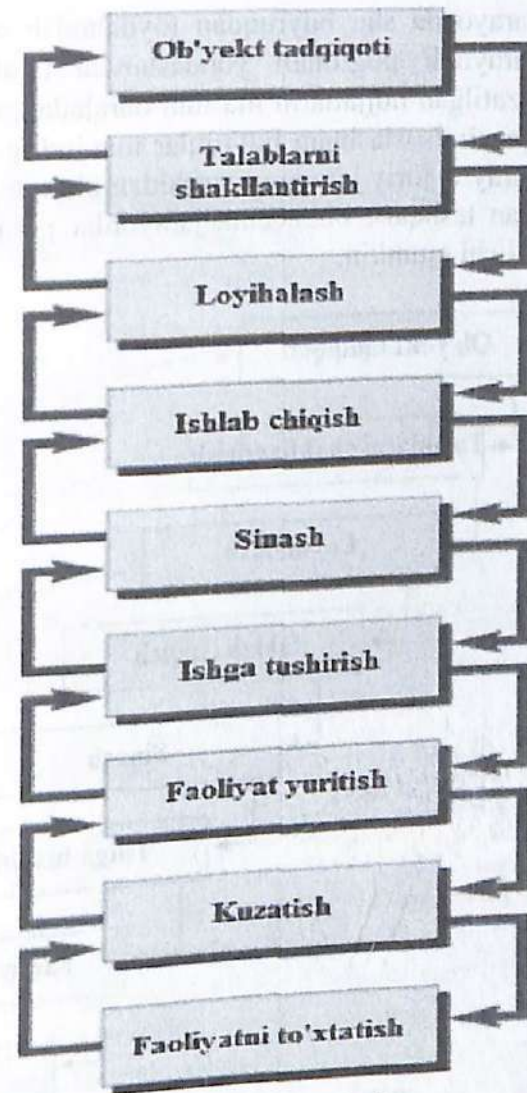
- ❖ bir yoki ikki bosqich orqaga qaytishga urinish xarajatlarining sezilarli darajada oshishiga olib keladi;
- ❖ bosqichlar orasidagi takrorlanishning oldini olmaydi;
- ❖ loyiha ustida ishlash jarayonoda noto'g'ri taassurot qoldirishi mumkin;
- ❖ tizim yakuniy bosqichlarda "bir zarbada" yig'iladi, natijada loyiha nuqsonlari paydo bo'lish ehtimoli yuqori;
- ❖ mijoz tizim bilan faqat oxirgi bosqichlarda tanishadi va shuning uchun loyihaga o'zgartirish kiritish imkoniyatidan mahrum bo'ladi;
- ❖ foydalanuvchilar tizimning so'nggi bosqichlari tugaguniga qadar mahsulot sifatini baholay olmaydilar;
- ❖ rivojlanishning barcha bosqichlaridan o'tish loyihani foydalanishga topshirishni kafolatlamaydi;
- ❖ modelni qo'llash xavfi o'xshashlari bo'lmagan tizimlarni ishlab chiqish uchun ortadi;
- ❖ hujjatlarga asoslanish, ortiqcha hujjatlar sonini ko'paytiradi;
- ❖ dasturiy tizimni ishlab chiqish bir vaqtning o'zida amalga oshiriladi, bu ishlab chiqarishni qismlarga ajratishga imkon bermaydi, natijada esa loyihani moliyalashtirish bilan bog'liq muammolarni keltirib chiqaradi.

**7.3. Pog'onali-qaytishli yondashuv**

Pog'onali modelning asosiy kamchiligi uning qayishqoq emasligida. Aynan ana shu kamchilikni pog'onali-qaytishli yondashuv engadi. Unda avvalgi bosqichga qaytish va avval qabul qilingan qarorlarga aniqlik kiritish yoki qayta ko'rish imkoniyati berilgan.

Pog'onali-qaytishli yondashuv dasturiy ta'minotni yaratishdagi takrorlash xususiyatlarini akslantiradi. Bunda dasturiy ta'minotni yaratishdagi aniq jarayon akslanadi, shu jumladan natija olishdagi sezilarni kechikish ham. Kechikish sababi esa qaytishdagi xatolarni to'g'rilash bilan bog'liq bo'ladi.

**Pog'onali-qaytishli yondashuv** istalgan natijaga erishilgunga qadar har bir jarayonning ketma-ket takrorlanishini ta'minlaydi (7.2- rasm). Har bir qaytish tugallangan bosqich bo'lib, uning oxirida muayyan natijaga erishiladi. Ehtimol, bu natija oraliq bo'ladi, barcha kutilgan funksionallikni amalga oshirmaydi.

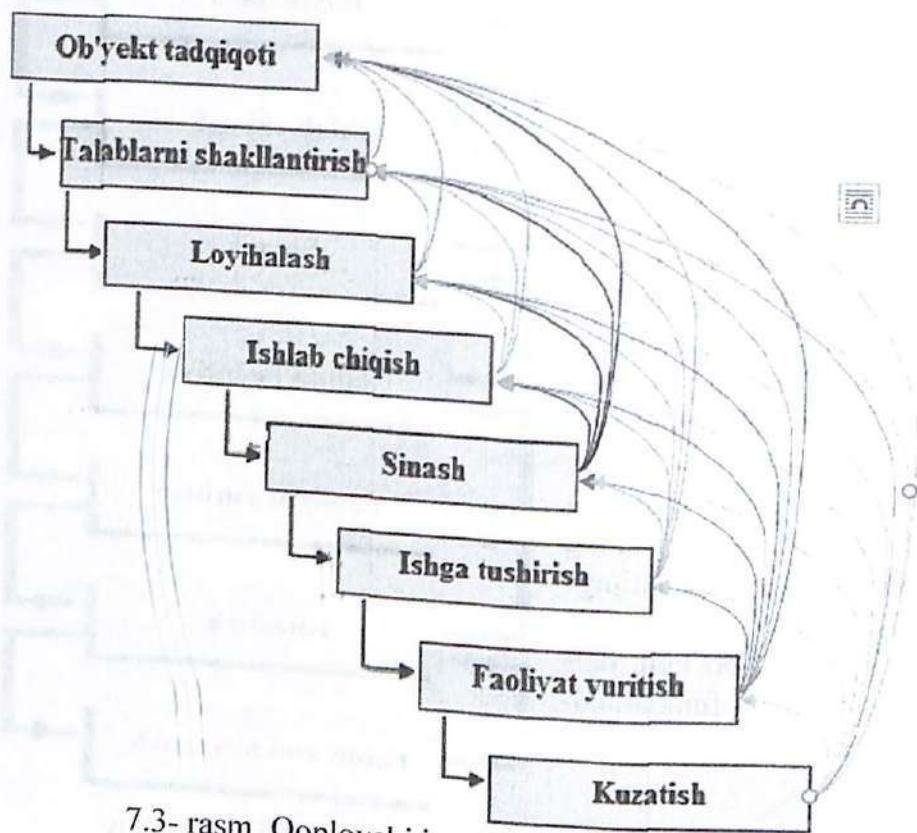


7.2- rasm. Pog'onali-qaytishli yondashuv

**7.4. Qoplovchi jarayonli pog'onali yondashuv**

Klassik pog'onali yondashuv har bir jarayonni alohida buyruq bilan bajarishga imkon beradi. Keyingi islab chiqishdagi xuddi shu

jarayonda shu buyruqdan foydalanish oqilona bo'ladi. Qoplovchi jarayonli pog'onali yondashuvda (waterfall with overlapping), uzatilgan hujjatlarni ma'lum darajada kamaytirishga imkon beradigan ixtisoslashgan buyruqlar mavjudligini nazarda tutadi. Keyingi jarayon joriy jarayon tugashidan oldin boshlanadi (7.3- rasm). Bundan tashqari, bir nechta jarayonlar parallel ravishda amalga oshirilishi mumkin.



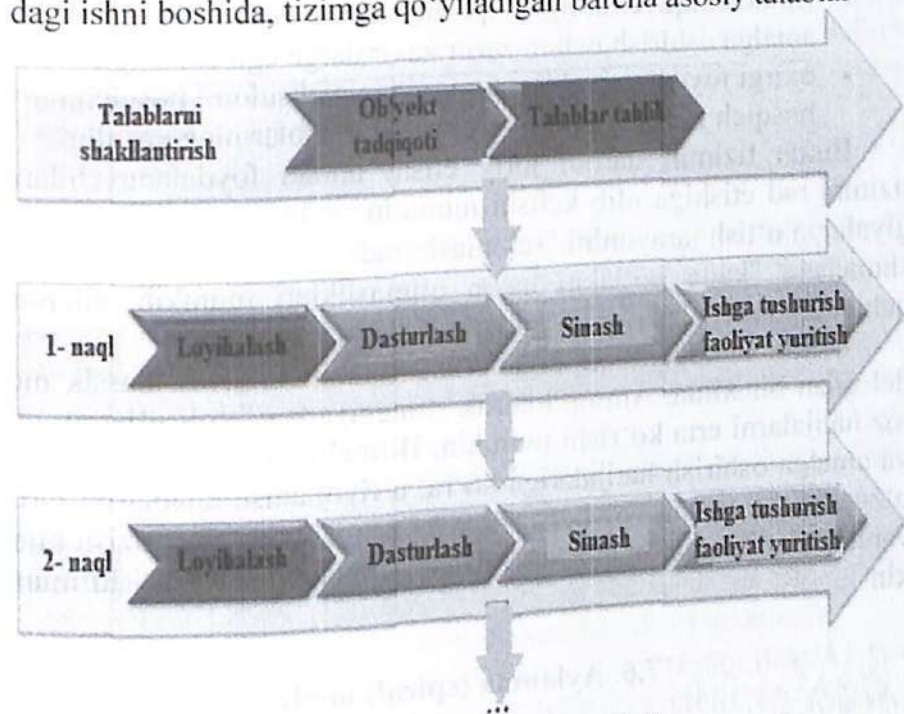
7.3- rasm. Qoplovchi jarayonli pog'onali yondashuv

### 7.5. Inkrement (orttirma) model

Orttirma model bosqichlarning chiziqli ketma-ketligi bilan, bir nechta naqlarda rejalashtirilgan takomillashtirish bilan, ATni ish-

lab chiqishni nazarda tutadi (7.4- rasm).

Ob'yekt to'liq tadqiqot va tahlil qilingandan keyin, loyiha ustidagi ishni boshida, tizimga qo'yiladigan barcha asosiy talablar



7.4- rasm. Orttirma model

aniqlanadi, shundan so'ng uni ishlab chiqish naqlar ketma-ketligini yaratish shaklida amalga oshiriladi. Bundan tashqari, har bir naql to'liq va ishlaydigan dasturiy mahsulot bo'ladi. Birinchi naql rejalashtirilgan xususiyatlarning bir qismini amalga oshiradi, keyingi naql qo'shimcha funksiyalarni amalga oshiradi va h.k., va bu jarayon to'liq tizim yaratilmaguncha davom etadi. Hayot siklining ushbu modeli murakkab va majmuaviy tizimlarni ishlab chiqishda o'rinlidir, mijoz tomonidan ham, ishlab chiquvchi tomonidan ham, ular uchun yakuniy natija (axborot tizimi) nimani anglatishi kerakligi to'g'risida aniq tasavvur mavjuddir. Naqlar turli sabablarga ko'ra ishlab chiqiladi:

- mijozning qimmatli loyihani darhol moliyalashtirish imkoniyatining yo'qligi;
- ishlab chiquvchining qisqa vaqt ichida murakkab loyihani amalga oshirish uchun zarur zaxiralarga ega emasligi;
- oxirgi foydalanuvchilar tomonidan mahsulotni bosqichma-bosqich joriy etish va o'zlashtirish talablarining yo'qligi.

Butun tizimni darhol joriy etish, undan foydalanuvchilarni tizimni rad etishiga olib kelishi mumkin va bu esa yangi texnologiyalarga o'tish jarayonini "sekinlashtiradi". Majoziy ma'noda, ular shunchaki "katta bo'lakni hazm qilmasliklari mumkin, shuning uchun uni maydalash va bo'laklarga bo'lish kerak".

Ushbu strategiyaning afzalliklari va kamchiliklari klassik model bilan bir xilda. Ammo klassik strategiyadan farqli o'laroq, mijoz natijalarni erta ko'rishi mumkin. Birinchi naqlini ishlab chiqish va amalga oshirish natijalariga ko'ra, u rivojlanish talablarini biroz o'zgartirishi, undan voz kechishi yoki yangi shartnoma tuzish bilan yanada rivojlangan mahsulotni ishlab chiqishni taklif qilishi mumkin.

### 7.6. Aylanma (spiral) model

Barri Boem tomonidan yaratilgan *aylanma model* (evolyutsion yoki yaqinlashuvchi model) naqlar ketma-ketligi shaklida rivojlanishni nazarda tutadi, ammo loyihaning boshida barcha talablar aniqlanmagan (7.5- rasm) bo'ladi. Talablar naqlarni ishlab chiqish natijasida aniqlanadi. Yanada umumiyroq ta'rif, *aylanma model* – bu ishlab chiqarilgan mahsulot elementlarini qayta ishlash uchun har qanday oldingi jarayonga qaytishga imkon beradigan tizimni ishlab chiqishning hayotiy siklidagi jarayonlar modeli.

Hayot siklining ushbu modeli innovatsion tizimlarni ishlab chiqishda o'rinlidir. Loyiha ustida ishlashning boshida mijoz va ishlab chiquvchida yakuniy mahsulot haqida aniq tasavvurga ega emas (talablarni aniq belgilash mumkin emas) yoki loyihani muvaffaqiyatli amalga oshirishga yuz foiz ishonch yo'q, chunki xatar-

lar juda katta. Shu munosabat bilan tizimga qo'yilgan talablarni o'zgartirgan holda uni qismlarga bo'lib ishlab chiqish yoki yanada rivojlantirishni rad etish to'g'risida qaror qabul qilinadi. 6- rasmdan ko'rinib turibdiki, loyihani ishlab chiqish nafaqat amalga oshirish bosqichidan keyin, balki xavfni tahlil qilish bosqichidan keyin ham yakunlanishi mumkin.

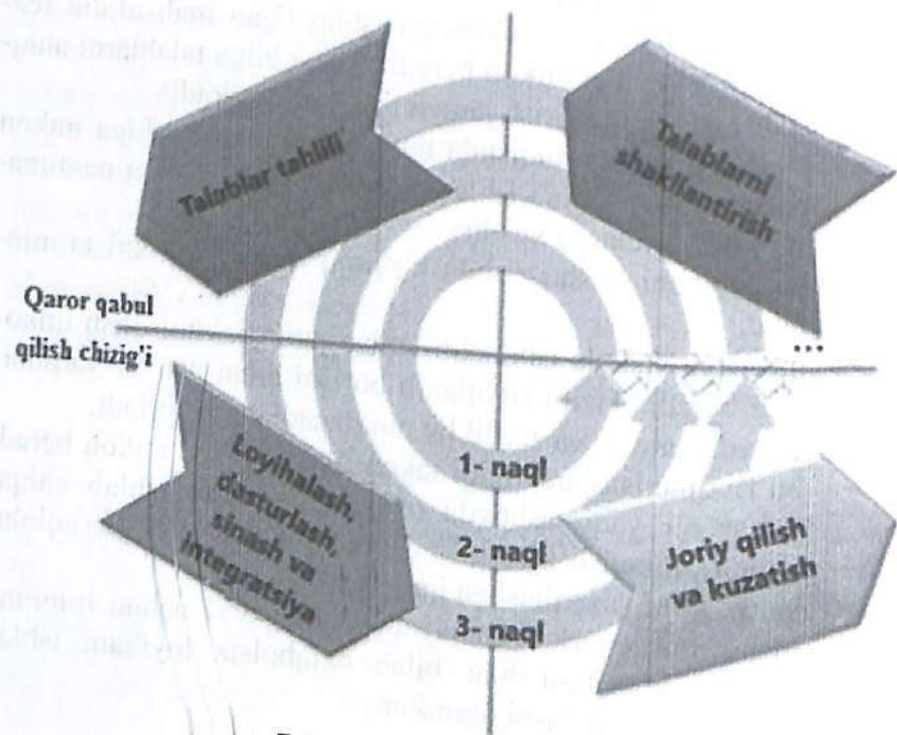
#### *Modelning afzalliklari:*

- bu tizim foydalanuvchilariga ishlaydigan mahsulotni tezroq ko'rsatishga imkon beradi va shu bilan talablarni aniqlashtirish va to'ldirish jarayonini faollashtiradi;
- bu ATni ishlab chiqishda talablarning o'zgarishiga imkon beradi, bu ko'pgina ishlanmalarga, shu jumladan namunaviy ishlanmalarga xosdir;
- bu loyihani boshqarishda ko'proq qayishqoqlikni ta'minlaydi;
- bu bizga yanada ishonchli va barqaror tizimni olish imkonini beradi. Tizim rivojlanib borishi bilan har bir yaqinlashuvda xatolar va zaifliklar aniqlanadi va tuzatiladi;
- bu rivojlanish jarayonini takomillashtirishga imkon beradi – har bir yaqinlashuvda o'tkazilgan tahlil ishlab chiqarishda nimani o'zgartirish kerakligini va keyingi yaqinlashuvda uni yaxshilashga imkon beradi;
- mijozning xatarlari kamayadi. Mijoz o'zi uchun minimal moliyaviy yo'qotishlar bilan istiqbolsiz loyihani ishlab chiqishni yakunlashi mumkin.

#### *Modelning kamchiliklari:*

- ❖ loyihani rivojlantirish istiqbollari ishlab chiquvchida noaniqlikni kuchaytiradi. Ushbu kamchilik modelning oldingi afzalliklaridan kelib chiqadi;
- ❖ umuman olganda, butun loyihani vaqtincha va zaxirali rejalashtirish amallari qiyin. Ushbu muammoni hal qilish uchun hayot siklining har bir bosqichi uchun, vaqt

chegaralarini kiritish kerak. O'tish rejalashtirilgan barcha ishlar bajarilmasa ham, rejaga muvofiq amalga oshiriladi. Reja oldingi loyihalarda olingan statistik berilganlar va ishlab chiquvchilarning shaxsiy tajribasi asosida tuziladi. Hozirgi vaqtda, hayot sikli aylanma modelidan foydalanishda, tavsiya etilishi mumkin bo'lgan bir nechta dasturiy ta'minotni ish-



7.5- rasm. Aylanma model

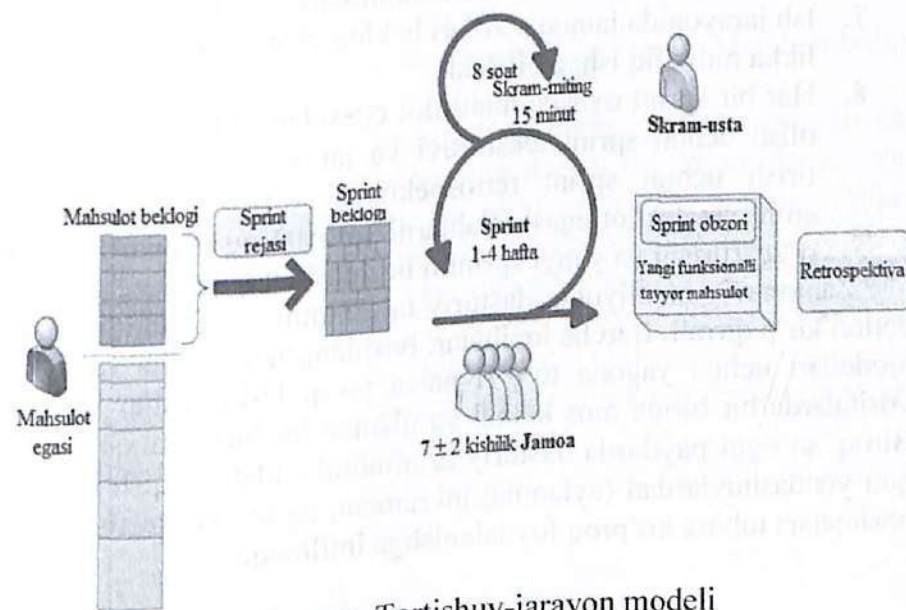
lab chiqish uslubiyatlari mavjud. Ulardan eng mashhurlari tezkor ilovalarni ishlab chiqish uslubiyati (Rapid Application Development, RAD) va Кент Бек muallifligidagi ekstremal dasturlashdir (eXtreme Programming, XP).

## 7.7. Tortishuv modeli

**Tortishuv (Scrum)** – dasturiy ta'minotni qayishqoq tarzda ishlab chiqish. Tortishuv modeli empirik usulga asoslangan va ishlab chiqish murakkab bo'lgan muhit uchun mo'ljallangan yuqori qiymatli mahsulotlar. **Tortishuv** agile g'oyasining yana bir timsolidir va majburiy bo'lgan bir qator qoidalarga asoslanadi (7.6-rasm).

**Scrum**-da ishlarni tashkil etishning asosiy tamoyillari va xususiyatlari [14]:

1. Barcha kerakli mutaxassislardan iborat kichik funksional jamoalar shakllantiriladi.



7.6- rasm. Tortishuv-jarayon modeli

2. Maxsus rollar ajratiladi:
  - i. scrum ustasi, jamoadagi jarayonlarga va konstruktiv muhitga rioya qilish uchun javobgardir;

- ii. mahsulot egasi barcha manfaatdor tomonlarni yig'ib, tablalar va ularning ustuvorliklari uchun javobgar bo'ladi.
  - 3. Tablalar kichik, iloji boricha mustaqil qismlarga bo'linadi. Natijada mahsulotning "beklogi" paydo bo'ladi. Keyin beklog elementlari ularning ahamiyati bo'yicha tartiblanadi va ularning mehnat xarajatlari baholanadi.
  - 4. Ish 1 dan 4 haftagacha qisqa muddatli takroriy yaqinlashuvlar bilan amalga oshiriladi, ularning har birining oxirida mahsulotning tugallangan funksionali – o'sishi qo'yiladi.
  - 5. Jamoa, o'z tezligiga ko'ra, vaqtni o'zgartirmaydigan yaqinlashish – sprint uchun vazifalarni to'playdi.
  - 6. Har kuni skrum-yig'ilish bo'lib o'tadi, unda jamoa o'z ishlarini sinxronlashtiradi va muammolarni muhokama qiladi.
  - 7. Ish jarayonida jamoa a'zolari beklog elementlarini ustuvorlikka muvofiq ishga olishadi.
  - 8. Har bir sprint oxirida mahsulot egasidan fikr-mulohazalarni olish uchun sprint tekshiruvi va jarayonlarni optimallashtirish uchun sprint retrospektivasi o'tkaziladi. Shundan so'ng, mahsulot egasi tablalarni va ularning ustuvorliklarini o'zgartirishi va yangi sprintni boshlashi mumkin.
- Zamoraviy amaliyotda dasturiy ta'minotni ishlab chiqish modellari ko'p qirrali. Barcha loyihalar, boshlang'ich shartlar va to'lov modellari uchun yagona to'g'ri narsa yo'q. Uslubiyatlar qisman vositalarda bir-biriga mos keladi va qisman bir-biriga o'xshashdir. Biroq, so'nggi paytlarda dasturiy ta'minotni ishlab chiqish qayish-qoq yondashuvlardan (aylanma, inkrement, agile, XP, tortishuv va boshqalar) tobora ko'proq foydalanishga intilmoqda.

### 7.8. RAD uslubiyati

RAD uslubiyati bo'yicha ishlab chiqarishda odatda 3 ta elementni o'z ichiga olgan jarayon anglatadi [10]:

- ✚ kichik dasturchilar jamoasi (10 kishigacha);

- ✚ qisqa, ammo puxta ishlab chiqilgan ishlab chiqarish jadvali (2 oydan 6 oygacha);
  - ✚ dastur shakllana boshlagach, yaratuvchilar buyurtmachi bilan mahsulotga bo'lgan tablalarni o'zaro aloqada olgan holda, amalga oshiradigan takroriy sikl.
- Muayyan buyurtmachi uchun ishlab chiqiladigan nisbatan kichik loyihalar uchun qo'llaniladi.
- Aylanma hayot sikli modeliga xos xususiyatlardan tashqari, RAD uslubiyati har bir yaqinlashuvda quyidagilardan foydalanishni o'z ichiga oladi:

- tablalarni shakllantirish va tahlil qilish uchun CASE vositalari, tizimni loyihalash, BB tuzilishi va avtomatik ravishda dastur kodini yaratish, shuningdek dasturiy ta'minotni avtomatik sinovdan o'tkazishni;
- tizimning vizual ishlab chiqishini ta'minlaydigan asbobiy vositalarni;
- ob'yektga yo'naltirilgan yondashuvni qo'llab-quvvatlovchi asbobiy vositalar. Ushbu vositalar ob'yektlar to'plami – xususiyatlar (atributlar) va xatti-harakatlar (usullar) bilan tavsiflangan haqiqiy dunyo ob'yektlari shaklida soha fazosining tavsifini yaratishga imkon beradiganlarni;
- voqeliklarni dasturlashni ta'minlaydigan asbobiy vositalar. Ilovaga kiritilgan har bir ob'yekt hodisalarni yaratishi va boshqa ob'yektlar tomonidan yaratilgan hodisalar ta'siriga javob berishi mumkin bo'lganlarni;
- o'zi hamda boshqalar tomon ishlab chiqarilgan tayyor yechimlar kutubxonalarini.

### 7.9. XP uslubiyati

Ushbu yondashuv noaniq yoki tez o'zgarib turadigan talablar sharoitida kichik va o'rta guruhlar tomonidan axborot tizimlarini ishlab chiqishga qaratilgan [13].

XP rivojlanishining o'ziga xos xususiyatlari:

- ✓ naqlar va takomillashtirishlarning tez-tez o'zgarishi (takrorlanish davomiyligi soat va daqiqagacha, odatda – 2 hafta; RADda-kamida 2 oy);
- ✓ buyurtmachi bilan uzluksiz aloqa;
- ✓ oddiy loyihalash – ishlab chiqishda har doim eng ko'p oddiy yechim tanlanadi;
- ✓ tizimni har bir vaqt uchun imkon qadar sodda tarzda loyihalash kerak. Interfeys qanchalik sodda bo'lsa, foydalanuvchilar tizimni tezroq va yaxshiroq o'zlashtiradilar;
- ✓ kodga jamoaviy egalik – kodning bir qismini yaxshilash imkoniyatini ko'rgan har bir dasturchi buni istalgan vaqtda amalga oshirishi mumkin. Bu kodga bir xil standartlar va qoidalarni qo'llashni nazarda tutadi;
- ✓ juftlikda dasturlash – ikkita dasturchiga bitta kompyuter. Ulardan biri to'g'ridan-to'g'ri dasturlashtirsa, ikkinchisi talablarni (funktsiyalar, BB va boshqalar) amalga oshirish masalalarini o'ylab turadi;
- ✓ tizimni uzluksiz va kesishgan loyihalash, ishlab chiqish, integratsiya va sinovdan o'tkazish.

**7.10. Nazorat uchun savollar**

1. Dasturiy ta'minot hayot siklining modelini ta'riflang.
2. Dasturiy ta'minot hayot siklining asosiy modellarini ayting.
3. Pog'onali modelni ta'riflang.
4. Pog'onali modelning afzalliklari nimadan iborat?
5. Pog'onali modelning kamchiliklari nimadan iborat?
6. Pog'onali-qaytishli yondoshuvni ta'riflang.
7. Qoplovchi jarayonli pog'onali yondashuvni ta'riflang.
8. Inkrement modelni ta'riflang.
9. Ortirma modelning afzalliklari nimadan iborat?
10. Ortirma modelning kamchiliklari nimadan iborat?
11. Aylanma modelni ta'riflang.
12. Tortishuv modelini ta'riflang.

13. Tortishuv modelining xususiyatlari nimadan iborat?
14. RAD uslubiyatini ta'riflang.
15. XP uslubiyatini ta'riflang.

## VIII- BOB. AXBOROT TIZIMLARINI YARATISHNI TASHKILLASHTIRISH

### 8.1. Axborot tizimlarini loyihalash

ATni loyihalash uchta asosiy sohani qamrab oladi:

- BBda amalga oshiriladigan berilganlar ob'yektlarini loyihalash;
- berilganlarga yo'naltirilgan so'rovlarning bajarilishini ta'minlaydigan dasturlar, ekran shakllari, hisobotlarni loyihalash;
- muayyan muhit yoki texnologiyani hisobga olish, ya'ni tarmoq topologiyasi, asbobiy vositalarning qiyofasi, ishlatiladigan "fayl-server" yoki "mijoz-server" arxitekturasi, parallel ishlov berish, taqsimlangan berilganlar ishlavi va boshqalar.

ATni loyihalash har doim loyiha maqsadini aniqlashdan boshlanadi. Umuman olganda, loyihaning maqsadi tizimni ishga tushirish paytida va uning hayot sikli davomida ta'minlashni o'z ichiga olgan bir qator o'zaro bog'liq muammolarni hal qilish sifatida belgilanishi mumkin:

- ✓ tizimning talab qilinuvchi funkcionalligi va uning o'zgaruvchan ish faoliyatiga moslashish darajasi;
- ✓ tizimning talab qilinuvchi o'tkazuvchanlik qobiliyati;
- ✓ tizimning so'rovga talab qilinuvchi javob berish vaqti;
- ✓ tizimning uzluksiz ishlashi;
- ✓ kerakli xavfsizlik darajasi;
- ✓ tizimni ishlatish va qo'llab-quvvatlash qulayligi.

Zamonaviy uslubiyatga ko'ra, ATni yaratish jarayoni uning hayot siklining barcha davrlarida bir qator muvofiqlashtirilgan modellarni qurish va ketma-ket o'zgartirish jarayonini o'z ichiga

oladi. Hayot siklining har bir davrida o'ziga xos modellar yaratiladi – tashkilotlar, ATga qo'yiladigan talablar, AT loyihasi, ilovalarga qo'yiladigan talablar va h.k.

Modellar loyiha guruhining ishchi guruhlari tomonidan shakllantiriladi, saqlanadi va loyihaning omborida to'planadi. Modellar yaratish, ularni boshqarish, o'zgartirish va jamoaviy foydalanish uchun taqdim etish maxsus dasturiy vositalar – CASE vositalaridan foydalangan holda amalga oshiriladi.

### 8.2. ATni kanonik loyihalash

AT kanonik loyihalashni tashkil etish asosan AT hayot siklining kaskad modelidan foydalanishga qaratiladi. Avtomatizatsiyalashtirish ob'yektining murakkabligiga va aniq ATni yaratishda hal qilinishi kerak bo'lgan vazifalar majmuasiga qarab, ishning davr va bosqichlari turli xil mehnat hajmiga ega bo'lishi mumkin. Loyihaning istalgan bosqichida ketma-ket bosqichlarni birlashtirish va hatto ulardan ba'zilarini chiqarib tashlash mumkin. Bundan tashqari, keyingi davrning ishini oldingi davr tugashidan oldin boshlashga ruxsat beriladi. Ishtirokchi tashkilotlar tomonidan amalga oshiriladigan AT yaratish davrlari va bosqichlari shartnomalarda va ishlarni bajarish uchun texnik shartlarda belgilanadi.

#### 1- davr. Ob'yekt tadqiqoti.

Dastlabki loyihalash davrida ishlar quyidagi bosqichlarga ajratiladi:

- avtomatizatsiyalashtirilayotgan ob'yektni har tomonlama va to'liq organish;
  - nimalar mavjudligini aniqlash;
  - infra-tuzilmani aniqlash;
  - nimalarga zarurat borligini tahlil qilish;
- 2- **AT uchun talablarni shakllantirish.**
- ob'yektni tekshirish va AT yaratish zaruriyatini asoslash;
  - ATda foydalanuvchi talablarini shakllantirish;

- bajarilgan ishlar va ishlab chiqish uchun taktik va texnik shartlar to'g'risida hisobot tayyorlash.
- 3- davr. AT loyihaviy konsepsiyasini ishlab chiqish.**
- zarur ilmiy-tadqiqot ishlarini olib borish;
  - foydalanuvchi talablariga javob beradigan AT g'oyasining variantlarini ishlab chiqish;
  - hisobot tayyorlash va g'oyani tasdiqlash.
- 4- davr. Texnik tasniflarni ishlab chiqish.**
- AT yaratish uchun texnik shartlarni ishlab chiqish va tasdiqlash;
  - infra-tuzilma texnik shartlarni ishlab chiqish va tasdiqlash.
- 5- davr. Boshlang'ich loyiha.**
- tizim va uning qismlari uchun dastlabki loyiha yechimlarini ishlab chiqish;
  - AT va uning qismlari uchun hujjatlar loyihasini ishlab chiqish.
- 6- davr. Texnik loyihalash.**
- tizim va uning qismlari uchun loyihaviy yechimlarni ishlab chiqish;
  - IP va uning qismlari uchun hujjatlarni ishlab chiqish;
  - butlovchi qismlarni yetkazib berish bo'yicha hujjatlarni ishlab chiqish va rasmiylashtirish;
  - loyiha qo'shni qismlarida uchun vazifalarni ishlab chiqish.
- 7- davr. Ishchi hujjatlar.**
- AT va uning qismlari uchun ishchi hujjatlarni ishlab chiqish;
  - dasturlarni ishlab chiqish va moslashtirish.
- 8- davr. Ishga tushirish.**
- avtomatizatsiyalashtirilayotgan ob'yektni tayyorlash;
  - xodimlarni tayyorlash;
  - ATni yetkazib beriladigan mahsulotlar (dasturiy ta'minot va texnik vositalar, dasturiy-asbob majmuasini, axborot mahsulotlari) bilan ta'minlash;
  - qurilish-montaj ishlari;

- ishga tushirish ishlari;
  - dastlabki sinovlarni o'tkazish;
  - sinov amaliyotini o'tkazish;
  - qabul qilish sinovlarini o'tkazish.
- 9- davr. ATni kuzatish.**
- kafolat majburiyatlariga muvofiq ishlarni bajarish;
  - kafolatdan keyingi xizmat.

### 8.3. Tashkilot tuzilmasini tekshirish va strategiyani aniqlash

**Tekshirish** – tashkilotning tashkiliy tuzilmasi, uning faoliyati va mavjud axborotni qayta ishlash tizimini o'rganish va tashxisli tahlil qilish. Tekshirish natijasida olingan materiallar quyidagilar uchun ishlatiladi:

- tizimlarni ishlab chiqish va bosqichma-bosqich joriy etishni asoslashda;
- tizimlarni ishlab chiqish uchun texnik shartlarni tuzishda;
- tizimlarning texnik va batafsil loyihalarini ishlab chiqishda.

Tekshirish bosqichida ikkita tarkibni ajratib ko'rsatish tavsiya etiladi – ATni amalga oshirish strategiyasini aniqlash va tashkilot faoliyatini batafsil tahlil qilish.

Tekshirishning birinchi bosqichining asosiy vazifasi, yuqori darajadagi avtomatlashtiriladigan ob'yektning aniqlangan funksiyalari va axborot elementlari asosida – loyiha aniq hajmini, maqsadini va vazifalarini baholashdan iborat. Ushbu vazifalar AT buyurtmachisi tomonidan mustaqil ravishda yoki konsalting tashkilotlarini jalb qilgan holda amalga oshirilishi mumkin. Ushbu bosqich tizimning asosiy foydalanuvchilari va tijorat mutaxassislari bilan yaqin hamkorlikni o'z ichiga oladi. O'zaro harakatning asosiy vazifasi buyurtmachi talablarini to'liq va aniq tushunishga ega bo'lish.

Odatda, kerakli ma'lumotlarni menejment, ekspertlar va foydalanuvchilar bilan savol-javoblar, suhbatlar yoki seminarlar orqali

olish mumkin.

Tekshirishning ushbu davrini tugatgandan so'ng, tizimni yaratishda mumkin bo'lgan texnik yondashuvlarni aniqlash va uni amalga oshirish xarajatlarini (asbobli ta'minot, sotib olingan dasturiy ta'minot va yangi dasturiy ta'minotni ishlab chiqish xarajatlari) baholash mumkin bo'ladi.

Strategiyani aniqlash bosqichining natijasida loyihaning texnik-iqtisodiy asoslagan hujjat yaratiladi. Unda buyurtmachi loyihani moliyalashtirishga rozi bo'lsa, nima olishi, ish jadvali bo'yicha qachon tayyor mahsulotni olishi va qancha miqdorda olishi aniqlanadi. Hujjatda nafaqat xarajatlarni, balki loyihaning foydasini, masalan, loyihaning o'zini oqlash muddati, agar kutilayotgan iqtisodiy samarani taxmin qilish mumkin bo'lsa, uni aks ettirish maqsadga muvofiqdir.

Ushbu hujjatning taxminiy mazmuni:

- loyiha muvaffaqiyatiga ta'sir ko'rsatishi mumkin bo'lgan cheklovlar, xavflar, muhim omillar;
- kelajakdagi tizim ishlashi kutilayotgan shartlar majmui – tizim arxitekturasi, asbob va dasturiy ta'minot zaxiralari, ish sharoitlari, texnik xizmat ko'rsatuvchi xodimlar va tizim foydalanuvchilari;
- alohida bosqichlarni yakunlash muddatlari, ishlarni qabul qilish/etkazib berish shakli, jalb qilinadigan zaxiralar, axborotni himoya qilish choralari;
- tizim tomonidan bajariladigan funksiyalarning tavsifi;
- tizimni rivojlantirish imkoniyatlari;
- tizimning axborot ob'yektlari;
- inson va tizim o'rasidagi interfeyslar va funksiyalarni taqsimlash;
- dasturiy ta'minot va dasturiy ta'minotning axborot tarkiblariga qo'yiladigan talablar, berilganlar bazasini boshqarish tizimi (BBBT)ga qo'yiladigan talablar;
- loyiha doirasida nimalar amalga oshirilmasligi.

#### 8.4. Tashkilot faoliyatining tahlili

Tashkilot faoliyatini batafsil tahlil qilish bosqichida boshqaruv funksiyalarini amalga oshirishni ta'minlaydigan vazifalar, tashkiliy tuzilma, shtat tarkibi va korxonani boshqarish bo'yicha ishlarning mazmuni, shuningdek yuqori boshqaruv tashilotlariga bo'ysunish xususiyati o'rganiladi. Ushbu bosqichda quyidagilarni aniqlash kerak bo'ladi:

- uslubiy va yo'riqnoma materiallariga asoslangan qism-tizimlarning tarkibi va masalalar ro'yxatini;
- muammolarni hal qilishda yangi usullardan foydalanish imkoniyatini.

Tahlilchilar ma'lumotlarni, bir-biriga bog'langan ikkita shaklda to'playdi va qayd etadi:

- *funksiyalarda* – tijoratda sodir bo'ladigan hodisalar va jaryonlar haqidagi ma'lumotlarni;
- *mohiyatlarda* – tashkilot uchun muhim bo'lgan va ma'lum bo'lgan narsalar haqidagi ma'lumotlarni.

Har bir funksional boshqaruv masalasi o'rganilayotganda, quyidagilar aniqlanishi kerak:

- masalaning nomi, muddati, echish vaqti va chastotasi;
- masalaning rasmiylashtirilganlik darajasi;
- masalani hal qilish uchun zarur bo'lgan axborot manbalari;
- ko'rsatkichlar va ularning miqdoriy tavsivlari;
- axborotni o'zgartirish tartibi;
- ko'rsatkichlarni hisoblashning joriy algoritmlari va mumkin bo'lgan nazorat usullari;
- axborotni to'plash, uzatish va qayta ishlashdagi mavjud vositalar;
- mavjud aloqa vositalari;
- masalani yechishdagi qabul qilingan aniqlik;
- masalani hal qilishning murakkabligi;
- dastlabki ma'lumotlarni va ularni qayta ishlash natijalarini hujjatlar shaklida taqdim etishning joriy shakllari;

- topshiriq bo'yicha natija ma'lumotlarining iste'molchilari.

Ushbu bosqichning eng ko'p mehnat talab qiladigan, yaxshi rasmiylashtirilgan vazifalaridan biri, bu tashkilotning hujjat aylanishini tavsiflashdir. Hujjatlar oqimini tekshirishda hujjat yo'nalishi chizmasi tuziladi, unda quyidagilarni aks ettirish kerak:

- hujjatlar soni;
- hujjat ko'rsatkichlarini shakllantirish joyi;
- hujjatlarni shakllantirish jarayonida ularning o'zaro aloqadorligi;
- hujjatning harakatlanish yo'nalishi va davomiyligi;
- ushbu hujjatdan foydalanish va saqlash joyi;
- ichki va tashqi axborot aloqalari;
- hujjatdagi belgilar soni.

Tekshirish natijalariga ko'ra, avtomatizatsiyalashtirish maqsadida boshqaruv vazifalarining ro'yxati tuziladi va ularni ishlab chiqish tartibi belgilanadi.

### 3.5. Tizim funksiyalarini sinflash

Tekshirish bosqichida tizimning rejalashtirilgan funksiyalarini muhimlik darajasi bo'yicha sinflash kerak. Bunday sinflashni ifodalash mumkin bo'lgan formatlaridan biri, bu MuSCow.

Ushbu qisqartirma quyidagi ma'noni anglatadi:

- ❖ *Must have* – zaruriy funksiyalarni;
- ❖ *Should have* – ma'qul funksiyalarni;
- ❖ *Could have* – kelishi mumkin bo'lgan funksiyalarni;
- ❖ *Won't have* – etishmayotgan funksiyalarni.

Birinchi toifadagi funksiyalar tizimning muvaffaqiyatli ishlashi uchun muhim bo'lgan imkoniyatlarni ta'minlaydi.

Ikkinchi va uchinchi toifadagi funksiyalarni amalga oshirish vaqt va moliyaviy nazar bilan cheklangan: zarur bo'lgan narsa ishlab chiqiladi, shuningdek, ikkinchi va uchinchi toifadagi funksiyalar ustuvorlik soni tartibida ishlab chiqiladi.

Funksiyalarning oxirgi toifasi ayniqsa muhimdir, chunki bu loyihaning chegaralarini va tizimda etishmayotgan funksiyalar to'plamini aniq tasavvur qilish uchun muhimdir.

Tashkiliy faoliyat modellari ikki turda yaratiladi:

- *“bori qanaqa” (AS-IS) modeli* – tashkilotdagi mavjud tijorat jarayonlarini aks ettiradi;
- *“qanaqa bo'lishi kerak” modeli (TO-BE)* – ATni joriy etishni hisobga olgan holda tijorat jarayonlardagi zaruriy o'zgarishlarni aks ettiradi.

Tahlil bosqichida quyidagi masalalarni hal qilish uchun, test guruhlarini ishga jalb qilish kerak:

- foydalanish uchun mo'ljallangan asbobli platformalar, operatsion tizimlar, BBBT va boshqa muhitlarni qiyosiy tavsiflarini olishda;
- yangi tizim va uning axborotning ishonchliligini ta'minlash va sinovlarni o'tkazish bo'yicha ish rejasini ishlab chiqish.

Har qanday loyiha uchun ishlab chiqishning dastlabki bosqichlarida sinovchilarni jalb qilish maqsadga muvofiq. Agar lohiaviy yechim muvaffaqiyatsiz bo'lib chiqsa va bu juda kech, ishlab chiqish bosqichida yoki undan ham yomoni, ishlash jarayonida aniqlansa, loyihalash xatosini tuzatish juda qimmatga tushadi. Sinov guruhlarini ATgi xatolarni qanchalik vaqtli aniqlasa, tizimni kuzatish xarajatlari shunchalik arzon bo'ladi. Tizimni sinovdan o'tkazish va aniqlangan xatolarni tuzatish vaqti nafaqat ishlab chiqish bosqichida, balki loyihalash bosqichida ham ta'minlanishi kerak.

Sinovni avtomatizatsiyalashtirish uchun xatolarni kuzatish tizimlaridan (masalan, *bug tracking*) foydalanish kerak. Bu bizga xatolarning yagona xotirasiga ega bo'lish, ularning takrorlanishini kuzatish, xatolarni tuzatish tezligi va samaradorligi boshqarish imkonini beradi, tizimning eng beqaror tarkibiy qismlarini ko'rish, shuningdek, ishlab chiqish va sinov guruhi o'rasidagi aloqani saqlab turish imkonini beradi. Loyiha qanchalik katta bo'lsa, *bug trackingga* zarurat shunchalik ko'p bo'ladi.

Tekshirish natijalari AT uchun Texnik topshiriq (TT) shartlarni shakllantirishga ob'yektiv asos yaratadi.

TT – bu avtomatizasiyalashtirilgan texnologiyalarni ishlab chiqish uchun zarur bo'lgan maqsadlar, talablar va asosiy kirish berilganlarini belgilaydigan hujjat.

Xomaki loyiha – tizim va uning qismlari uchun dastlabki loyihaviy yechimlarni ishlab chiqishni nazarda tutadi.

Xomaki loyihalash bosqichini bajarish qat'iy shart emas. Agar da asosiy loyihaviy yechimlar ilgari aniqlangan bo'lsa yoki ma'lum bir AT va avtomatizasiyalashtirish ob'yekti uchun juda aniq bo'lsa, unda bu bosqich zarur emas. Dastlabki loyihaning mazmuni tizimning TTda ko'rsatiladi.

Odatda, xomaki loyihalash bosqichida quyidagilar aniqlanadi:

- ATning funksiyalari;
- qism-tizimlarning funksiyalari, maqsadlari va amalga oshirishdan kutilayotgan samaradorlik darajasi;
- masalalar majmuasi va alohidagi masalalarning tarkibi;
- axborot bazasi va uning kengaytirilgan tuzilmasi;
- BBTning funksiyalari;
- kompyuter tizimlari va boshqa texnik vositalarning tarkibi;
- asosiy dasturiy ta'minotning funksiyalari va parametrlari.

Bajarilgan ish natijalariga ko'ra, hujjatlar to'liq tavsiflash uchun zarur bo'lgan darajada tuziladi, kelishiladi va tasdiqlanadi.

#### 8.6. Nazorat ugun savollar

1. Axborot tizimlarini loyihalashdagi asosiy sohalar va ularning vazifasi.
2. ATni kanonik loyihalash asosida ATning hayot siklidagi qaysi model yotadi?
3. ATni kanonik loyihalashdagi davrlarni sanab bering.
4. ATni kanonik loyihalashdagi davrlarni izohlab bering.
5. Tashkilot tuzilmasini tekshirishdagi maqsad nimadan iborat?

6. Tashilot strategiyasi nimaga asoslanadi va uning mohiyati nimadan iborat?
7. Tashkilot faoliyatining tahlilidagi asosiy vazifalar nimalardan iborat?
8. Tizim funksiyalarini sinflashtirish maqsad nima?
9. AS-IS modelini ta'riflang.
10. TO-BE modelini ta'riflang.
11. Xomaki loyihalash bosqichida qanday vazifalar aniqlanadi?
12. Texnik topshiriq qachon aniqlanadi?

## IX- BOB. TEXNIK TOPSHIRIQNING TUZILISHI

### 9.1. Umumiy qoidalar

AT uchun texnik topshiriq (TT) avtomatizatsiyalashtirilgan tizimni yaratish talablari va tartibini belgilaydigan asosiy hujjat bo'lib, unga muvofiq AT ishlab chiqiladi va ishga tushirilganda qabul qilinadi.

AT uchun TT mustaqil ravishda yoki boshqa tizimning bir qismi sifatida ishlash uchun mo'ljallangan butun tizim uchun ishlab chiqiladi.

Qo'shimcha ravishda, AT qismlari uchun ham, TT ishlab chiqilishi mumkin:

- davlat standarti talablariga muvofiq AT qism-tizimlariga, AT masalalari majmualariga va boshqalarga;
- davlat standartlarga muvofiq texnik ta'minot tarkiblari va dasturiy-texnik majmualarga;
- standartlariga muvofiq DTga.

Davlat standartida belgilangan hajmlarga talablar yangi yaratilayotgan avtomatizatsiyalashtiriladigan ob'yektni loyihalash vazifasiga kiritilishi mumkin. Odatda bunday holda, AT uchun TT ishlab chiqilmaydi.

AT uchun TTga kiritilgan talablar fan va texnologiyaning zamonaviy rivojlanish darajasiga mos kelishi va eng yaxshi zamonaviy mahalliy va xorijiy o'xshash tizimlarga qo'yiladigan talablardan kam bo'lmash kerak. TTda o'rnatilgan talablar tizim ishlab chiqaruvchisini eng samarali texnik, texnik-iqtisodiy va boshqa yechimlarni topish va amalga oshirishda cheklamasligi kerak.

TT ishlab chiqishdagi boshlang'ich berilganlar, standart tomonidan belgilangan "AT bo'yicha tadqiqot va yaratish asoslari" (shartli nom) hujjatlari asosida ishlab chiqiladi.

TTda ob'yektning o'ziga xos xususiyatlari bilan belgilanadigan talablar mavjud bo'lishi mumkin.

AT uchun TTga o'zgartirishlar buyurtmachi va ishlab chiquvchi tomonidan imzolangan qo'shimcha yoki bayonnoma bilan amalga oshiriladi. Qo'shimcha yoki ko'rsatilgan bayonnoma AT uchun TTning ajralmas qismi bo'lib hisoblanadi. TTning sarlavha sahifasida amal qilishning sanasi ko'rsatilishi shart.

### 9.2. Texnik topshiriq tarkibi va mazmuni

ATdagi TT quyidagi bo'limlarni o'z ichiga oladi, ularni kichik bo'limlarga bo'lish mumkin:

- umumiy ma'lumotlar;
- tizimni yaratish maqsadi va tayinlanishi;
- avtomatizatsiyalashtirish ob'yektlarining xususiyatlari;
- tizimga qo'yiladigan talablar;
- tizimni yaratish bo'yicha ishlarning tarkibi va mazmuni;
- tizimni boshqarish va qabul qilish tartibi;
- tizimni ishga tushirish uchun avtomatizatsiyalashtirish ob'yektini tayyorlash bo'yicha ishlarning tarkibi va mazmuniga qo'yiladigan talablar;
- hujjatlashtirish talablari;
- ishlab chiqarish.

TTga ilovalar kiritilishi mumkin.

Tayinlanishi, maqsadi va o'ziga xos xususiyatlariga qarab avtomatizatsiyalashtirish ob'yekti va tizimning – TT bo'limlarini ilovalar shaklida tuzing, qo'shimcha dasturlarni kiriting, TT bo'limlarini chiqarib tashlash yoki birlashtirish shartlariga ruxsat beriladi.

Umuman olganda, TT tizimning qismlari tarkibini takrorlaydigan bo'limlarni o'z ichiga olmasligi kerak.

"Umumiy ma'lumotlar" bo'limida:

- tizimning to'liq nomi va uning belgisi;
- mavzu shifri yoki shartnoma raqami;
- tizimni ishlab chiquvchi va buyurtmachi korxonaning nomi

va uning tafsilotlari;

- tizim yaratilgan hujjatlar ro'yxati, ushbu hujjatlar kim tomonidan va qachon tasdiqlangan;
- tizimni yaratish bo'yicha ishlarni boshlash va tugatishning rejalashtirilgan muddatlari;
- ishlarni moliyalashtirish manbalari va tartibi to'g'risidagi ma'lumotlar;
- tizimni (uning qismlarini) yaratish, tizimning shaxsiy vositalarini (texnik, dasturiy, axborot) va dasturiy-texnik (dasturiy-uslubiy) majmualarini ishlab chiqarish va sozlash bo'yicha ish natijalarini buyurtmachiga taqdim etish tartibi.

"Tizimni yaratish (rivojlantirish) maqsadi va maqsadlari" bo'limi quyidagi bo'limlardan iborat bo'lishi tavsiya etiladi:

- tizimning tayinlanishi;
- tizimni yaratish maqsadlari.

"Tizimning tayinlanishi" bo'limida avtomatizatsiyalashtirish faoliyatining turi (boshqarish, loyihalash va boshqalar) va undan foydalanish rejalashtirilgan avtomatizatsiyalashtirish ob'yektlari (ob'yektlari) ro'yxati ko'rsatiladi.

Avtomatizatsiyalashtirilgan boshqarish tizimlari uchun qo'shimcha ravishda avtomatizatsiyalashtirilgan boshqaruv organlari (punktlari) va boshqariladigan ob'yektlar ro'yxati ko'rsatiladi.

"Tizimni yaratish maqsadlari" bo'limida esa avtomatizatsiyalashtirish ob'yektining texnik, texnologik, ishlab chiqarish-iqtisodiy yoki boshqa ko'rsatkichlarining nomlari va talab qilinadigan qiymatlari keltirilgan bo'lib, ularga avtomatizatsiyalashtirilgan tizimni yaratish natijasida erishishi kerak va tizimni yaratish maqsadlariga erishishni baholash mezonlari ko'rsatilishi kerak.

"Avtomatizatsiyalashtirish ob'yektining xususiyatlari" bo'limida quyidagilar keltiriladi:

- avtomatizatsiyalashtirish ob'yekti haqida qisqacha ma'lumot yoki bunday ma'lumotlarni o'z ichiga olgan hujjatlarga havolalar;

- avtomatizatsiyalashtirish ob'yektining ishlash shartlari va atrof-muhit xususiyatlari to'g'risida ma'lumot.

"Tizim talablari" bo'limi quyidagi qism-bo'limlardan iborat bo'ladi:

- ❖ tizimga qo'yiladigan umumiy talablar;
- ❖ tizim tomonidan bajariladigan funksiyalarga (vazifalarga) qo'yiladigan talablar;
- ❖ ta'minot turlariga qo'yiladigan talablar.

ATda TTning ushbu bo'limiga kiritilgan tizimga qo'yiladigan talablarning tarkibi ma'lum bir tizimning turi, maqsadi, o'ziga xos xususiyatlari va ishlash sharoitlariga qarab belgilanadi. Har bir qism-bo'limda tegishli turdagi tizimlarga qo'yiladigan talablarni belgilaydigan amaldagi ilmiy-texnikaviy berilganlarga havolalar keltiriladi.

### 9.3. Tizimga qo'yiladigan umumiy talablar

"Tizimga qo'yiladigan umumiy talablar" qism-bo'limida quyidagi talablar keltiriladi:

- tizimning tuzilishi va ishlashiga;
- tizim xodimlarining soni va malakasiga va ularning ish tartibotiga;
- tayinlash ko'rsatkichlariga;
- ishonchlilikka;
- xavfsizlikka;
- ergonomikaga va texnik estetikaga;
- harakatlanuvchi AT uchun tashish qobiliyatiga;
- tizim tarkiblarini ishlatish, texnik xizmat ko'rsatish, ta'mirlash va saqlashga;
- ma'lumotni ruxsatsiz kirishdan himoya qilishga;
- baxtsiz hodisalar paytida ma'lumotlarning xavfsizligini saqlashga;
- tashqi ta'sirlardan himoya qilishga;
- patent tozaligiga;

- standartlashtirish va yagonalashtirishga;
- qo'shimchalarga.

*Tizimning tuzilishi va ishlashiga qo'yiladigan talablarda*, quyidagilar keltirilishi so'raladi:

- ✓ qism-tizimlar ro'yxati, ularning maqsadi va asosiy xususiyatlari, shajaraviy darajalar soniga qo'yiladigan talablar va tizimni markazlashtirish darajasi;
- ✓ tizim tarkiblari o'rasida axborot almashinuvi uchun aloqa usullari va vositalariga qo'yiladigan talablar;
- ✓ yaratilgan tizimning qo'shni tizimlar bilan o'zaro bog'liqligi xususiyatlariga qo'yiladigan talablar, uning muvofiqligi talablari, shu jumladan ma'lumot almashish usullari (avtomatik ravishda, hujjatlarni yuborish, tarmoq yoki telefon orqali va boshqalar);
- ✓ tizimning ishlash tartibotlariga qo'yiladigan talablar;
- ✓ tizim tashxisi talablari;
- ✓ tizimni rivojlantirish, takomillashtirish istiqbollari.

*ATdagi xodimlar soni va malakasiga qo'yiladigan talablar:*

- xodimlar (foydalanuvchilar) soniga qo'yiladigan talablar;
- xodimlarning malakasiga, ularni tayyorlash va bilim va ko'nikmalarni nazorat qilish tartibiga qo'yiladigan talablar;
- xodimlardan talab qilinadigan ish tartibi.

*AT tayinlanish ko'rsatkichlariga qo'yiladigan talablar* tizimning maqsadiga muvofiqligi darajasini tavsiflovchi parametrlarning qiymatlarini beradi.

AT uchun quyidagilarni ko'rsatish mumkin:

- ❖ tizimning boshqaruv jarayonlari va usullarining o'zgarishiga boshqaruv ob'yekti parametrlarining og'ishlariga moslashish darajasini;
- ❖ tizimni modernizatsiya qilish va rivojlantirishning ruxsat etilgan chegaralarini;
- ❖ tizimning maqsadi saqlangan holda, ehtimollik-vaqt xususiyatlarini.

*Ishonchlilik talablari* quyidagilarni o'z ichiga olisi kerak:

- tizim yoki uning qism-tizimlari uchun ishonchlilik ko'rsatkichlarining tarkibi va miqdoriy qiymatlari;
- ishonchlilik talablari va tegishli ko'rsatkichlarning qiymatlari tartibga solinishi kerak bo'lgan favqulodda vaziyatlar ro'yxati;
- texnik vositalar va dasturiy ta'minotning ishonchliligi talablari;
- amaldagi me'yoriy-texnik hujjatlarga muvofiq tizimni yaratishning turli bosqichlarida ishonchlilik ko'rsatkichlarini baholash va nazorat qilish usullariga qo'yiladigan talablar.

*Xavfsizlik talablariga* tizimning texnik vositalarini (elektr tokining ta'siridan, elektromagnit maydonlardan, akustik shovqinlardan va boshqalardan himoya qilish), yorug'lik, tebranish va shovqin yuklarining ruxsat etilgan darajalari bo'yicha o'rnatish, ishga tushirish, ishlatish, texnik xizmat ko'rsatish va ta'mirlash paytida xavfsizlikni ta'minlash talablari kiradi.

*Ergonomika va texnik estetika* talablariga inson va mashina o'rasidagi o'zaro munosabatlarning zarur sifatini va xodimlarning ish sharoitlarining qulayligini belgilaydigan AT ko'rsatkichlari kiradi.

Harakatlanuvchi AT uchun transportga qo'yiladigan talablar tizimning texnik vositalarining transportga yaroqliligini ta'minlaydigan talablarini, shuningdek transport vositalariga qo'yiladigan talablarni o'z ichiga oladi.

*Tizimdan foydalanish, texnik xizmat ko'rsatish, ta'mirlash va saqlash* talablari quyidagilarni o'z ichiga oladi:

- ❖ belgilangan texnik ko'rsatkichlarga ega tizimning texnik vositalaridan foydalanishni ta'minlashi kerak bo'lgan foydalanish shartlari va qoidalari (tartiboti), shu jumladan ATga xizmat ko'rsatish turlari va chastotasi yoki texnik xizmat ko'rsatmasdan ishlashga ruxsat berish;
- ❖ tizim xodimlari va transport vositalarini joylashtirish uchun ruxsat etilgan maydonlarga, energiya ta'minoti tarmoqlari parametrlariga va boshqalarga qo'yiladigan talablar;

- ❖ xizmat ko'rsatuvchi xodimlarning soni, malakasi va ularning ish tartibotlariga qo'yiladigan talablar;
- ❖ ehtiyot qismlar va asboblarning to'plamining tarkibi, joylashishi va saqlash sharoitlariga qo'yiladigan talablar;
- ❖ xizmat ko'rsatish qoidalariga qo'yiladigan talablar.

Axborotni ruxsatsiz kirishdan himoya qilish talablari mijozning tarmog'ida (idorasida) faoliyat yuritadigan ilmiy-texnik berilganlarda belgilangan talablarni o'z ichiga oladi.

*Axborotni saqlash bo'yicha talablarda* voqealar ro'yxati keltiriladi – baxtsiz hodisalar, texnik vositalarning ishlamay qolishi (shu jumladan elektr manbasining yo'qolishi) va boshqalar, bunda tizimdagi ma'lumotlarning xavfsizligi ta'minlanishi kerak.

*Tashqi ta'sirlardan himoya qilish talablariga* quyidagilar kiradi:

- AT vositalarini radioelektron himoya qilish talablari;
- tashqi ta'sirlarga (dastur muhitiga) qarshilik, barqarorlik va chidamlilik talablari.

*Patent tozaligi to'g'risidagi talablarda* tizim va uning qismlarining patent tozaligi ta'minlanishi kerak bo'lgan mamlakatlar ro'yxati ko'rsatiladi.

*Standartlashtirish va yagonalashtirish talablari* quyidagilarni o'z ichiga oladi – tizim funksiyalarini amalga oshirishning standartini, yagonalashtirish usullarini, etkazib beriladigan dasturiy vositalarni, standart matematik usullar va modellarni, namunaviy loyihalashtirish yechimlarini, belgilangan boshqaruv hujjatlarining yagona shakllarini, texnik-iqtisodiy ma'lumotlarning tasniflagichlarini va boshqa toifadagi sinflovchilarni, bunda qo'llash sohasiga mos ravishda foydalanishning zarur darajasini belgilaydigan ko'rsatkichlarni bilan bir qatorda standart avtomatlashtirilgan ish o'rinlari, tarkiblari va majmualaridan foydalanish talablari hisobga olinadi.

*Qo'shimcha talablar* quyidagilarni o'z ichiga oladi:

- tizimni xodimlarni o'qitish uchun qurilmalar va hujjatlar bilan jihozlashga qo'yiladigan talablar;

- tizim elementlarini tekshirish uchun xizmat ko'rsatish uskunalarini, stendlarga qo'yiladigan talablar;
- maxsus ish sharoitlari bilan bog'liq bo'lgan, tizimga qo'yiladigan talablar;
- tizimning yaratuvchisi yoki buyurtmachining xohishiga ko'ra maxsus talablar.

Tizim tomonidan bajariladigan "Funksiyalar (masalalar)ga talab" qism-bo'limida quyidagilar keltiriladi:

- ✓ har bir qism-tizim uchun avtomatizasiyalashtirilishi kerak bo'lgan funksiyalar, vazifalar yoki ularning majmualari (shu jumladan tizim qismlarining o'zaro ta'sirini ta'minlaydigan) ro'yxati; tizimni yaratishdagi ikki yoki undan ortiq navbat bo'lgan holatlarda, birinchi va undan keyingi navbatlarda tizimga kiritilgan funksional qism-tizimlar, shaxsiy funksiyalar yoki vazifalar ro'yxati;
- ✓ har bir funksiyani (vazifa yoki vazifalar to'plamini) amalga oshirish uchun vaqt reglamenti;
- ✓ har bir funksiyani (vazifa yoki vazifalar to'plamini) amalga oshirish sifatiga, chiqish ma'lumotlarini taqdim etish shakliga, kerakli aniqlik va bajarilish vaqtining xususiyatlariga, funksiyalar guruhini bir vaqtda bajarishga qo'yiladigan talablarga, natijalarni berishning ishonchliligiga qo'yiladigan talablar;
- ✓ ishonchlilik talablari belgilangan har bir funktsiya uchun rad etish ro'yxati va mezonlari.

#### 9.4. TTda ATni loyihalashdagi ta'minot turlariga talablar

*"Ta'minot turlariga qo'yiladigan talablar"* bo'limida tizim turiga qarab matematik, axborot, lingvistik, dasturiy, texnik, metrologik, tashkiliy, uslubiy va tizimni ta'minlashning boshqa turlariga qo'yiladigan talablar keltiriladi.

*Tizimni matematik qo'llab-quvvatlash* uchun tizimda matematik usullar va modellar, ishlab chiqilishi kerak bo'lgan standart

algoritmlar va algoritmlardan foydalanish tarkibi, ko'lam (cheklov-lari) va usullariga qo'yiladigan talablar keltiriladi.

*Tizimni axborot bilan ta'minlash uchun quyidagi talablar keltiriladi:*

- tizimdagi berilganlarning tarkibi, tuzilishi va tashkil etish usullariga;
- tizim tarkiblari o'rasida axborot almashinuviga;
- qo'shni tizimlar bilan ma'lumotlarning muvofiqligiga;
- mazkur korxonada amalda bo'lgan umumittifoq va ro'yxatdan o'tkazilgan respublika, tarmoq sinflovchilari, yagona hujjatlar va sinflovchilardan foydalanishga;
- BBBTdan foydalanishga;
- tizimda berilganlarni yig'ish, qayta ishlash, uzatish va berilganlarni taqdim etish jarayonining tuzilishiga;
- tizimning elektr ta'minotidagi baxtsiz hodisalar va nosozliklar paytida berilganlarni halokatdan himoya qilishga;
- berilganlarni boshqarish, saqlash, yangilash va tiklashga;
- AT texnik vositalar bo'yicha ishlab chiqarilgan hujjatlarga huquqiy kuch berish tartibiga.

*Tizimni lingvistik qo'llab-quvvatlash uchun tizimda yuqori darajadagi dasturlash tillari, foydalanuvchilarning o'zaro ta'sir tillari va tizimning texnik vositalaridan foydalanish talablari, shuningdek, berilganlarni kodlash va aks kodlash talablari, berilganlarni kiritish-chiqarish tillari, berilganlarga ishlov berish vositalari kiritish-chiqarish sohani (ob'yekti) tavsiflash vositalari, tizim bilan foydalanuvchning muloqatini tashkil qilish usullari.*

*Tizim dasturiy ta'minoti uchun sotib olinadigan dasturiy ta'minot ro'yxati, shuningdek quyidagi talablar keltiriladi:*

- dasturiy vositalarning ishlatiladigan hisoblash texnikasi vositalari va amaliy muhitning mustaqilligiga;
- dasturiy ta'minot sifati, shuningdek uni ta'minlash va boshqarish usullariga;
- yangi ishlab chiqilgan dasturiy ta'minotni algoritmlar va dasturlar fondi bilan muvofiqlashtirish zaruratiga.

*Tizimni texnik qo'llab-quvvatlash uchun talablar:*

- texnik vositalar turlariga, shu jumladan tizimda foydalanish uchun ruxsat etilgan texnik vositalar, dasturiy-texnik majmualar va boshqa butlovchi qismlar majmualarining turlariga;
- tizimni texnik qo'llab-quvvatlash vositalarining funksional, tuzilmaviy va foydalanishdagi xususiyatlariga.

*Metrologik ta'minotga qo'yiladigan talablar:*

- o'lchash kanallarining dastlabki ro'yxatiga;
- parametrlarni o'lchash aniqligi va/yoki o'lchash kanallarining metrologik xususiyatlariga;
- tizimning texnik vositalarining metrologik muvofiqligiga;
- aniqlik xususiyatlarini baholash kerak bo'lgan tizimning boshqaruv va hisoblash kanallariga;
- tizimni sozlash va sinovdan o'tkazishda foydalaniladigan tizimning o'lchash kanallari, o'rnatilgan boshqaruv vositalari, o'lchash kanallari va o'lchash vositalarining metrologik yaroqliligini tashkil etuvchi texnik va dasturiy vositalarni metrologik ta'minlashga;
- metrologik attestatsiya turi (davlat yoki idoraviy), uni amalga oshirish tartibi va attestatsiyani o'tkazuvchi tashkilotlarga.

*Tashkiliy qo'llab-quvvatlash uchun talablar:*

- ❖ tizim faoliyatida ishtirok etadigan yoki foydalanishni ta'minlaydigan bo'linmalarning tuzilishi va funksiyalariga;
- ❖ tizimning ishlashini tashkil etish hamda AT xodimlari va avtomatizasiyalashtirish ob'yekti xodimlari orasida o'zaro harakat qilish tartibiga;
- ❖ tizim xodimlarining noto'g'ri harakatlaridan himoya qilishga.

ATni uslubiy qo'llab-quvvatlash uchun me'yoriy-texnik hujjatlari tarkibiga qo'yiladigan talablar (uning ishlashi uchun ishlatiladigan standartlar, meyorlar, usullar va boshqalar ro'yxati) qo'yiladi.

### 9.5. TTda ATni yaratish bo'yicha ishlarning tarkibi va mazmuni

"Tizimni yaratish (rivojlantirish) bo'yicha ishlarning tarkibi va mazmuni" bo'limida tizimni yaratish bo'yicha ishlarning bosqichlari va bosqichlari ro'yxati, ularni amalga oshirish muddatlari, ishlarni bajaruvchi tashkilotlar ro'yxati, ushbu tashkilotlarning tizimni yaratishda ishtirok etishga roziligini tasdiqlovchi hujjatlarga havo-lalar yoki mas'ul shaxsni belgilaydigan yozuv bo'lishi kerak.

Ushbu bo'limda quyidagilar ham keltiriladi:

- ❖ ishlarning tegishli bosqichlari va bosqichlari oxirida taqdim etiladigan hujjatlar ro'yxati;
- ❖ texnik hujjatlarni ekspertizadan o'tkazish turi va tartibi (bosqich, tekshirilayotgan hujjatlar hajmi, ekspert tashki-loti);
- ❖ ishlab chiqilayotgan tizimning zarur darajadagi ishonchli-gini ta'minlashga qaratilgan ishlar dasturi;
- ❖ tizimni yaratishning barcha bosqichlarida metrologik qo'l-lab-quvvatlash bo'yicha ishlarning ro'yxati, ularning bajari-lish muddatlari va ijrochilarning tashkilotlari ko'rsatiladi.

"Tizimni nazorat qilish va qabul qilish tartibi" bo'limida quyidagilar ko'rsatiladi:

- tizim va uning tarkibiy qismlarini sinash turlari, tarkibi, hajmi va usullari (ishlab chiqilayotgan tizimga taalluqli amaldagi standartlarga muvofiq sinov turlari);
- ishlarni bosqichma-bosqich qabul qilishga qo'yiladigan umumiy talablar (ishtirok etuvchi korxonalar va tashkilotlar ro'yxati, o'tkaziladigan joy va muddatlar), qabul hujjatlarini tasdiqlash va tasdiqlash tartibi;
- qabul komissiyasining maqomi (davlat, idoralari, idoraviy).

"Tizimni ishga tushirish uchun avtomatizasiyalashtirish ob'yektini tayyorlash bo'yicha ishlarning tarkibi va mazmuniga qo'yiladigan talablar" bo'limida avtomatizasiyalashtirish ob'yektini

ishga tushirish uchun tayyorlashda bajarilishi kerak bo'lgan asosiy tadbirlar va ulardagi ijrochilarning ro'yxati keltirigadi.

Asosiy tadbirlar ro'yxatiga quyidagilar kiritiladi:

- tizimga kiradigan ma'lumotlarni, axborot va lingvistik qo'l-lab-quvvatlash talablariga muvofiq, kompyuter yordamida qayta ishlashga yaroqli shakliga keltirish;
- avtomatizasiyalashtirish ob'yektida amalga oshirilishi kerak bo'lgan o'zgarishlar;
- avtomatizasiyalashtirish ob'yektining ishlashi uchun sharoit yaratish, unda yaratilgan tizimning TT tarkibidagi talablarga muvofiqligi kafolatlash;
- tizim ishlashi uchun zarur bo'lgan bo'linmalar va xizmatlarni yaratish;
- xodimlarni yollash, o'qitish shartlari va tartibi.

Masalan, boshqaruvchi AT uchun:

- qo'llaniladigan boshqaruv usullarining o'zgarishi;
- tizimning TT tarkibidagi talablarga muvofiqligi kafolatlangan

AT tarkibiy qismlarining ishlashi uchun sharoit yaratish.

"Hujjatlashtirish talablari" bo'limida quyidagilar keltiriladi:

- tizimni ishlab chiquvchi va buyurtmachi tomonidan kelishil-gan buyurtmachi sanoatining ilmiy-taxnikaviy berilgan-lariga mos keladigan ishlab chiqilishi kerak bo'lgan huj-jatlar to'plamlari va turlari ro'yxati – mashina vositalarida chiqarilgan hujjatlar ro'yxati; hujjatlarni mikrofilmlashga qo'yiladigan talablar;
- tarmoqlararo dastur tarkibiy qismlarini hujjatlashtirishga qo'yiladigan talablar;
- tizim elementlarini hujjatlashtirishga qo'yiladigan talablarni belgilaydigan davlat standartlari bo'lmagan taqdirda, qo'-shimcha ravishda bunday hujjatlarning tarkibi va mazmu-niga qo'yiladigan talablarni o'z ichiga oladi.

"Ishlab chiqarish manbalari" bo'limida hujjatlar va axborot materiallari (texnik-iqtisodiy asoslash, tugallangan ilmiy-tadqiqot ishlari to'g'risidagi hisobotlar, mahalliy, xorijiy analog tizimlar

uchun axborot materiallari va boshqalar) ko'rsatilishi kerak, ular asosida TT ishlab chiqilishi va tizimni yaratishda ishlatilishi kerak.

Tasdiqlangan usullar mavjud bo'lganda, ATni yaratishdagi TT tarkibiga quyidagilar kiradi:

- tizimning kutilayotgan samaradorligini hisoblash;
- tizimning ilmiy va texnik darajasini baholash.

Ilovalar tizimni ishlab chiqaruvchi va buyurtmachi o'rasidagi kelishuvga binoan ATning TT tarkibiga kiritiladi.

### 9.6. Nazorat uchun savollar

1. TT kim tomonidan va nima maqsadda ajratiladi?
2. TTning tarkibi va mazmunida qanday vazifalar yotadi?
3. TTda qanday bo'limlar bo'lishi kerak?
4. TTda tizimga qo'yiladigan umumiy talablar nimalardan iborat?
5. TTda ATni loyihalashdagi qaysi ta'minot turlariga talablar qo'yiladi?
6. ATdagi ta'minot turlarini aytib bering.
7. Tizimni ishga tushirish uchun qanday asosiy tadbirlar amalga oshirilishi kerak?
8. Tizim ustidagi ilmiy-tadqiqot ishlari qaysi davrda amalga oshiriladi?
9. Tizimni texnik-iqtisodiy asoslash qachon amalga oshiriladi?

## X- BOB. AXBOROT TA'MINOTINI MODELLASHTIRISH

### 10.1. Berilganlarni modellashtirish

Axborot ta'minotining asosiy qismlaridan biri berilganlar bazasi hisoblanadi. *Berilganlar bazasi* – bu ma'lum bir tarzda tashkil etilgan va boshqaruv jarayonlari va hal qilinadigan vazifalarning axborot ehtiyojlari qondiriladigan fayllar shaklida hisoblash tizimi xotirasida saqlanadigan axborotlar to'plami. Berilganlar bazasining tuzilishi loyihalash jarayonida aniqlanadi. Berilganlar bazasini loyihalash jarayoni juda murakkab bo'lishi mumkin. Berilganlar bazasini ishlab chiqish berilganlarni modellashtirish yordamida amalga oshiriladi. Berilganlarni modellashtirishning maqsadi ATni ishlab chiqaruvchisini bitta model yoki bir nechta mahalliy modellar ko'rinishidagi konseptual berilganlar bazasi chizmasi bilan ta'minlashdir, ularni har qanday berilganlar bazasi tizimiga nisbatan osonlik bilan ko'rsatish mumkin. Berilganlar bazasining eng muhim jihatlaridan biri, uning doimiy ravishda o'zgarishi va ko'payishidan iborat. Birorta tashkilotning axborotga bo'lgan talablarini qondirish uchun, alohida foydalanuvchilarning ehtiyojlarini qondiruvchi oddiy fayl tizimlaridan iborat ilovalarni yaratishga nisbatan, umuman boshqacha yondashuvni talab qiladi. BB asosidagi tizimlarni yaratish uchun birinchi navbatda berilganlar, ana undan keyin ilovalar xususida o'ylash mumkin. Bunday yondashuvlarning almashishini loyihalash tamoyillarining almashishi deb baholanishi mumkin. Tizim, foydalanuvchilarning so'rovlariga to'liq javob berishi uchun, berilganlar bazasini loyihalash jarayoniga juda katta e'tibor bilan munosabatda bo'lish kerak. Noto'g'ri loyihalangan BB xatoliklar tug'diradi, natijada noto'g'ri qarorlar qabul qilinadi, bu

esa tashkilotni qiyin vaziyatga olib kelishi mumkin. Boshqa tomondan, yaxshi loyihalangan BB bilan ishlovchi tizimga aniq ma'lumotlar chiqarib berilsa, to'g'ri va samarali qarorlarni qabul qilishga zamin bo'ladi.

Afsuski, hozirgi kundagi berilganlar bazalarini loyihalash uslublari keng tarqalmagan. Ko'pchilik tashkilotlar yoki alohida dastur mahsulotlarini yaratuvchilar, juda kam holatlarda birorta uslubiyatga tayanadilar. Axborot tizimlarini yaratishdagi ana shu kamchilik, ko'pgina muvaffaqiyatsizliklarga olib kelmoqda.

Berilganlar bazasini loyihalashdagi tuzilmali yondashuvlarning yo'qligi, yaratishga ketadigan vaqt va vositalarga yetarlicha e'tibor bermaslikka, yaratilgan BB samarasiz yoki ilovalarning talablariga javob bermaydigan holatlarga olib kelmoqda. Dastur mahsulotining tavsifi va uni amalga oshirish bo'yicha ko'rsatmalar yozilgan hujjatlar yetarlicha yaxshi emasligi, berilganlar bazalarini kuzatishdagi katta qiyinchiliklarni tug'dirmoqda.

Berilganlarni modellashtirishning eng keng tarqalgan vositasi – bu mohiyat-aloqa diagrammasi (Entity-Relationship Diagram – ERD). ERD yordamida DFD diagrammasi berilganlar drayverlari batafsil bayon qilinadi, shuningdek, tijorat tizimining axborot jihatlarini, shu jumladan domen (mohiyatlar) uchun muhim bo'lgan mohiyatlarni, ushbu mohiyatlarning xususiyatlarini (atributlarini) va ularning boshqa mohiyatlar bilan aloqalarini aniqlab hujjatlashtiriladi.

## 10.2. «Mohiyat-aloqa» modeli

«Mohiyat-aloqa» modelining asosiy elementlari – *mohiyatlar, atributlar, aniqlovchilar va aloqalardan* iborat.

*Mohiyat (entity)* – umumiy atributlar yoki xususiyatlarga ega bo'lgan haqiqiy yoki mavhum ob'yektlarning (odamlar, hodisalar, holatlar, g'oyalar, ob'yektlar va boshqalar) ko'plab nusxalari tizim-

ning har qanday mohiyati noyob tarzda aniqlanishi kerak bo'lgan faqat bitta mohiyat bilan ifodalanishi mumkin. Bunday holda, mohiyat nomi ob'jektning o'ziga xos nusxasini emas, balki turini yoki sinfini aks ettirishi kerak. Mohiyatlarga misolar – XIZMATCHI Baxtiyor Salimov, MIJOZ 77345, BUYURTMA 0049 va MAX-SULOT 590310. Bir xil turdagi mohiyatlarning guruhlari mohiyatlarning sinflarini tashkil etadi. XIZMATCHI turidagi mohiyatlarning sinfi hamma XIZMATCHI turidagi mohiyatlardan iborat. Mohiyatlarning sinflarini bosh harflar bilan belgilaymiz.

Har bir mohiyat noyob aniqlovchiga ega bo'lishi kerak. Mohiyatning har bir nusxasi noyob tarzda aniqlanishi va ushbu turdagi boshqa barcha misollardan farq qilishi kerak. Har bir mohiyat quyidagi xususiyatlarga ega bo'lishi kerak:

- ❖ noyob ismga ega bo'lish; bir xil nom har doim bir xil talqin qo'llanilishi kerak; bir xil talqin, agar u taxallus bo'lmasa, turli xil nomlarga nisbatan qo'llanilishi mumkin emas;
- ❖ mohiyatga tegishli bo'lgan yoki bog'lanish orqali meros qilib olingan bitta yoki bir nechta atributlarga ega bo'lish;
- ❖ mohiyatning har bir nusxasini noyob tarzda aniqlaydigan bitta yoki bir nechta atributlarga ega bo'lish;
- ❖ har bir mohiyat modelning boshqa har qanday miqdordagi mohiyatlari bilan aloqalarga ega bo'lishi mumkin.

Mohiyatlar sinfi va mohiyat nusxasi tushunchalari orasidagi farqni aniqlashtirib o'taylik. *Mohiyatlar sinfi* deganda, mohiyatlarining majmuasi tushuniladi va u o'zining turi bilan tavsiflanadi. *Mohiyat nusxasi* esa, aniq bitta mohiyatni bildiradi, masalan, BUYURTMA 0049 va shu mohiyatning qiymati bilan ifodalanadi. Demak, mohiyatlar sinfi mohiyat nusxalarining to'plamidan iborat deyishimiz mumkin.

*Xususiyat (atribut, attribute)* – ko'rib chiqilayotgan soha uchun muhim bo'lgan va mohiyat holatini saralash, aniqlash, tasniflash, miqdoriy tavsiflash yoki ifodalash uchun mo'ljallangan mohiyatning har qanday xususiyati. Boshqacha aytsak, mohiyatlar

atributlarga ega bo'lib, ularni *xususiyat* (properties)lar, deb ham atashadi. Xususiyatlar mohiyatlarni tavsiflaydi.

Misollar sifatida Hodimning\_ismi, tug'ilgan\_sanasi va ish\_staji atributlarini keltirish mumkin. «Mohiyat-aloqa» modelidagi berilgan sinfning hamma nusxalari bir xil atributga ega, deb faraz qilinadi. «Mohiyat-aloqa» modelining boshlang'ich ta'rif *kompozit* (composite) va *ko'p qiymatli* (multi-valued) atributlarni o'z ichiga oladi. Kompozit atributga misol sifatida Manzil atributini olish mumkin, u Indeks, Shahar, Mavze, Ko'cha va Uy atributlar guruhidan iborat. Ko'p qiymatli atributga misol Ustoz\_familiyasi, bo'lishi mumkin, chunki har bitta talabning bir qancha ustoz bo'ladi. *Atribut bir paytda ham kompozit, ham ko'p qiymatli bo'lishi* mumkin. Masalan, Hudud\_kodi, telefon\_raqami atributlar guruhidan tuzilgan Telefon kompozit atributi, ko'p qiymatli bo'lishi mumkin, bu esa bitta odamning bir nechta telefon raqamlarini saqlash imkoniyatini yaratadi. Ko'pgina «mohiyat-aloqa» modelarini amalga oshiruvchi vositalar bir qiymatli kompozit atributlarni inkor qiladi, ko'p qiymatli atributlarni esa mohiyatlarda akslanishini talab qiladi.

Atribut ko'plab haqiqiy yoki mavhum ob'yektlar (odamlar, joylar, hodisalar, holatlar, g'oyalar, narsalar va boshqalar) bilan bog'liq bo'lgan xususiyatlar yoki xususiyatlar turini anglatadi. Atribut nusxasi – bu to'plamning alohida elementining o'ziga xos xususiyati. Atribut nusxasi, tavsifning turi va uning qiymati deb nomlangan atribut qiymati bilan belgilanadi. "Mohiyat-aloqa" diagrammasida atributlar aniq mohiyatlar bilan bog'lanadi. Shunday qilib, mohiyat nusxasi bog'langan atribut uchun yagona aniq qiymatga ega bo'lishi kerak.

Mohiyatlarning nusxalari *aniqlovchilarga (identifiers)* – ega bo'lib, nusxalar nomlanadi, ya'ni aniqlanadi. Masalan, XIZMAT-CHI sinfi mohiyati nusxalari "Solliq\_to'lovchini aniqlovchi raqami" yoki "Xizmatchining\_familiyasi\_ismi\_sharifi" atributi bilan aniqlanadi. "Ishga\_kirgan\_sanasi" yoki "Maosh" atributlari XIZ-

MATCHI mohiyat sinfi nusxalarining aniqlovchilari bo'la olmaydi, chunki ular xodimni bir qiymatli aniqlay olmaydi.

Mohiyat nusxasini aniqlovchisi bitta yoki bir nechta atributdan tashkil topishi mumkin. Aniqlovchi *noyob* (unique) yoki *noyob bo'lmasligi* (nonunique) mumkin. Agarda aniqlovchi noyob bo'lsa, uning qiymati mohiyat nusxasining faqat bittasini aniqlaydi. Agarda aniqlovchi noyob bo'lmasa, uning qiymati mohiyat nusxalarining qandaydir to'plamini aniqlaydi. Masalan, "Zachetka\_raqami" aniqlovchisi noyob, "Familiya", "Ism", "Otasi\_ismi" aniqlovchilari noyob emas, chunki xuddi shunday familiya, ism va otasining ismidagi ikkita va undan ortiq odam bo'lishi mumkin.

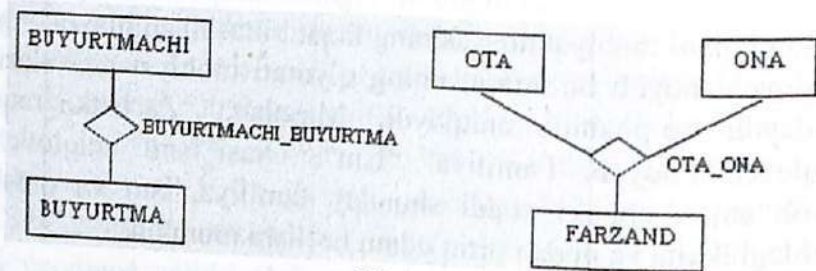
Bir nechta atributlardan iborat aniqlovchilar *kompozit aniqlovchilar*, deyiladi. Misol, "Hudud\_kodi", "telefon\_raqami" aniqlovchilari.

*Aloqa (relationship)* – ko'rib chiqilayotgan soha uchun muhim bo'lgan ikkita mohiyat o'rasidagi nomlangan uyushma. Mohiyatlarning o'zaro munosabatlari *aloqalar* orqali ifodalanadi. Bog'lanish – bu mohiyatlar o'rasidagi uyushma bo'lib, bunda bitta mohiyatning har bir nusxasi ikkinchi mohiyatning ixtiyoriy sondagi (shu jumladan 0) nusxalari bilan bog'liq va aksincha.

«Mohiyat-aloqa» modellari, aloqalar sinfini va aloqalarning nusxalarini o'z ichiga oladi. *Aloqalar sinfi (relationships classes)* – mohiyatlar sinflari orasidagi o'zaro munosabatlardan, *aloqalar nusxalari (relationships instances)* esa, mohiyatlar nusxalari orasidagi o'zaro munosabatlardan iborat. Aloqalarning ham xususiysi bo'lishi mumkin.

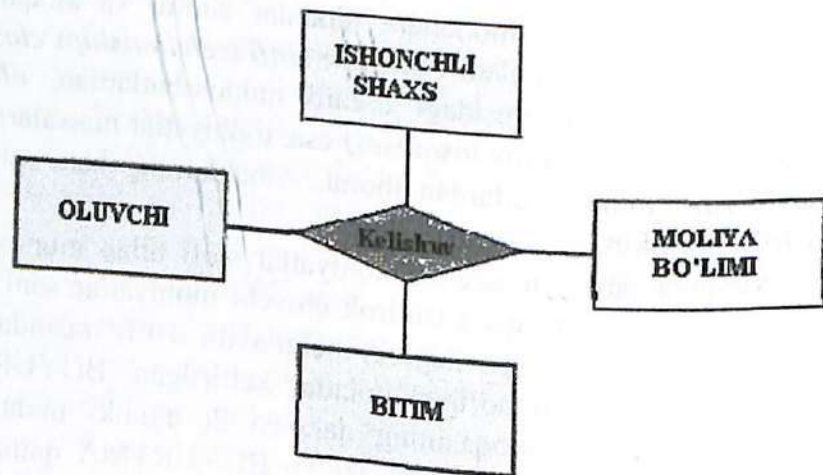
Aloqalar sinfi bir nechta mohiyatlar sinfi bilan munosabatda bo'lishi mumkin. Aloqada ishtirok etuvchi mohiyatlar soni, *aloqa darajasini* (relationships degree) belgilaydi. 10.1- rasmda aloqa darajasi ikki va uch bo'lgan holatlar keltirilgan. BUYURTMA-CHI\_BUYURTMA aloqasining darajasi 2, chunki unda ikkita mohiyat sinflari BUYURTMACHI va BUYURTMA qatnashgan. OTA\_ONA aloqasining darajasi 3, chunki unda uchta mohiyatlar

sinflari OTA, ONA va FARZAND ishtirok etgan. Aloqa darajasi ikkiga teng bo'lgan holatlar juda keng tarqalgan bo'lib, ularni *binar (ikkilik) aloqalar* (binary relationships) deb ataladi. Aloqaning darajasi 3 bo'lsa, *uch tomonlama aloqa* (ternary) deb ataladi.



10.1- rasm.

To'rtinchi darajali aloqa *to'rt tomonlama aloqa* (quaternary) deb ataladi. 10.2- rasmda to'rt tomonlama kelishuv aloqasi keltirilgan. Unda to'rtta qatnashuvchi mohiyatlar, ya'ni OLUVCHI, ISHONCHLI\_SHAXS, MOLIYA\_BO'LIMI va BITIM qatnashgan. Bu aloqa bo'yicha, oluvchi ishonchli shaxs tomonidan maslahat olib va moliya bo'limi qo'llab-quvvatlagan holda, bitim tuzadi.



10.2- rasm.

### 10.3. Binar aloqalarning uch turi

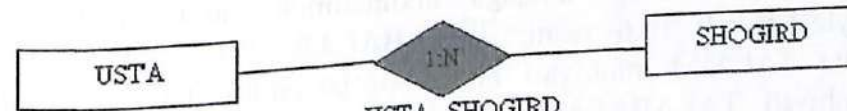
Binar aloqalar uch turga bo'linadi. 10.3- rasmda bitta mohiyat nusxasi boshqa turdagi bitta mohiyat nusxasi bilan 1:1 turdagi aloqa orqali bog'langan. TALABA-KOMPYUTER aloqasi TALABA sinfidagi bitta mohiyatni KOMPYUTER sinfidagi bitta mohiyat bilan bog'lagan. Bu bog'lanishga asosan, har bir talabaga faqat bitta kompyuter biriktirib qo'yilgan va bitta kompyuter faqat bitta talabaga tegishli, degan ma'no yotadi.



10.3- rasm. 1:1 aloqa

10.4- rasmda ikkinchi turdagi binar aloqaga misol keltirilgan. Bu aloqa «birga-ko'p» deb ataladi. Masalan, hunarmandlikdagi USTA-SHOGIRD aloqasini olsak, USTA mohiyat sinfidagi yagona nusxa SHOGIRD mohiyat sinfidagi bir nechta nusxalar bilan bog'langan. Ushbu bog'lanishga nisbatan, ustaning Nta shogirdi bor, bu shogirdlarning hammasini ustozini ana shu bitta USTA, degan ma'no yotadi.

Bundan tashqari 1 : N shakldagi yozuvning ham o'z ma'nosi bor. 1 raqami USTA joylashgan tomonda, N esa SHOGIRD joylashgan tomonda turibdi. Agarda teskarisi yozilganda edi, shogird bitta va uning esa bir vaqtda bir nechta ustozini bor, degan ma'no hosil bo'lar edi. Bu esa haqiqatan yiroq.



10.4- rasm. 1:N aloqa

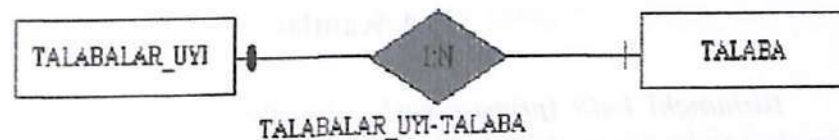
Binar aloqaning uchinchi xili  $N:M$  bo'lib, «ko'plikdan-ko'plikka» ma'nosini anglatadi. 10.5- rasmdagi TALABA-TO'GARAK aloqasi bo'lib, TALABA mohiyatlar sinfiga tegishli nusxalarni TO'GARAK mohiyatlar sinfiga tegishli nusxalar bilan bog'lagan. Ushbu aloqa, bitta talaba bir nechta to'garakka a'zo bo'lishi va bitta to'garakda bir nechta talabalar bo'lishi mumkinligini anglatadi.



10.5- rasm.  $N:M$  aloqa

Romb ichidagi aloqalarni sonini ifodalovchi raqamlar, har bir tomondagi mohiyatlarning maksimal nusxalarini bildiradi. Bu chegaralar *maksimal kardinal sonlar*, bunday ikkita chegaradan iborat ikki tomonli aloqa birligi *aloqaning maksimal kardinalligi* (maximum cardinality) deb ataladi. Masalan, 10.5- rasmdagi aloqaning maksimal kardinalligi  $1:N$ . Lekin albatta, bu turdagi aloqalardagi kardinallik boshqacha ham bo'lishi mumkin. Masalan, *VOLEYBOL\_KOMANDA* va *O'YINCHI* mohiyatlari orasidagi aloqa  $1:6$ , bu voleybol komandasida maydonda bir paytda oltitadan ko'p o'yinchi bo'lishi mumkin emasligini bildiradi.

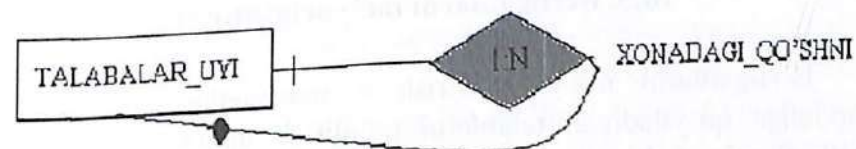
Minimal kardinallikni ko'rsatishning bir nechta usullari mavjud. Ulardan biri 10.6-rasmda keltirilgan. Aloqada qatnashishi majbur bo'lgan mohiyat tomondagi aloqaga perpendikular chiziqcha tortiladi, aloqada qatnashishi mumkin, lekin shart bo'lmagan mohiyat tomondagi aloqaga tuxumsimon yassi shakl (oval) joylashtiriladi. 10.6- rasm, *TALABALAR\_UYI* mohiyati kamida bitta *TALABA* mohiyati bilan bog'langanini, lekin *TALABA* mohiyati *TALABALAR\_UYI* mohiyati bilan bog'lanishi shart emasligini bildiradi.



10.6- rasm.

*TALABALAR\_UYI-TALABA* aloqasiga qo'yilgan to'liq cheklashlarning ma'nosi quyidagicha, *TALABALAR\_UYI* mohiyatining minimal kardinalligi birga va maksimal kardinalligi soni «ko'p» *TALABA* mohiyatidan iborat. *TALABA* nolga teng minimal kardinallikka va maksimal kardinalligi *TALABALAR\_UYI* mohiyatining bitta nusxasidan iborat.

Bitta sinfga tegishli mohiyatlar orasida ham aloqalar bo'lishi mumkin. Misol uchun *TALABA* mohiyatlar sinfi uchun *XONADAGI\_QO'SHNI* aloqasini aniqlash mumkin. Bunday aloqaga misol 10.7- rasmda keltirilgan. *XONADAGI\_QO'SHNI* aloqasining nusxalari Jalil, Halim, Salim yoki boshqa ixtiyoriy talaba bo'lishi mumkin. Bitta sinfga tegishli mohiyatlar orasidagi aloqalar *rekursiv aloqalar (recursive relationships)*, deb ataladi.



10.7- rasm.

Yuqorida keltirilgan uch xildagi binar aloqalarni ba'zan «*MAVJUD*» aloqa turi yoki *egalik aloqalari* (HAS-A relationships) deb atashadi. Buning sababi, bitta mohiyatning boshqa mohiyat bilan aloqasi *mavjud* (has) ligini, ya'ni aloqa borligini bildiradi. Masalan, talabaning kompyuteri bor, ustaning shogirdi bor, to'garakka yozilgan talabalar bor.

#### 10.4. Kalitlar

**Birlamchi kalit (primary key)** – bu mohiyat nusxasini noyob tarzda aniqlaydigan atribut yoki atributlar guruhi bo'lishi mumkin. Diagrammadagi birlamchi kalit atributlar maxsus belgilanadi, agar da belgilanmasa – bu gorizontaal chiziq ustidagi atributlar ro'yxatida joylashgan birinchi atribut bo'ladi.

Bitta mohiyatda asosiy kalitga da'vogar, bir nechta atributlar yoki atributlar termasi bo'lishi mumkin. Bunday da'vogorlar **nomzod kalitlar (candidate key)**, deb nomlanadi.

Kalitlar murakkab bo'lishi mumkin, ya'ni bir nechta atributlarni o'z ichiga oladi. Har bir mohiyat kamida bitta nomzod kalitga ega bo'lishi kerak. Ko'pgina mohiyatlar faqat bitta nomzod kalitga ega bo'ladi. Bunday kalit birlamchi bo'ladi. Ba'zi mohiyatlar bir nechta mumkin bo'lgan kalitlarga ega bo'lishi mumkin. Keyin ulardan biri birlamchi, qolganlari esa muqobil kalitlarga aylanadi.

**Muqobil kalit (alternate key)** – bu birlamchi kalit bo'la olmagan nomzod kalitdir.

#### 10.5. Berilganlarni me'yorlashtirish

Berilganlarni me'yorlashtirish – berilganlarning relyatsion modeliga qo'yiladigan talablarni qondirish uchun mohiyalar va atributlarni tekshirish va qayta tashkil etish jarayoni. Me'yorlashtirish har bir atribut o'z mohiyati uchun aniqlanganligiga ishonch hosil qilish, shuningdek, berilganlarni saqlash uchun xotira hajmini sezilarli darajada kamaytirish va berilganlarni saqlashni tashkil etishdagi me'yordan chekinishlarni bartaraf etish imkonini beradi. Me'yorlashtirish natijasida berilganlar tuzilishi yaratilishi kerak, unda har bir dalil to'g'risidagi berilganlar faqat bitta joyda saqlanadi. Me'yorlashtirish jarayoni berilganlar tuzilmasini me'yoriy shakllarga – berilganlarni tashkil qilish uchun rasmiylashtirilgan

talablarga ketma-ket olib kelishgacha kamayadi. Oltita me'yoriy shakl mavjud [2].

#### 10.6. Domenlar

Domenni atribut qiymatlari olinadigan qiymatlar to'plami sifatida aniqlash mumkin. Har bir atribut faqat bitta domenda aniqlanishi mumkin. Domen tushunchasi nafaqat berilganlar turini, balki berilganlar qiymatlari maydonini ham o'z ichiga oladi. Masalan, "Yosh" domenini butun musbat son sifatida aniqlash va "Xizmatchi\_yoshi" atributini ushbu domenga tegishli, deb aniqlash mumkin.

Domenlar – berilganlar bilan ishlashni loyihalash bosqichida ishlab chiquvchilarni ham, tizimning faoliyati davomida berilganlar bazasi ma'murlarining ham ishini osonlashtiradi. Mantiqiy darajada domenlarni aniq fizik xususiyatlarsiz tavsiflash mumkin. Jismoniy darajada ular avtomatik ravishda qo'lda o'zgartirilishi mumkin bo'lgan o'ziga xos xususiyatlarga ega bo'ladilar. Shunday qilib, "yosh" domeni mantiqiy darajada raqam turiga ega bo'lishi mumkin, jismoniy darajada domen ustuniga *Integer* turi beriladi.

Masalan, ko'chmas mulk ob'yektidagi xonalar soni 1 tadan 25 tagacha bo'lishi mumkin desak, demak, tegishli atributning turidagi qiymatlar birdan yigirma beshgacha bo'lgan butun sondan iborat bo'ladi. Turli atributlar bitta domendan birgalikda foydalanishi mumkin. Domenlar, shuningdek, boshqa domenlarning kombinatsiyasidan ham tuzilishi mumkin. Berilganlarning to'liq yaratilgan modeli har bir atributning domenlarini o'z ichiga olgan holda ER-modelda tasvirlanadi.

Bo'linmas atributlar *oddiy*, bo'linuvchi atributlar esa *tarkibli* hisoblanadi. Masalan, Maosh atributi oddiy atribut, Manzil atributi esa tarkibli atribut, chunki uni bir nechta tobe atributlarga ajratish mumkin.

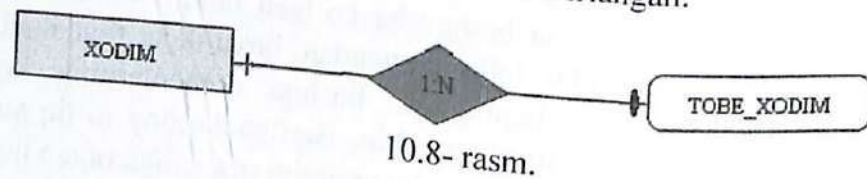
### 10.7. Zaif va kuchli mohiyatlar

«Mohiyat-aloqa» modelida mohiyatning ikkita maxsus turi aniqlangan bo'lib, ulardan biri *zaif mohiyat (weak entity)*, deb ataladi. *Zaif mohiyatlarga*, mavjudligi boshqa mohiyatlarga mantiqan bog'liq bo'lgan mohiyat kiradi. Mavjudligi boshqa mohiyat turiga bog'liq bo'lmagan mohiyat turi, *kuchli mohiyat (strong entity)*, deb ataladi.

Zaif mohiyatlarga qanday mohiyatlar kirishini aniqlashtirish uchun XODIM va TOBE\_XODIM mohiyatlar sinflaridan tuzilgan berilganlar bazasini ko'raylik.

Faraz qilaylik, o'rnatilgan ish tartibiga mos ravishda XODIM mohiyat nusxasi TOBE\_XODIM sinfidagi hech bir mohiyat bilan aloqa o'rnatmagan holda mavjud bo'lishi mumkin, lekin TOBE\_XODIM mohiyati esa XODIM sinfidagi mohiyatlar bilan aloqasiz mavjud bo'lishi mumkin emas. U holda, TOBE\_XODIM mohiyati zaif bo'ladi. Bu esa, TOBE\_XODIM mohiyatiga tegishli bo'lgan berilganlar bazasida, ushbu mohiyat faqatgina XODIM sinfidagi birorta mohiyat bilan aloqasi bo'lsagina, paydo bo'lishi mumkinligini bildiradi.

10.8- rasmda ana shu vaziyat keltirilgan bo'lib, zaif mohiyat burchaklari yassilangan to'rtburchakda tasvirlangan.



10.8- rasm.

Mohiyatni zaiflar toifasiga kiritish uchun, u mantiqan boshqa mohiyatga bog'liq bo'lishi kerak. Ayrim holatlarda USTOZ va TALABA mohiyatlar orasidagi aloqadan kelib chiqqan holda, TALABA mohiyati zaif deb hisoblanadi, chunki talabaning ustoz bo'lishi kerak va u berilganlar bazasiga faqat USTOZ sinfiga

tegishli mohiyatlardan biri bilan aloqada bo'lishi kerak, deb yuritiladi. Lekin talaba fizik va mantiqiy jihatdan ham ustoz bilan bog'lanmagan, shuning uchun ham TALABA mohiyati kuchli mohiyat hisoblanadi. Talaba ish tartibi yo'l qo'ymasa ham, mantiqiy jihatdan ustozsiz ham mavjud bo'lishi mumkin.

Ushbu tasdiqni yanada yaxshi namoyish qilish uchun bir nechta misollarni ko'ramiz. Faraz qilaylik, ta'limga oid berilganlar bazasi FAN\_NOMI va REYTING\_BALI mohiyatlarini o'z ichiga olgan bo'lsin va ular orasida aloqa o'rnatilgan bo'lsin (10.9-rasm). REYTING\_BALI mohiyatining ixtiyoriy nusxasi FAN\_NOMI sinfidagi mohiyat bilan albatta aloqada bo'lishi mantiqan shart, chunki reyting bali o'z holicha mavjud bo'la olmaydi, lekin ixtiyoriy fan reyting balisiz ham mavjud. Demak, FAN\_NOMI mohiyati kuchli, REYTING\_BALI mohiyati esa zaif mohiyat.



10.9- rasm.

Endi BEMOR va RETSEPT mohiyatlari orasidagi 10.10- rasmdagi diagrammaga e'tibor beraylik. Retsept bemorsiz mantiqan mavjud emas. RETSEPT mohiyatining minimal kardinal soni birga teng bo'lsa ham, u BEMOR mohiyatisiz mavjud bo'la olmaydi. Demak, RETSEPT zaif, BEMOR esa kuchli mohiyat.



10.10- rasm.

BUYURTMA va SOTUVCHI mohiyatlari orasidagi aloqani ko'rib chiqaylik (10.11- rasm). SOTUVCHI mohiyatining minimal kardinalligi 1, bu mantiqiy zaruratdan emas, balki ish tartibidan kelib chiqqan holda shunday bo'ladi. Buyurtmani huquqiy shaxs

ham berishi mumkin. Shunday ekan, demak, BUYURTMA mohiyati SOTUVCHI mohiyatisiz ham mavjud bo'lishi mumkin. Xulosa, BUYURTMA va SOTUVCHI mohiyatlarining ikkalasi ham kuchli.



10.11- rasm.

Zaif mohiyatlarni ba'zan *voris, bog'liq yoki tobe mohiyat*, kuchli turdagi mohiyatlarni esa *ajdod, ega mohiyat yoki dominant mohiyat* deb atashadi.

«Mohiyat-aloqa» modelida zaif mohiyatlarning o'zi alohida turga ega, ularni *aniqlangan bog'liq mohiyatlar (ID dependent entities)*, deb atashadi. Bular shunday mohiyatlarki, ularning aniqlovchilari boshqa mohiyatni aniqlovchilarini saqlaydi. Masalan, ko'p qavatli uylardagi UY va XONADON mohiyatlari orasidagi aloqani ko'rib chiqaylik (10.12- rasm).



10.12- rasm.

UY mohiyatining aniqlovchisi Uy\_raqami atributi bo'lsa, XONADON mohiyatining tarkibli aniqlovchisi esa (Uy\_raqami, Xonadon\_raqami) atributlaridan iborat bo'ladi. XONADON mohiyatining aniqlovchisi UY mohiyatini aniqlovchisidan (Uy\_raqami) iborat bo'lgani uchun, XONADON mohiyati UY mohiyatiga aniqlash bo'yicha bog'liq. Boshqacha aytganda, XONADON mohiyati ham mantiqiy, ham fizik jihatdan uysiz mavjud bo'la olmaydi.

## 10.8. Bir va ko'p qiymatli atributlar

*Bir qiymatli atribut* deganda, ma'lum bir turdagi mohiyat nusxasining bitta qiymatini saqlovchi atribut tushuniladi. Ko'pgina atributlar bir qiymatli bo'ladi. OLUVCHI mohiyatining har bir alohida nusxasi uchun doimiy ravishda Oluvchi\_raqami atributida yagona qiymat mavjud, masalan X004. Shuning uchun Oluvchi\_raqami atributi bir qiymatli bo'ladi.

*Ko'p qiymatli atribut* deganda, ma'lum bir turdagi mohiyat nusxasining bir nechta qiymatini saqlovchi atribut tushuniladi. Ba'zi bir atributlar mohiyatning har bir nusxasi uchun bir nechta atributdan iborat bo'lishi mumkin. Masalan, OLUVCHI mohiyatida Telefon\_raqami atributining bir nechta raqami bo'lishi mumkin. Oluvchi\_raqami atributidagi X004 raqamli mahsulot oluvchida 233-45-55, 233-45-56 va 233-25-39 raqamli telefonlar bo'lishi mumkin. Demak, Telefon\_raqami atributi ko'p qiymatli bo'ladi. Ko'p qiymatli atributlardagi qiymatlar soni albatta ikki tomondan chegaralangan bo'lishi shart.

## 10.9. Mohiyatlarning ostki va ustki turlari

Ayrim mohiyatlarda zarur bo'lmagan atributlar ham bo'ladi, bunday mohiyatlar *ostki turlar* (subtypes) orqali ifodalanadi. Masalan, Mijoz\_raqami, Mijoz\_nomi va To'lov\_miqdori atributlariga ega bo'lgan MIJOZ mohiyatini ko'rib chiqaylik. Faraz qilaylik, mijoz fizik shaxs, jamiyat yoki vazirlik va bu mijozlarning turiga nisbatan ba'zi bir qo'shimcha ma'lumotlarni ko'rsatish zarur bo'lsin. Bu ma'lumotlar quyidagicha bo'lsin:

JISMONIY\_SHAXS – Familiya\_Ism,  
Soliq\_to'lovchining\_shaxsiy\_raqami, Manzili  
JAMIYAT – Nomi, Hisob\_raqami, Soliq\_to'lovchining\_shaxsiy\_raqami, Manzili

VAZIRLIK –Nomi, Hisob\_raqami, Soliq\_to'lovchining\_shaxsiy\_raqami, Manzili.  
 Keltirilgan atributlarni hammasini MIJOZ mohiyati tarkibiga kiritilgandan keyingi atributlar tarkibi 10.13- rasmda keltirilgan:

Mijoz\_raqami  
 Mijoz\_nomi  
 To'lov\_miqdori  
 Familiya\_Ism  
 Soliq\_to'lovchining\_shaxsiy\_raqami  
 Manzili  
 Nomi  
 Hisob\_raqami

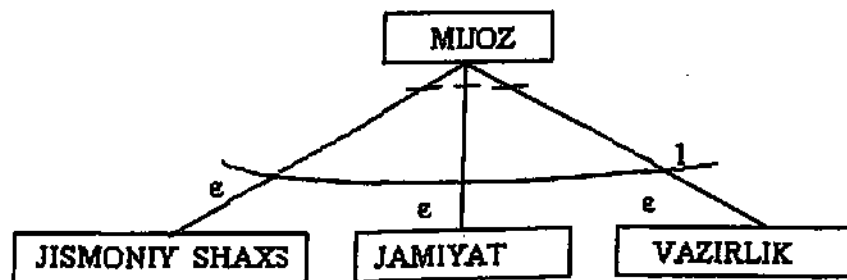
10.13- rasm.

Lekin bu holda, MIJOZ mohiyatidagi ayrim atributlar keraksiz bo'lib qoladi. Masalan, jamiyat va vazirlik uchun Familiya\_Ism atributi kerak emas yoki fizik shaxs uchun Nomi va Hisob\_raqami atributlari kerak emas, shunday ekan, bu atributlar qiymatga ega bo'la olmaydi.

Aniq vaziyatga eng yaqin holat, bu MIJOZ mohiyatini uchta ostki turga ajratishdan iborat.

MIJOZ mohiyati tarkibi – Mijoz\_raqami, Mijoz\_nomi va To'lov\_miqdori;  
 JISMONIY\_SHAXS tarkibi – Familiya\_Ism, Soliq\_to'lovchining\_shaxsiy\_raqami, Manzili;  
 JAMIYAT tarkibi – Nomi, Hisob\_raqami, Soliq\_to'lovchining\_shaxsiy\_raqami, Manzili;  
 VAZIRLIK tarkibi – Nomi, Hisob\_raqami, Soliq\_to'lovchining\_shaxsiy\_raqami, Manzili.  
 10.14- rasmda JISMONIY\_SHAXS, JAMIYAT va VAZIRLIK mohiyatlari MIJOZ mohiyatining ostki turlari sifatida tasvirlangan. MIJOZ mohiyati esa JISMONIY\_SHAXS, JAMIYAT va VAZIR-

LIK mohiyatlari uchun *ustki tur (supertype)* hisoblanadi. Aloqa chiziqlari yonidagi *e* belgisi JISMONIY\_SHAXS, JAMIYAT va VAZIRLIK mohiyatlari MIJOZ mohiyatining ostki turlari ekanligini bildiradi. 1 raqamli egri chiziq MIJOZ mohiyati bitta va faqat bitta ostki turga tegishli ekanligini bildiradi. Bu esa, ostki turlar o'zaro bir-birini o'rmini bosadi (o'zaro almashadi) va ulardan faqat bittasi zarur, degan ma'noni beradi.



10.14- rasm.

### 10.10. «Mohiyat-alloqa» diagrammalari

Yuqoridagi 10.3 – 10.5- rasmlardagi chizmalarni yoki mohiyatlarni ular orasidagi aloqalar hamda aloqalarning maksimal kardinalligi ko'rsatilgan holda maxsus chizma tarzida ifodalash «mohiyat-alloqa» diagrammalari yoki ER-diagrammalar (entity-relationship diagrams, ER-diagrams) deb ataladi. Bu diagrammalar standartlashtirilgan, lekin juda qat'iy ravishda emas. Bu standartga asosan *mohiyatlar* – to'g'ri to'rtburchaklar, *atributlar* – ellipslar, *alloqalarning nomi* – romblar ichida tasvirlanadi. Ba'zan birlik aloqalar  $\rightarrow$  tarzda, ko'plik aloqalar esa  $\leftarrow$  shaklda belgilanadi. Keyinchalik ko'pgina mualliflar ana shunday diagrammalarning o'zlari yaratgan naqlarini taklif qildilar (Dj. Martin notatsiyalari, IDEFIX belgilari, Barker notatsiyasi va h.k.). Bundan tashqari, bir xildagi notatsiyani amalga oshirgan dasturlar ham o'z

imkoniyatlari bo'yicha turlicha bo'lishi mumkin. Aslini olganda, «mohiyatlar-aloqalar» diagram-masining hamma naqllari rasm matnli tavsifdan yaxshi, deyilgan fikrga asoslanadi. Hamma ana shunday diagrammalar ob'yekt mohiyatining grafik tasviridan, ularning atributlari va mohiyatlar orasidagi aloqalardan foydalanadilar.

«Mohiyat-aloqa» diagrammasining elementlaridan tuzilgan, mahsulotning sotuvchigacha bo'lgan BBning modeliga misol 10.15- rasmda keltirilgan.

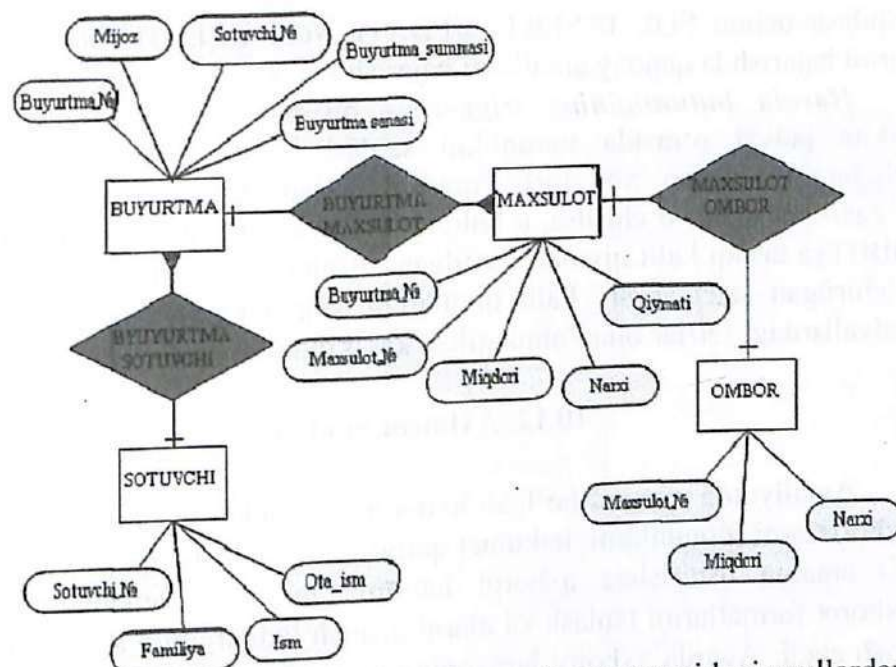
Undagi diagrammadan ko'rinib turibdiki, BUYURTMA va MAXSULOT mohiyatlari mos ravishda beshtadan atributga – Buyurtma\_No, Mijoz, Sotuvchi\_No, Buyurtma\_summasi, Buyurtma\_sanasi; Buyurtma\_No, Mahsulot\_No, Miqdori, Narxi, Qiymati; SOTUVCHI mohiyati to'rtta atributga – Sotuvchi\_No, Familiya, Ism, Ota\_ismi; OMBOR mohiyati esa uchta atributga ega – Maxsulot\_No, Miqdori, Narxi.

BUYURTMA-SOTUVCHI aloqasining maksimal kardinalligi N:1, BUYURTMA-MAXSULOT aloqasining maksimal kardinalligi 1:N, MAXSULOT-OMBOR aloqasining maksimal kardinalligi esa 1:1.

Agarda mohiyatning atributlari ko'p bo'lsa, tasvir ancha murakkablashib, qabul qilish qiyinlashadi. Bunday hollarda mohiyatlarga tegishli atributlar alohida ro'yxatlarda keltiriladi.

### 10.11. Triggerlar va saqlangan protseduralar

Triggerlar va saqlangan protseduralar – SQL kodining nomlangan bloklari bo'lib, ular so'rovlarni qayta ishlash, berilganlarni tasdiqlash va boshqa tez-tez bajariladigan funksiyalarni tezda bajarish uchun oldindan tuziladi va serverda saqlanadi. Serverda kodni saqlash va bajarish, BB bilan ishlaydigan har bir dasturda emas, balki faqat bir marta kodni yaratishga imkon beradi. Bu das-



10.15- rasm. «Mohiyat-aloqa» diagrammasidagi usullardan biri.

turlarni yozish va ularga hamroh bo'lish vaqtini tejaydi. Shu bilan birga, qaysi mijoz ilovasining berilganlariga kirishidan qat'i nazar berilganlar yaxlitligi va tijorat qoidalarini qo'llab-quvvatlash kafolatlanadi.

Triggerlar va saqlangan protseduralar mijoz dasturidan tarmoq orqali uzatilishi shart emas, bu esa tarmoq trafiginini sezilarli darajada kamaytiradi.

**Saqlangan protsedura** – bu mijoz dasturidan yoki boshqa saqlangan protseduradan chaqirilishi mumkin bo'lgan, oldindan tuzilgan SQL buyruqlarining nomlangan termasi.

**Trigger** – bu hodisaga ta'sir sifatida avtomatik ravishda bajariladigan protsedura. Bunday hodisa mavjud jadvalga satr qo'shish, o'zgartirish yoki o'chirish bo'lishi mumkin. Trigger BBTga serverda bajariladigan qo'shimcha funksiyalarni ta'

minlash uchun SQL INSERT, UPDATE yoki DELETE buyruqlarini bajarishda qanday amallarni bajarish kerakligini aytadi.

**Havola butunligining triggeri** – bu bir-biriga bogʻlangan ikkita jadval oʻrasida butunlikni saqlash uchun ishlatiladigan triggerning oʻziga xos turi. Agar bitta jadvalga satr kiritilsa, oʻzgartirilsa yoki oʻchirilsa, u holda mos yozuvlar butunligi trigger BBBTga tashqi kalit qiymati kiritilgan satrning (oʻzgartirilgan yoki oʻchirilgan satr) asosiy kalit qiymatiga toʻgʻri keladigan boshqa jadvallardagi satrlar bilan nima qilish kerakligini aytadi.

### 10.12. Axborot modeli

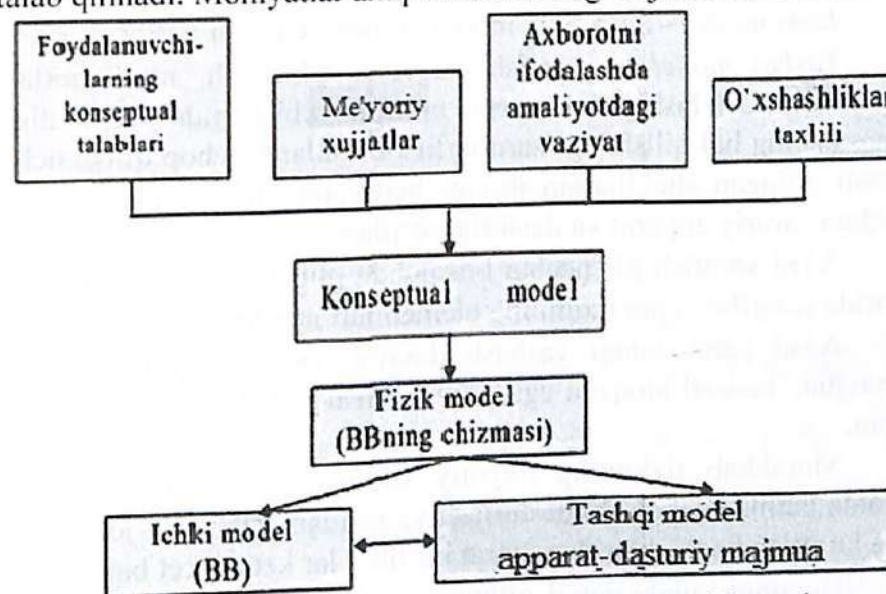
Amaliyotda mavjud boʻlgan axborot jarayonlarini va meʼyotiy axborotlarni (qonunlarni, hukumat qarorlarini, sohadagi standartlarni) amalga oshirishda axborot hajmini aniqlash, saqlanadigan axborot formatlarini tanlash va ularni uzatish juda muhim ahamiyat kasb etadi. Agarda axborot jarayonining qayta qurishda tub oʻzgarishlar boʻlmasa, u holda bu axborotni saqlash formatlarini va ularni qayta ishlashdagi jurnal, jadval yoki boshqa biron-bir qogʻozda saqlash usullaridan ham foydalanish tavsiya etiladi.

Lekin avvaldan, axborotlarning yangi hisob tizimlariga oʻtishi, saqlanishi va ishlov berilishi masalalarini bozorda mavjud boʻlgan yirik axborot kompaniyalari tomonidan yaratilgan oʻxshash dastur mahsulotlarini, qoʻyilgan masalalarga qisman yoki toʻliq mos kelishi nuqtayi nazaridan tahlil etilishi lozim.

Axborot modelini shakllantirish chizmasi 10.16- rasmda keltirilgan.

**Konseptual model** (10.16- rasmga qarang) – axborot obʼyektlarini, ularning atributlari, ular orasidagi aloqani koʻrsatadi, lekin axborotlarni saqlashning fizik usullarini koʻrsatmaydi (predmet sohaning modeli, baʼzida uni mantiqiy axborot modeli deb ataladi). Axborot obʼyekti odatda mohiyat deb atalib, alohida obʼyektlar yoki voqealardan iborat boʻlib, ular haqidagi maʼlumotlarni saqlash

talab qilinadi. Mohiyatlar aniq atributlarning majmuasidan tuziladi.



10.16- rasm. Axborot modelining chizmasi.

**Fizik model** – axborot obʼyektlarining bazalaridagi hamma atributlarni (atributlarni) va ular orasidagi aloqalarni, bazalarni saqlash usulini (qaysi BBBT foydalanishni hisobga olgan holda) akslantiradi.

Jismoniy modelning ikkita darajasi mavjud:

- transformatsion model;
- BBBT modeli.

Transformatsion model umumiy ATning bir qismi boʻlishi mumkin boʻlgan va predmet sohaning qism-toʻplamini tavsiflashi mumkin boʻlgan alohida loyihani amalga oshirish uchun maʼlumotlarni oʻz ichiga oladi. Ushbu model loyihachilar va BB maʼmurlariga berilganlar lugʻatida qaysi BB obʼyektlari saqlanishini yaxshiroq tasavvur qilish va jismoniy model AT talablariga qanchalik mos kelishini tekshirish imkonini beradi.

BBBT modeli avtomatik ravishda transformatsiya modelidan hosil

bo'radi va BBBT tizim katalogining aniq ko'rinishini akslantiradi.

*Ichki model* – aniq fizik modelga mos keluvchi BBdir.

*Tashqi model* – yaratish, saqlash, tahrirlash, ma'lumotlarni o'chirish va izlash jarayonlarini ta'minlovchi hamda yechiladigan masalaning hal qilishdagi zaruriy hisoblashlar va chop qilish uchun hosil qilingan shakllardan iborat, berilganlar bazasi bilan ishlash uchun zaruriy apparat va dasturlar to'plami.

ATni yaratish bir nechta bosqichda olib boriladi, ularning har birida yaratilayotgan tizimning elementlari aniqlashtiriladi.

ATni yaratishdagi yashash davrini tasvirlovchi turli usullar mavjud. Teskari aloqaga ega pog'ona usuli 10.17- rasmda keltirilgan.

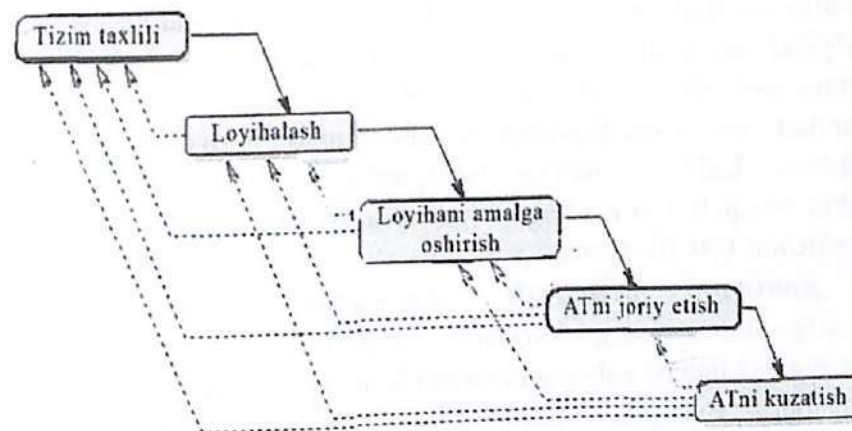
Murakkab tizimning hayotiy siklini ishlab chiqish bunday holda tahlil, loyihalash, dasturlash va testdan o'tkazish, joriy qilish va kuzatish bosqichlaridan iborat bo'lib, ular ketma-ket bajariladi.

Bugungi kunda qabul qilingan me'yorlarda AT loyihalarining ustida quyidagilar ishlanadi:

- predmet sohaga taalluqli tijorat-jarayonni o'rganuvchi va ularni modellashtiruvchi tijoratchi-tahlilchilar;
- yechimlarni ko'rishni, masalalar, ilovalar va ma'lumotlarni loyihalashtiruvchi tizimli tahlilchi va ma'murlar;
- kodlar ilovasining mualliflari;
- sinovdan o'tkazish va sifatiga baho beruvchi mutaxassislar;
- hujjatlash mualliflari;
- distributivlar mualliflari;
- joriy qilish bo'yicha mutaxassislar.

Odatda, bu funksiyalar har xil soha mataxassislari orasida taqsimlanadi, lekin amaliyotda ba'zida bitta mutaxassis bir nechta vazifani bajarishi mumkin.

AT ob'yektlarini yaratishdagi loyihalash va dasturlash bosqichlarida ob'yektga yo'naltirilgan usuldan foydalanish mumkin (vorislik, inkapsulatsiya, polimorfizm).



10.17- rasm. ATning pog'onali yashash davrining chizmasi

### 10.13. Predmet sohani modellashtirish darajalari

Berilganlar bazasi aniq bir *predmet sohani*, lozim topilgan tadqiqot uchun zarur bo'lgan, haqiqiy dunyodagi bir qism (bir yoki bir nechta ob'yektlar)ni akslantiradi.

Berilganlar bazalarining yashash (mavjudlik) davridagi asosiy bosqichlardan biri, bu berilganlar bazalarini loyihalash bo'lib, undagi aniq uslubiyatlardan biri relatsion berilganlar bazalarini loyihalashdan iborat.

**Loyihalash uslubiyati** – bu aniq tuzilishga ega bo'lgan va maxsus protseduralar, texnik usullar, vositalar va hujjatlardan foydalanishni hisobga olgan hamda loyihalash jarayonini qo'llab-quvvatlash va soddalashtirishga yo'naltirilgan jarayondir. Loyihalash uslubiyati yaxlit jarayonni bir qancha davrlarga bo'lgan bo'lishni, ularning har biri esa bir qancha bosqichlardan iborat bo'lishini nazarda tutadi. Har bir bosqichda yaratuvchiga, ishlab chiqishning shu davriga bog'liq bo'lgan masalalarini yechish uchun texnik usullar majmuasi taklif qilinadi. Bundan tashqari loyihalash uslubiyati rejalash usullari, muvofiqashtirish, boshqarish, loyihani

amalga oshirish bosqichini baholash hamda berilganlar bazalariga qo'yiladigan talablar majmuasini tahlil qilish va modellashtirish uchun tuzilishli yondashuv usulini taklif qiladi, bu harakatlarni standart va tashkillashgan tartibda amalga oshirishga imkon yaratadi. Ushbu uslubiyat berilganlar bazalarini loyihalashdagi uchta bosqichni o'z ichiga oladi – konseptual, mantiqiy va fizik bosqichlar (10.18- rasm).

**Konseptual loyihalash** – eng muhim mohiyatlar turlarining tavsifi, ular orasidagi aloqalar va atributlarni o'z ichiga olgan berilganlar bazasi (ob'yektlar)ning konseptual tasavvurini yaratishdan iborat. Berilganlar bazasini konseptual loyihalash korxonaning berilganlarini konseptual modelini yaratishdan boshlanadi. Bu jarayonda konseptual model amalga oshirish bosqichlaridagi hech bir qism yoki bo'laklarga bog'lanmagan holda yaratiladi.

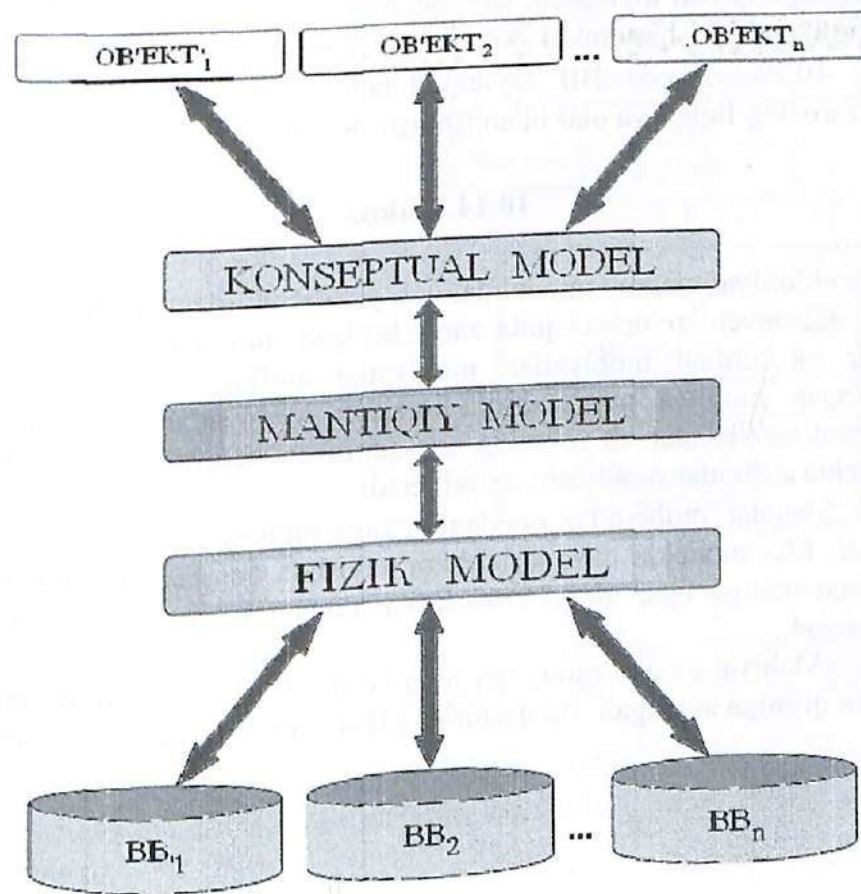
**Mantiqiy loyihalash** – konseptual tasavvurni berilganlar bazalarining mantiqiy tuzilishlariga aylantirish, shu jumladan munosabatlarni loyihalash. Mantiqiy loyihalashdagi berilganlar bazasining tuzilishi, *chizma* deb ataladi.

**Fizik loyihalash** – tanlangan BBB tizimidan foydalangan holda, mantiqiy modelni jadvallar yordamida fizik jihatdan berilganlar bazalariga akslantirish bo'yicha qaror qabul qilishdan iborat. Bunda kompyuter tashuvchilarida berilganlarni joylashtirish usullari aniqlanadi, berilganlarni taqsimlash, ularga kirish va indekslash usullari hisobga olinadi. Zamonaviy amaliy dastur vositalarida tashkillashtirishning bu bosqichi foydalanuvchining ishtirokisiz avtomatik tarzda amalga oshiriladi. Foydalanuvchi, amaliy dasturlar va universal dastur vositalar orqali ishlaganida, BBB tizimlaridagi tushunchalarga asoslangan holda ish olib boradi. Shu nuqtayi nazardan, loyihalashdagi eng asosiy bosqichlar bu konseptual va mantiqiy modellarni yaratishdan iborat bo'ladi.

Turli foydalanuvchilarga mantiqiy modelning turli qism to'plamlaridan foydalanishga to'g'ri keladi. Aniq bir foydalanuvchi nuqtayi nazaridan berilganlar bazasini mantiqiy tuzilishini

tavsiflovchi modelni, *tashqi model*, uning tavsifi esa *qism chizma* deyiladi. Foydalanuvchining tashqi modeli, ana shu foydalanuvchining konseptual talablarini akslantirishdan iborat.

Foydalanuvchi faqatgina qism chizmadagi berilganlarga kirishi mumkin. Bu esa ma'lum darajada berilganlarning himoyasini ta'minlaydi. Har bir qism chizma doirasida berilganlarga ishlovning mumkin bo'lgan holatlari beriladi va bu ham qo'shimcha himoyani ta'minlaydi. Qism chizmalardan foydalanish foydalanuvchining ishini ancha yengillashtiradi, chunki u faqatgina o'ziga tegishli berilganlar bazasining tuzilishini bilishi yetarli bo'ladi.



10.18- rasm.

Loyihalash bosqichlari orasidagi o'zaro bog'lanishlar 10.19- rasmda keltirilgan.

Loyihalash jarayoni konseptual modelni qurishdan boshlanadi. Dastlabki model loyiha oldi bosqichida quriladi va loyihalashning keyingi bosqichlarida yana aniqlashtiriladi. Keyin uning asosida mantiqiy model quriladi. Fizik va tashqi modellar bulardan keyin ixtiyoriy ketma-ketlikda qurilishi mumkin, shu jumladan parallel ravishda ham. Umumiy qurish jarayoni davomida avvalgi bosqichlarga qaytib loyihalash jarayonlarini o'zgartirish yoki modelga aniqliklar kiritish mumkin.

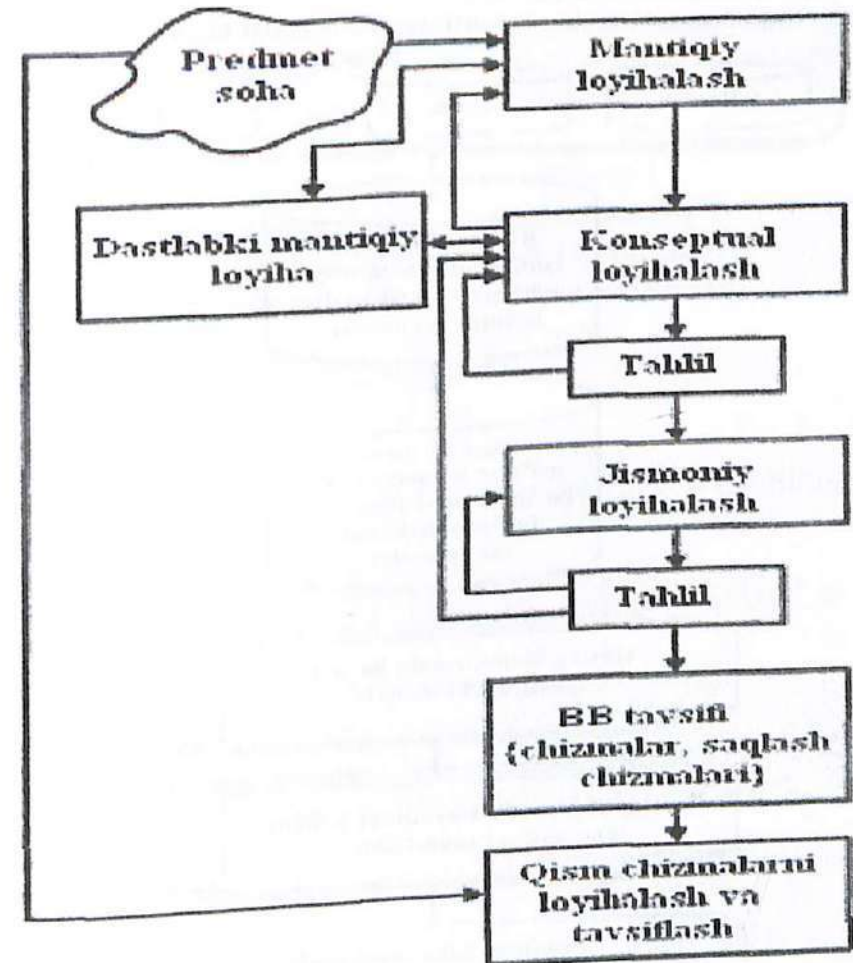
10.20- rasmda BB loyihalashdagi asosiy tushunchalarning o'zaro bog'liqligi va ular bilan ishlash chizmasi keltirilgan.

#### 10.14. Xulosa

«Mohiyat-aloqa» modelida, ob'yektlarni identifikatorlari – foydalanuvchilar uchun juda zarur bo'lgan, mohiyatlar aniqlandi. Bir xil turdagi mohiyatlar, mohiyatlar sinfini tuzadi. Alohida olingan mohiyat, nusxa sifatida ko'riladi. Mohiyatlar atributlardan iborat bo'lib, ular ob'yektning xossalarini tavsiflaydi. Bir yoki bir nechta atributlar mohiyatni tashkil etadi.

Aloqalar mohiyatlar orsidagi o'zaro munosabatlarni akslantiradi. ER- modelda aloqalar oshkor ravishda aniqlanadi – har bir aloqa nomga ega; aloqalar sinfi va aloqa nusxalarining sinflari mavjud.

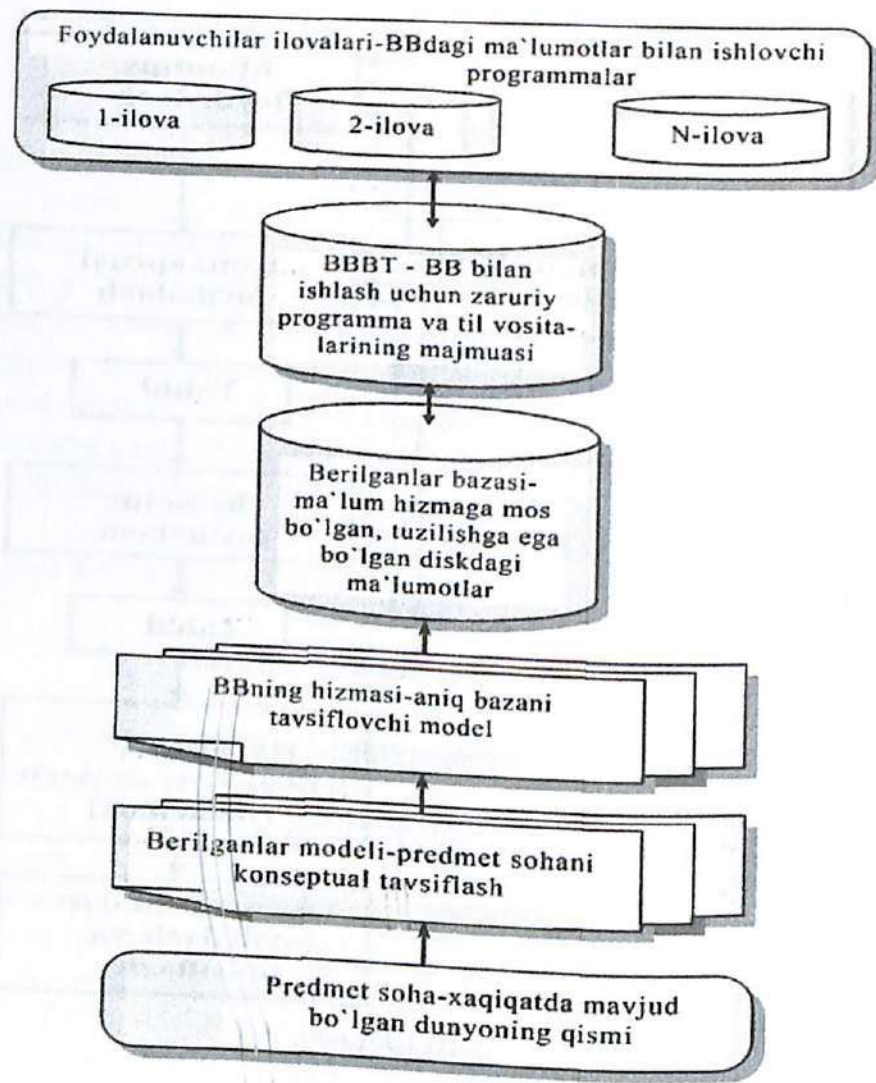
«Mohiyat-aloqa» modellari juda ko'p CASE vositalarning muhim qismiga aylangan. Bu dasturlar ER-diagrammalarni «Mohiyat-



10.19- rasm.

aloqa» modellari juda ko'p CASE vositalarning muhim qismiga aylangan. Bu dasturlar ER-diagrammalarni yaratish va saqlash vositalariga ega.

Modellashni unifikatsiyalashgan (yagonalashgan) tili (UML)da «mohiyat-aloqa» diagrammalarini qurishning yangi shakli taklif qilingan.



10.20- rasm. BB loyihalashdagi asosiy tushunchalarning o'zaro harakati va ular bilan ishlash.

### 10.15. Nazorat uchun savollar

1. «Mohiyat-aloqa» modeli nima uchun ishlatiladi?

2. «Mohiyat-aloqa» modeli qachon va kim tomonidan yaratilgan?
3. «Mohiyat-aloqa» modeli asosiy elementlari nimalardan iborat?
4. Mohiyatlar sinfi deganda nimani tushunasiz?
5. Mohiyat nusxasi nimani bildiradi?
6. Kompozit atribut, deganda nimani tushunasiz?
7. Ko'p qiymatli atribut, deganda nimani tushunasiz?
8. Mohiyani aniqlovchisi, deganda nimani tushunasiz?
9. Kompozit aniqlovchilar, deganda nimani tushunasiz?
10. Aloqalar sinfi, deganda nimani tushunasiz?
11. Aloqalar nusxasini ta'riflang.
12. Aloqa darajasi nimani belgilaydi.
13. Binar va ternar aloqalarni ta'riflang.
14. Binar aloqalar uch turini ta'riflang va unga misollar keltiring.
15. Romb nimani belgilaydi?
16. Kordinallik nimani bildiradi?
17. «Mohiyat-aloqa» diagrammalaridagi maxsus belgilarni ta'riflang.
18. Kuchli mohiyatni ta'riflang va unga misol keltiring.
19. Zaif mohiyatni ta'riflang va unga misol keltiring.
20. Bir qiymatli atributni ta'riflang va unga misol keltiring.
21. Ko'p qiymatli atributni ta'riflang va unga misol keltiring.
22. Mohiyatlarning ostki turlari nimani bildiradi?
23. Mohiyatlarning ustki turlari nimani bildiradi?

## XI- BOB. UML – YAGONALASHGAN MODELLASHTIRISH TILI

Garchi ko'plab umumiy dasturlash tillari kabi bo'lmasa-da, yagonalashgan modellashtirish tili – UMLni rasmiy sun'iy til sifatida qarashimiz mumkin. Sun'iylikning belgisi – uchta mashhur mualliflar – Gradi Buch, Jeyms Rambo va Ivar Yakobsonning mavjudligi. Bu til ana shu mualliflar tomonidan 1995-yilda yaratilgan.



Gradi Buch



Jeyms Rambo



Ivar Yakobson

Rasmiy sun'iy tillarni (xususan, dasturlash tillarini) tasvirlash uchun ko'plab turli usullar ixtiro qilingan va ishlatilgan. Biroq, amalda bunday tavsiflar uchun umumiy qabul qilingan tuzilma paydo bo'ldi. Agar tavsif kamida quyidagi qismlarni o'z ichiga olgan bo'lsa, sun'iy til to'g'ri tasvirlangan bo'ladi:

- sintaksis, ya'ni til konstruksiyalarini tuzish qoidalarini aniqlash;
- semantika, ya'ni qoidalarning ta'rifi til tuzilmalariga ma'no berish;
- amaliyoti, ya'ni qoidalarning ta'rifi muayyan maqsadlarga erishish uchun til tuzilmalaridan foydalanish.

Rasmiy sun'iy til sifatida UML sintaksis, semantika va amaliyotga ega bo'lgani holda, dasturlash tillariga nisbatan odatdagidan boshqacha tasvirlangan va ba'zi hollarda bu qismlar boshqacha nomlangan bo'lishi mumkin, chunki, birinchidan, UML dasturlash emas, balki grafika elementlaridan foydalanuvchi modellashtirish tilidir.

Kengroq ma'noda, UML – bu tizimlarning tasnifi, namoyishi, tuzilmasi va hujjatlashtirish uchun mo'ljallangan grafik modellashtirish tilidir.

### 12.1. UML – bu modellashtirish tili

UML nomidagi "modellashtirish" so'zi mavjud belgilangan usullarga ko'ra ko'p ma'noli soyalarga ega.

Dasturiy ta'minotni ishlab chiqish nuqtayi nazaridan tahlil va loyihalash bosqichlarining natijalari, ma'lum bir til yordamida yaratilsa, odatda *model* deb ataladi. Shu kabi tabiiy tillarni rasmiylashtirishda ham maxsus yaratilgan tildan foydalanilsa, uni ham *model* deb ataymiz.

Demak, modellarni kompilyatsiya qilish faoliyatini tabiiy ravishda modellashtirish deb atash mumkin. Aynan shu ma'noda UML modellashtirish tilidir.

Shunday qilib, UML modeli, birinchi navbatda, ob'yekt yoki hodisani tavsiflovchidir.

Hurmatli UML mualliflarining sa'y-harakatlari hamda butun xalqaro dasturlash hamjamiyatining ko'magi va yordami bilan ma'lum bo'lgan dasturlash tili elementlarining ko'pini birlashtira oldilar. Agar biz UML elementlarining paydo bo'lishi va rivojlanishi tarixini ikkala darajada ham kuzatishga harakat qilsak, asosiy g'oyalari va texnik tafsilotlar darajasi hali ham rivojlanayotgani va UML tilining ham ochiq kod sifatida takomillashayotganini ko'rishimiz mumkin.

Shu o'rinda UML mualliflarining ta'rifini keltiramiz.

UML bu – tizimlarning dasturiy ta'minotini ishlab chiqish jarayonida yaratilgan barcha suniy dalillarni *tasniflash, namoyishlash, loyihalash va hujjatlashtirish* bo'yicha yaratilgan umumiy modellashtirish uchun grafik tildir [3].

### **Tasniflash**

Ilovani ishlab chiqishda odatiy holatlarda kamida ikkita aktyor ishtirok etadi: mijoz (aniq shaxs yoki odamlar guruhi yoki tashkilot) va ishlab chiquvchi (bu bitta dasturchi, vaqtinchalik loyiha jamoasi yoki dasturiy ta'minotni ishlab chiqishga ixtisoslashgan burorta tashkilot bo'lishi mumkin). Asosan, ikkita subyektdan iborat harakatlanuvchilarninig mavjudligi sababli ko'p narsa ularning o'zaro tushunish darajasiga bog'liq.

Ilovalarni ishlab chiqishning asosiy bosqichlaridan biri ishlab chiqilayotgan ilova qanday talablarga javob berishi kerakligini aniqlashdir. Ushbu bosqich natijasida rasmiy yoki norasmiy sun'iy dalillardan iborat hujjat paydo bo'ladi, u boshqacha nomlanadi, taxminan bir xil narsani anglatadi – muammoning qo'yilishi, talablar, texnik topshiriqlar, tashqi tasniflar va boshqalar.

**Tasnif** – standartlar, sifat mezonlari va boshqalarni hisobga olgan holda algoritm, qoidalar, ob'yektlar harakatlarining qanday yaratilgani yoki ishlashining deklarativ tavsifidir.

Bunda tasnifning uchta talqinini hisobga olish kerak:

1. Harakatlanuvchi shaxs (aktyor) nazarda tutgan narsa, tasnif manbasi (masalan, mijoz) bo'lishini.
2. Harakatlanuvchi shaxs nazarda tutgan narsa, tasnif iste'molchisi (masalan, yaratuvchi) bo'lishini.
3. Tabiat tomonidan ob'yektiv ravishda belgilanadigan narsa belgilangan ob'yekt ekanligini.

Xususiyatlarning ushbu uchta talqini juda ko'p holatlarda birbiriga mos kelmasligi mumkin.

Shu nuqtayi nazardan, UMLning bir tomondan yetarli darajada rasmiy, ikkinchidan yetarli darajada qulay va uchinchi tomondan

tasniflardagi turlicha qarashlar orasidagi tafovutlarni qaysidir darajaga kamaytiruvchi universal vositani taqdim etishdan iborat.

### **Namoyishlash**

Inson idrokiyatining o'ziga xos xususiyatlari shundan iboratki, rasmiy matnni yolg'iz matnga qaraganda osonroq idrok etish. Matnli rasmlar esa yanada osonroq. UML modellari rasmlar ko'rinishida taqdim etishga imkon beradi va bu rasmlar namoyishli, intuitiv, deyarli bir ma'noda talqin qilinishi mumkin va oson tuzilgan. Darhaqiqat, ushbu tezisning rivojlanishi va tafsiloti bo'yicha juda ko' marta to'xtab o'tamiz.

### **Loyihalalash**

Asl nusxada ushbu UML maqsadi tuzilma so'zi yordamida aniqlanadi, biz uni diqqat bilan "loyihalash" atamasi bilan yetkazamiz. Gap shundaki, UML nafaqat ilovalarning mavhum modellarni tavsiflash, balki ushbu ilovalarni tashkil etuvchi artefaktlarni, jumladan, dastur kodi kabilarni bevosita boshqarish uchun mo'ljallangan.

Boshqacha qilib aytganda, UML maqsadlaridan biri, masalan, tegishli ilovalarning dastur kodini (yoki kod qismlarini) avtomatik ravishda yaratish mumkin bo'lgan modellarni yaratishdir. Bundan tashqari, UML modellarining tabiati shundan iboratki, teskari jarayon ham mumkin, ya'ni tugallangan dastur kodidan foydalangan holda modelni avtomatik ravishda qayta qurish. Avtomatik (yoki kompyuter yordamida) loyihalash va ilovalarni tasniflarga muvofiq loyihalash qiyin, ammo umidsiz emas. UML ni qo'llab-quvvatlaydigan vositalar doimo takomillashib bormoqda, shuning uchun kelajakda UML ning bu maqsadi birinchi o'ringa chiqishi mumkin.

### **Hujjatlar**

UML modellari elektron hujjat yoki qog'oz nusxada saqlanishi va ishlatilishi mumkin bo'lgan artefaktlardan iborat. UML ning so'nggi naqllarida ushbu maqsadga ko'proq moslashish uchun juda

ko'p ishlar qilindi. Xususan, UML modellarining XMI formatidagi hujjatlar ko'rinishida taqdim etilishi tasniflashgan, bu modellar bilan ishlashda amaliy o'zaro muvofiqlikni ta'minlaydi. Boshqacha qilib aytadigan bo'lsak, UML modellari rasmlarni chop etishdan tortib avtomatik ravishda odam o'qiy oladigan matn tavsiflarini yaratishgacha bo'lgan turli xil usullarda qo'llanilishi mumkin bo'lgan hujjatlardir.

Standart modelning ichki ko'rinishida har bir modellashtirish elementi uchun ushbu elementning norasmiy matn tavsifi saqlanishi mumkin bo'lgan joy ajratilishini talab qiladi. Aksariyat vositalar ushbu talabni bajaradi: *diagrammadagi tom ma'noda har bir satr yoki rasm uchun siz ushbu aniq chiziq yoki rasmining ma'nosi va maqsadini tushuntiruvchi matnni kiritishingiz mumkin*. Bundan tashqari, ko'plab vositalar ushbu matn tavsiflaridan foydalanishi mumkin. To'liq mazmunli va yaxshi formatlashgan bunday matnli tavsiflardan modellashtirilayotgan tizimda foydalanish mumkin.

## 11.2. Model va uning elementlari

UML modeli – bu cheklangan to'plamdagi til tuzilmalari bo'lib, uning asosiylari mohiyatlar va ular o'rtasidagi munosabatlar.

Modeldagi mohiyatlar va munosabatlarning o'zi esa, meta-modeldagi metasinflarning nusxalaridan iborat.

### Mohiyatlar

Ko'rib chiqish qulayligi uchun UMLdagi mohiyatlarni to'rt guruhga bo'lish mumkin:

- tuzilmaviy;
- xulqiy;
- guruhli;
- izohli.

Tuzilmaviy tuzilishlar tuzilishni tavsiflash uchun mo'ljallangan. Odatda, Tuzilmaviy tuzilishlar quyidagilarni o'z ichiga oladi:

**Ob'yekt (object)** – bu noyoblik xossasiga ega bo'lgan va atrofidagi begona holat va xatti-harakatni qamrab olgan mohiyatdir.

**Sinf (class)** – bu ob'yektlar to'plamining umumiy holatini belgilovchi atributlar va xususiyatini aniqlovchi amallar tavsifidir.

**Interfeys (interface)** – bu nomlangan amallar termasi bo'lib, iste'molchi tomonidan so'ralishi mumkin bo'lgan xizmatlar termasini va xizmat ko'rsatuvchi provayder tomonidan shu xizmatlarni taqdim etilishini aniqlovchi.

**Hamkorlik (collaboration)** – bu biron bir maqsadga erishish uchun o'zaro harakatlanuvchi ob'yektlar to'plami.

**Aktor (harakatlanuvchi shaxs)** – bu tashqarida joylashgan mohiyat bo'lib, u modellashtiriladigan tizimdan tashqarida va u bilan bevosita o'zaro harakat qiladi.

**Tarkib (komponent)** – bu tizimning modulli qismi bo'lib, talab qilinuvchi va taqdim etiluvchi interfeyslarning aniq termasidan iborat.

**Artefakt (artifact)** – bu dasturiy ta'minotni ishlab chiqish bo'yicha mutaxassislar faoliyatining har qanday mahsuloti. Boshqacha qilib aytganda, artefakt – bu model elementi (masalan, sinf yoki tarkib)ning jismoniy birligidan iborat amalga oshirish birligi.

**Tugun (node)** – bu artefaktlar joylashtiriladigan va kerak bo'lganda bajariladigan hisoblash zaxirasi.

Xulq-atvor mohiyatlari tavsiflash uchun mo'ljallangan xulq-atvor. Faqat ikkita asosiy xulq-atvor mohiyatlari mavjud - holat va harakat. Ayrim vaziyatlarda, holatning alohida holati sifatida qaraluvchi faoliyat ham qo'llanilishi mumkin.

**Holat (state)** – bu ob'yektning yashash davri bo'lib, bunda ob'yekt ma'lum bir shartni qondirishi va o'z faoliyatini amalga oshirishi yoki biron bir hodisaning sodir bo'lishini kutadigan mavjudlik vaqti.

**Faoliyat (activity)** holatning alohidagi xususiy voqeligi, deb hisoblash mumkin. Uzoq vaqt davomida, atomar bo'lmagan hisob-kitoblar bilan tavsiflanish holati.

**Harakat (action)** – bu eng sodda atomar hisobdir. Bu xulq-atvor mohiyatlar aysbergining faqat uchi, chunki holatlar juda farq qiladi. Bundan tashqari, xatti-harakatni modellashtirishda bir qator yordamchi mohiyatlar qo'llaniladi, chunki ular faqat ko'rsatilgan asosiy mohiyatlari bilan birga mavjud.

Biroz ajralib turuvchi mohiyat, bu – foydalanish varianti bo'lib, unda ham tarkibiy va ham xulq-atvor bilan jihatlari mavjud.

**Foydalanish holati (use case)** – ko'p ssenariylar, qandaydir mezon va tavsifga ko'ra birlashtiriladigan va tizim tomonidan bajariladigan harakatlar ketma-ketligi bo'lib, ba'zi aktyorlar uchun muhim bo'lgan natijani taqdim etadi.

Yuqorida keltirilgan mohiyatlar sinfi to'liq emas. Ushbu mohiyatlarning har biri turli xil maxsus holatlar va o'zgarishlarga ega bo'lishi mumkin.

UMLda faqat bitta guruhlash mohiyati, paket mavjud, lekin universal.

**Paket (package)** – modeldagi elementlar guruhi (shu jumladan paketlar).

Izohli mohiyat ham yagona – *eslatma (comment)*, lekin unga hamma narsani qo'yish mumkin, chunki eslatmaning mazmuni UML bilan cheklanmagan.

11.1- jadvalda, keltirilgan mohiyatlarning minimal tarzda variantining standart notatsiyasi (belgilanishi) keltirilgan.

### 11.3. Munosabatlar

UMLda munosabatlarning asosiy to'rtta turi qo'llaniladi:

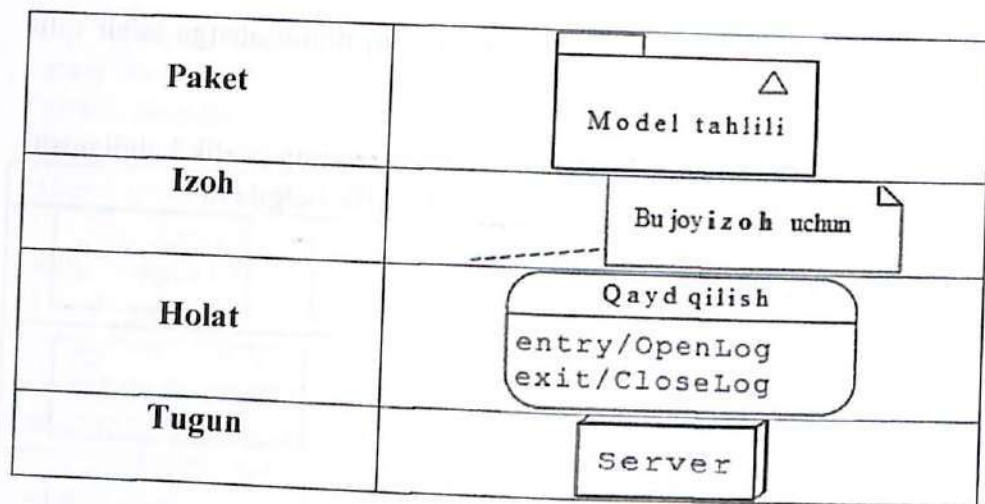
- bog'liqlik (dependency);
- uyushma (association);
- umumlashtirish (generalization);
- amalga oshirish (realization).

**Bog'liqlik** ikkita shaxs o'rtasidagi munosabatlarning eng keng tarqalgan turidir. Bog'liqlik munosabati, mustaqil mohiyatning

o'zgartirishdagi qaysidir ma'noda, bog'liq munosabatga ta'sir qilishini ko'rsatadi.

11.1- jadval. Asosiy mohiyatlarning grafik belgilanishi

Nomi	Grafik belgilash	
Artefakt	«artifact» Talab tasnifi	«library» QT
Foydalaninish varianti		
Harakatlanuvchi shaxs (aktyor)		Mijoz
Faoliyat va harakat		
Interfeys	IAudio	«interface» IAudio
Sinf	Mahsulot	Buyurtma {mavhum}
Tarkib	Berilganlar bazasi	Berilganlar bazasi «Tarkibi»
Hamkorlik		
Ob'yekt	:To'rtburchak	



Grafik jihatdan, bog'liqlik munosabati tobe mohiyatdan (2) mustaqil munosabatga (3) yo'naltirilgan millik (1) uzma chiziq sifatida tasvirlanadi (11.1- rasm). Qoida tariqasida, ma'lum bir bog'liqlikning semantikasi qo'shimcha ma'lumotlar yordamida modelda aniqlashtiriladi. Masalan, "foydalanish" qattiq turi bilan bog'liqlik, tobe mohiyat mustaqil munosabatdan foydalanishini (aytaylik, amakni chaqirishini) anglatadi.

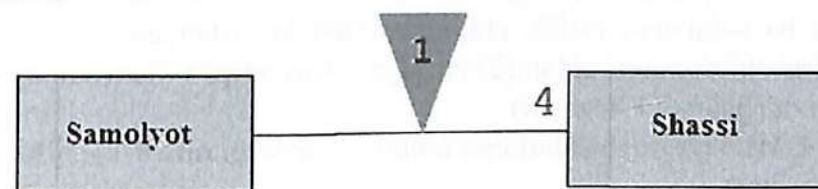


11.1- rasm. Bog'liqlik munosabati

**Yushma** bu – mohiyatlar o'rasidagi, munosabatlarning eng ko'p qo'llaniladigan turidir.

Agar bitta mohiyat boshqasiga bevosita bog'liq bo'lsa (yoki boshqalar bilan, uyushma nafaqat ikkilik bo'lishi mumkin) uyushma munosabatlari yuzaga keladi.

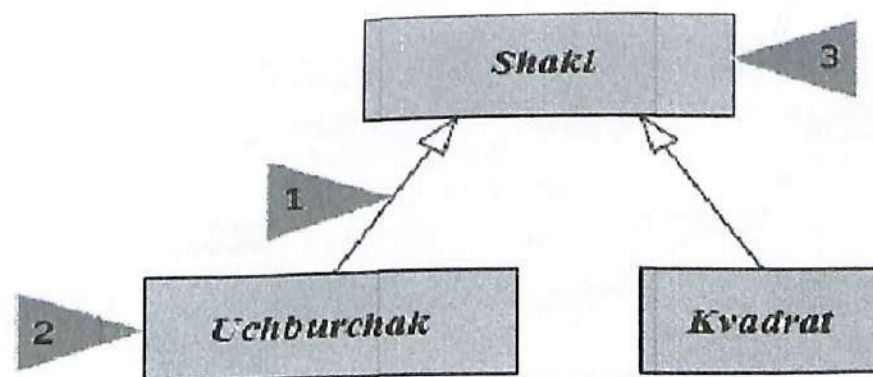
Grafik jihatdan uyushma bir-biriga bog'langan munosabatlarni bog'laydigan turli qo'shimchalar bilan uzluksiz chiziq (1) sifatida tasvirlangan (11.2- rasm). Dastur darajasida bevosita aloqa turli yo'llar bilan amalga oshirilishi mumkin, asosiysi, bog'langan munosabatlarning bir-birini bilishi. Masalan, munosabatning butun qismi, uyushmaning alohida holi bo'lib, *yig'ma munosabat* deb ataladi.



11.2- rasm. Uyushma munosabati

**Umumlashtirish** bu – ikkita mohiyatlar o'rtasidagi munosabatlardan iborat bo'lib, ulardan biri ikkinchisining xususiy (maxsus) holatidir.

Grafik jihatdan umumlashtirish uzluksiz oxirida uchburchak shaklidagi bo'yalmagan millik chiziq sifatida tasvirlangan (1), xususiy mohiyat (qism-sinf)dan (2) umumiy mohiyat (yuqori sinf)ga (3) yo'naltirilgan (11.3- rasm).



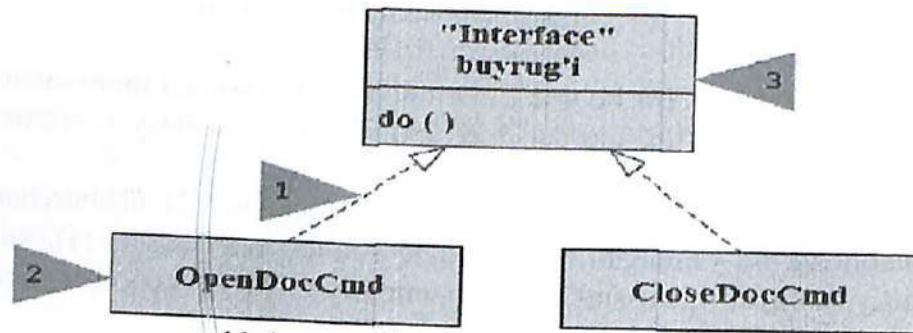
11.3- rasm. Umumlashtirish munosabati

**Amalga oshirish** munosabati, oldingi uch turdagi munosabatlarga nisbatan kamroq qo'llaniladi, chunki ular ko'pincha oshkor-mas ravishda ishlatiladi.

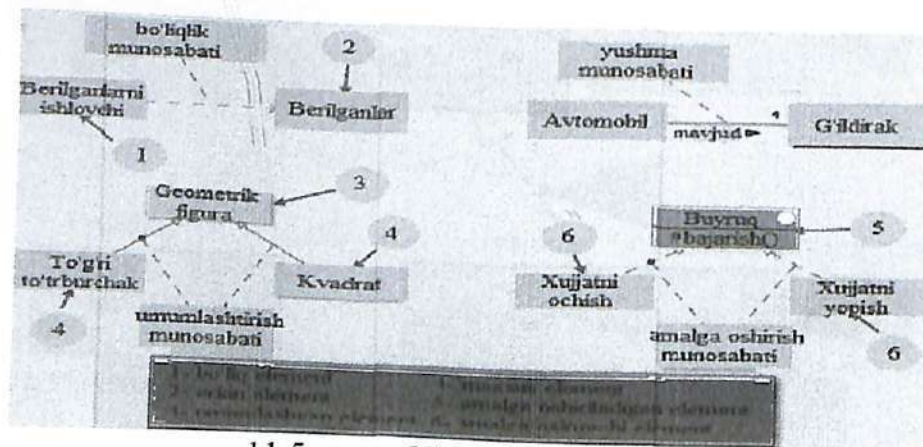
Amalga oshirish munosabati, bitta munosabat boshqasini amal oshirishning natijasi ekanligini bildiradi. Masalan sinf bu – interfeysning amalga oshirilishining natijasidir.

Grafik jihatdan amalga oshirish oxirida (1) uchburchak shaklidagi bo'yalmagan millik chiziq sifatida tasvirlangan, uni amalga oshiruvchi munosabatdan (2) amalga oshiriladigan munosabatga (3) yo'naltirilgan (11.4- rasm).

UMLdagi munosabatlarga umumiy tarzda misol 11.5- rasmda keltirilgan.



11.4- rasm. Amalga oshirish munosabati



11.5- rasm. UML dagi munosabatlar

## 11.4. Diagrammalar

UML diagrammalari bu – modelni yaratish va undan foydalanishni osonlashtiradigan modelga qo'yilgan asosiy tuzilmadir.

**Diagramma** – bu sinflar, ssenariylar, holatlar va boshqalar yordamida tizimni modellashtirishning grafik tasviri.

Umuman olganda, mohiyatlar va munosabatlarning har qanday (to'g'ri) kombinatsiyasi diagrammaga kiritilishi mumkin, ammo bu masaladagi o'zbohimchalik modellarni tushunishni qiyinlashtiradi. Shuning uchun UML mualliflari foydalanish uchun tavsiya etilgan diagramma turlari to'plamini aniqladilar, ular **kanonik diagramma turlari** deb ataladi.

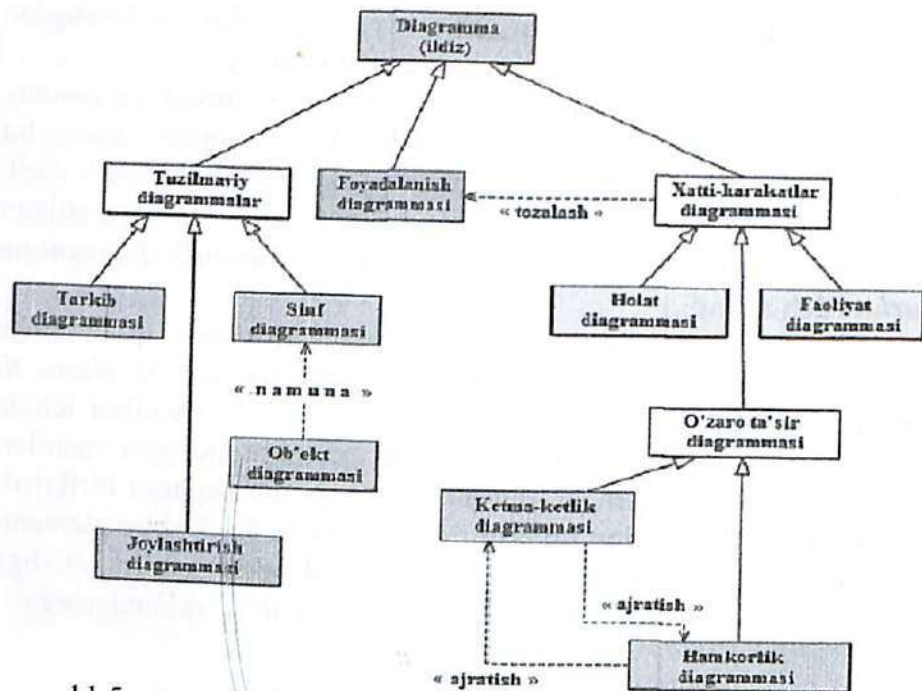
Diagrammadagi mohiyatlar va munosabatlarga qo'shimcha ravishda modelning boshqa elementlari ham borki, biz ularni **til konstruksiyalari** deb ham ataymiz. Bular ob'yekt shakllari ichida yoki aloqa chiziqlari yonida yozilishi mumkin bo'lgan matnlar, diagrammalar qoliplari va ularning qismlari, chiziq'larga birlashtirilgan yoki shakllar ichiga joylashtirilgan belgilardir. Ushbu elementlar nafaqat modelni ko'proq namoyish shaklda taqdim etishga yordam beradi, balki sezilarli darajadagi semantik yuklamaga ega.

### Diagrammalarning tasnifi

UML 1 jami 9 ta kanonik diagramma turlarini belgilaydi. Ushbu darslikda qabul qilingan ularning nomlari quyida keltirilgan (boshqa manbalarda farqlar mavjud):

- ❖ foydalanish (Use Case) diagrammasi;
- ❖ sinf (Class) diagrammasi;
- ❖ ob'yekt (Object) diagrammasi;
- ❖ holat (State chart) diagrammasi;
- ❖ faoliyat (Activity) diagrammasi;
- ❖ ketma-ketlik (Sequence) diagrammasi;
- ❖ hamkorlik (Collaboration) diagrammasi;
- ❖ tarkib (Component) diagrammasi;

❖ joylashtirish (Deployment) diagrammasi.  
Keltirilgan diagrammalarning shartli ravishdagi sinflarga ajratish natijasi quyidagi 11.6- rasmda keltirilgan.



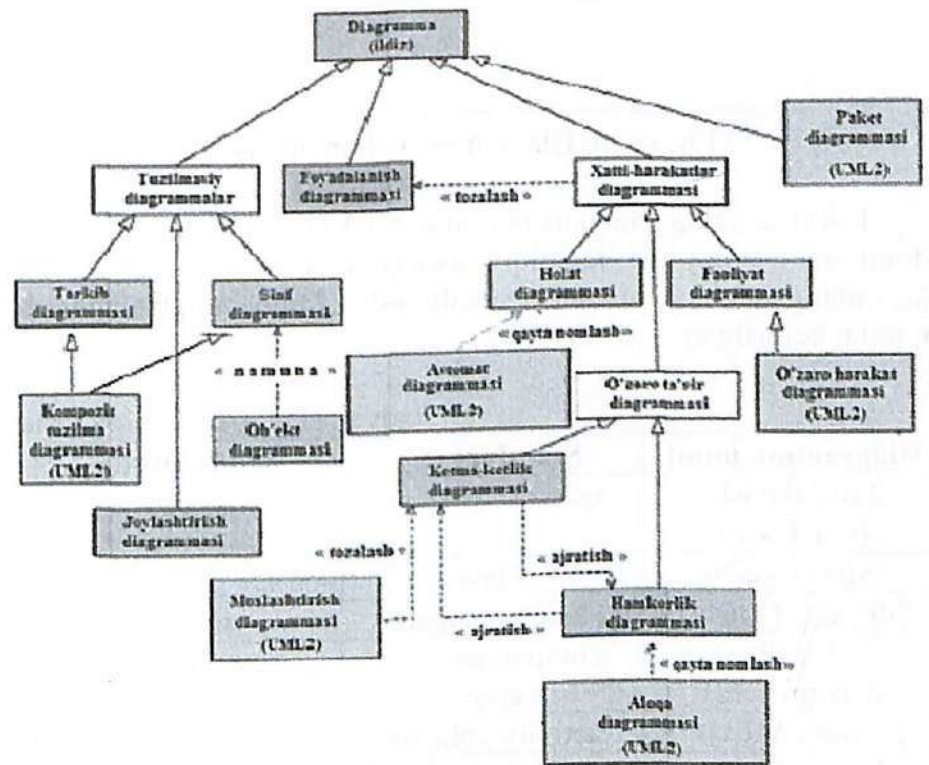
11.5- rasm. UML 1 dagi diagrammalar turlarining shajarasi

UML 2 da kanonik diagrammalar soni 13 taga etdi (11.7- rasm) va mavjud til konstruksiyalari ro'yxati uni qo'llash doirasini sezilarli darajada kengaytirilgan. Bundan tashqari, 2ta diagrammaning nomi ham o'zgartirildi – hamkorlik diagrammasi *aloqa diagrammasiga* va holat diagrammasi *avtomat diagrammasiga* o'zgartirilgan.

Yangi qabul qilingan va nomi o'zgartirilgan diagrammalar ro'yxati:

- ❖ kompozit tuzilma (Composite Structure) diagrammasi;
- ❖ paket (Package) diagrammasi;

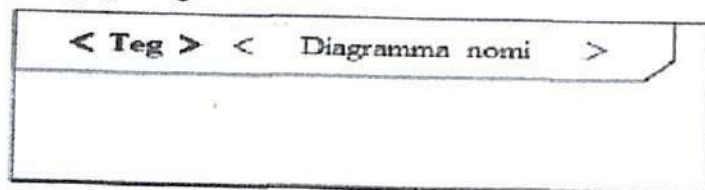
- ❖ o'zaro harakat (Interaction Overview) diagrammasi;
- ❖ moslashtirish (Timing diagram) diagrammasi;
- ❖ avtomat (State machine) diagrammasi;
- ❖ aloqa (Communication) diagrammasi.



11.7- rasm. UML 2 dagi diagrammalar turlarining shajarasi

Diagrammani tayyorlash standart talabi bo'yicha tayyorlanishi kerak. Tayyorlashda asosiy 2ta element ishlatiladi – biri sohani chegaralovchi tashqi to'rtburchak qolip bo'lsa, diagrammaning nomi yozilgan yorliq bo'lishi kerak. Bu maxsus formatda 11.8- rasmdagi kabi tayyorlanadi.

Diagrammalar uchun mumkin bo'lgan teglar 11.2- jadvalda ko'rsatilgan. Bunda, standart tomonidan tavsiya etilgan teglar ikkinchi ustunga yozilgan.



11.8- rasm. Diagramma uchun notatsiya

Lekin, amalda, standart tomonidan taklif qilingan qoidalar har doim ham qulay va mantiqiy asosga ega bo'lmashligi sababli, jadvalning uchinchi ustunida amaliyotdan kelib chiqqan muqobil nomlar keltirilgan.

Jadval 11.2. Diagramma turlari

Diagramma nomi	Standart teg	Muqobil teg
Foydalanish (Use Case)	use case yoki uc	use case
Sinf (Class)	Class	class
Ob'yekt (Object)	aniqlanmagan	object
Tarkib (Component)	component yoki cmp	component
Faoliyat (Activity)	activity yoki act	activity
Ketma-ketlik (Sequence)	interaction yoki sd	sd
Joylashtirish (Deployment)	aniqlanmagan	deployment
Moslashtirish (Timing)	interaction yoki sd	timing
Avtomat (State machine)	state machine yoki stm	state machine
Aloqa	interaction yoki sd	comm

(Communication)		
O'zaro harakat (Interaction Overview)	interaction yoki sd	interaction
Paket (Package)	package yoki pkg	package
Kompozit tuzilma (Composite Structure)	class	class yoki component

### 11.5. Umumiy diagrammalar

UMLdagi diagrammalarni shartli ravishda ikki guruhga ajratishadi. Birinchisi umumiy guruh bo'lib, bu modellashtirish sohasiga bog'lanmagan diagrammalardan iborat. Ularni sohaga bog'lanmagan ixtiyoriy dasturiy loyihalar yechimlar sohasi va boshqalar uchun ishlatish mumkin. Bu sinfga – foydalanish (use case), sinf (class), avtomat (state machine), faoliyat (activity), ketma-ketlik (sequence), aloqa (communication), tarkib (component), joylashtirish (Deployment) diagrammasi.

Ikkinchi diagrammalar sinfi maxsus tavsifga ega bo'lib, ko'pincha qaysidir umumiy diagrammani to'ldirish uchun yoki uning hususiy holati sifatida, yoki uning ayrim qismlarini aniqlovchi yordamchi diagramma sifatida ishlatiladi. Bular – ob'yektlar, kompozit, o'zaro ta'sir, moslashish va paket diagrammalaridan iborat.

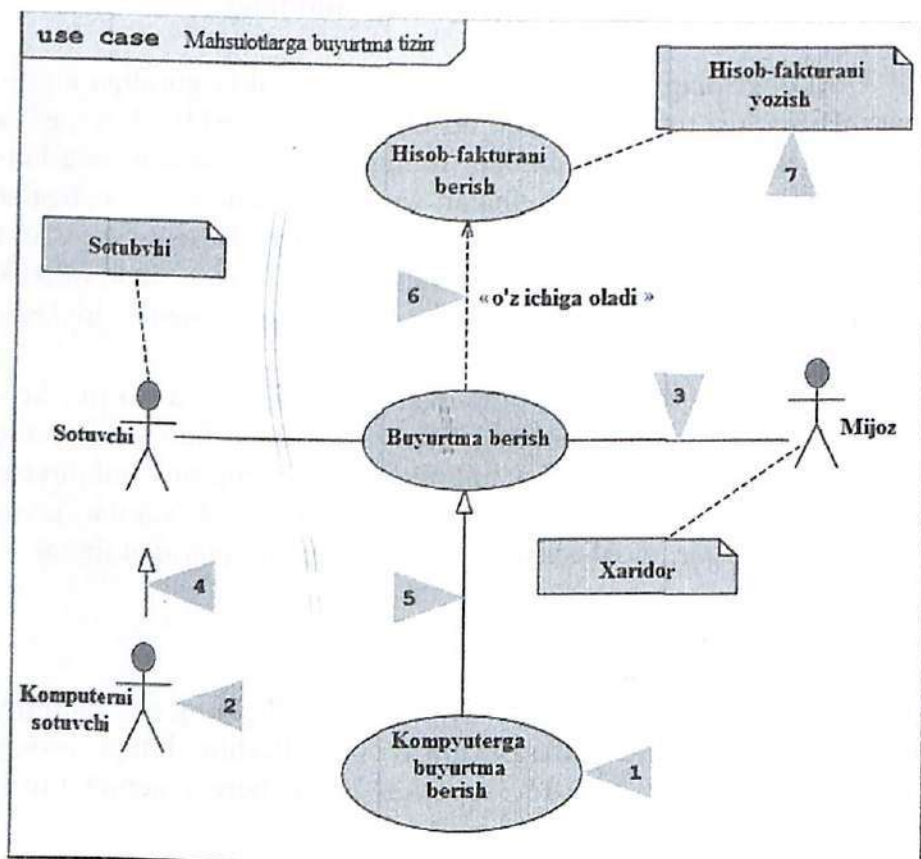
#### Foydalanish diagrammasi

Foydalanish diagrammasi tizim funksionalligining eng umumiy ifodasidir. Foydalanish diagrammasi modellashtirishdagi asosiy "Bu tizim nima ish qiladi?" savoliga javob berish uchun mo'ljallangan.

Foydalanish diagrammasi ikki turdagi asosiy ob'yektlardan foydalanadi: foydalanish holatlari (1) va aktyorlar (2), ular o'rasida munosabatlar quyidagi asosiy turlari o'rnatiladi:

- aktyor va foydalanish varianti o'rasidagi bog'liqlik (uyushma)dan foydalanish (3);
- aktyorlar munosabatlarni umumlashtirish (4);
- foydalanish variantlarini umumlashtirish (5);
- foydalanish holatlari o'rtasidagi bog'liqliklarni umumlashtirish (6).

Foydalanish diagrammasida, boshqalarida bo'lgani kabi, izohlardan foydalanish mumkin. Diagrammada ishlatiladigan yozuvlarning asosiy elementlari 11.9- rasmda ko'rsatilgan.



11.9- rasm. Foydalanish diagrammasi

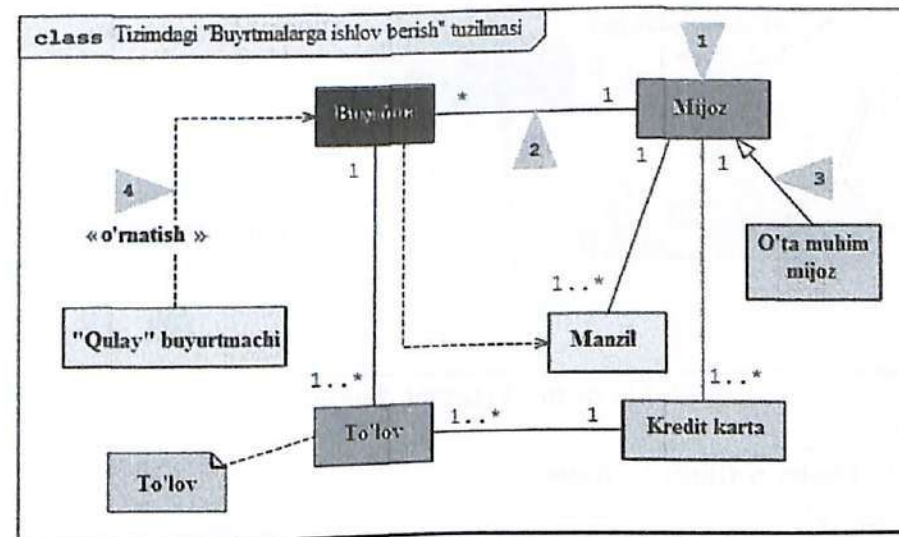
## Sinf diagrammasi

Sinf diagrammasi tizim tuzilishini tasvirlashdagi asosiy usul hisoblanadi. UML birinchi navbatda ob'ektga yo'naltirilgan til bo'lib, sinflar esa asosiy "yaratish materiali"dir.

Sinf diagrammasida ob'ektning asosiy mohiyatidan foydalanadi: sinflar (1), shu jumladan sinflarning ko'plab maxsus xususiy holatlari: aloqalar, sodda turlar, yushma sinflari va boshqalar. Sinf diagrammasida ular orasidagi munosabatlarning asosiy quyidagi turlari o'rnatiladi:

- sinflar o'rasidagi yushmalar (2) (ko'pgina qo'shimcha ma'lumotlar bilan);
- sinflarni umumlashtirish (3);
- sinflar hamda sinflar va aloqalar o'rasidagi (4) har xil turdagi bog'liqliklar.

Sinf diagrammasida ishlatiladigan notalarning ba'zi elementlari 11.10- rasmda ko'rsatilgan.



11.10- rasm. Sinf diagrammasi

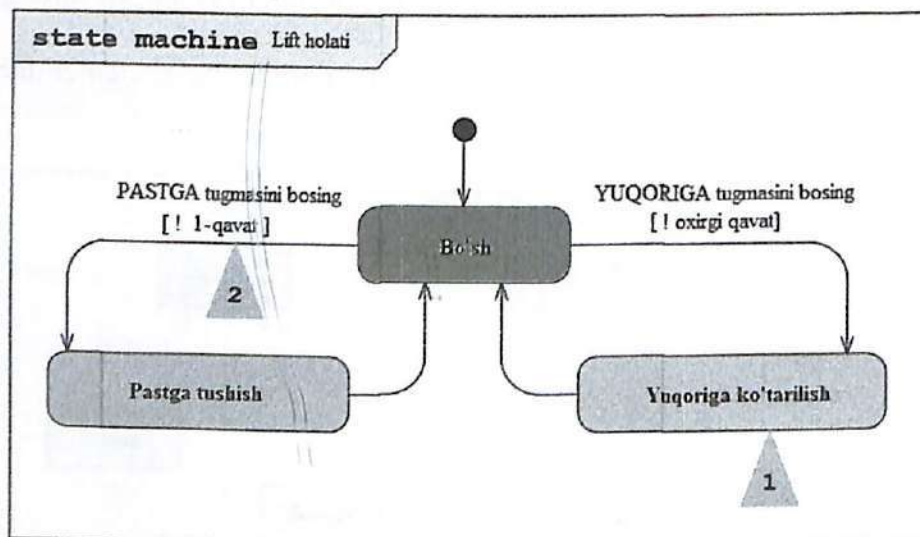
## Avtomat (holat) diagrammasi

Avtomat diagrammasi yoki UML 1 dagi holat diagrammasi – bu UML tilida holatni har tomonlama tavsiflash usullaridan biridir.

Avtomat diagrammasining mohiyati o'z nomidan kelib chiqishiga nisbatan, holatlar grafi va chekli avtomatdagi o'tishlardan iborat bo'lib, qo'shimcha qismchalar va tavsilotlar bilan yuklangan.

Avtomat diagrammasida mohiyatlarning bitta asosiy turi holatlar (1) va munosabatlarning bitta turi o'tishlardan (2) iborat, lekin ikkalasi uchun maxsus vaziyatlar va qo'shimcha belgilar to'plami aniqlangan.

11.11- rasmda avtomat diagrammasidagi belgilashlar keltirilgan.



11.11- rasm. Avtomat diagrammasi

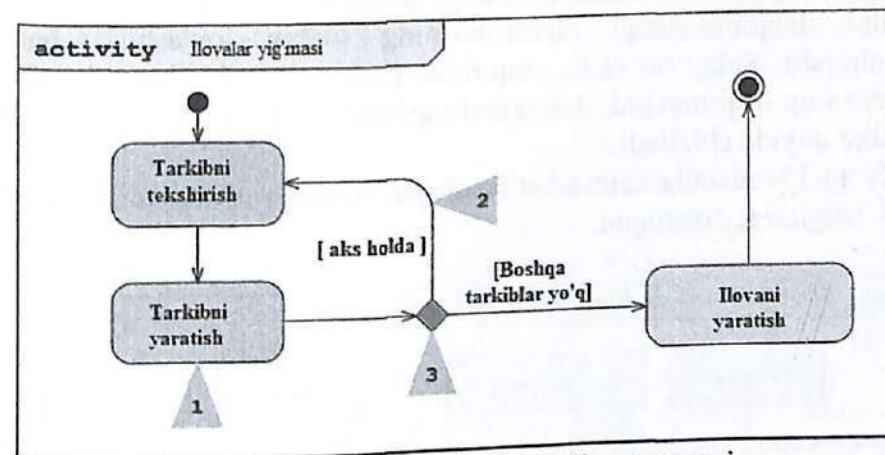
## Faoliyat diagrammasi

Faoliyat diagrammasi bu – xatti-harakatni tasvirlashning yana bir usuli bo'lib, u algoritmning blokli chizmalar ravishdagi namoyi-

shni eslatadi. Biroq, takomillashtirilgan belgilar tufayli, ob'yektga yo'naltirilgan yondashuvga mos keladigan UML faoliyat diagrammasi tizimning xatti-harakatlarini tavsiflash uchun kuchli vositadir.

Faoliyat diagrammasida mohiyat (ob'yekt)larning bitta asosiy turi - harakat (1) va munosabatlarning bitta turi - o'tishlar (nazoratni uzatish) (2) ishlatiladi. Shuningdek, ob'yektlarga o'xshash ayrilishlar, birlashishlar, qo'shilishlar, bog'lanishlar, tarmorlar (3) kabi tuzilmalar ham qo'llaniladi, lekin aslida ular bunday emas, balki gipergrafdagi hiper yoylarning ayrim maxsus holatlarini tasvirlashning grafik usuli hisoblanadi.

Faoliyat diagrammasida ishlatiladigan belgilashlarning asosiy elementlari 11.11- rasmda ko'rsatilgan.



11.11- rasm. Faoliyat diagrammasi

## Ketma-ketlik diagrammasi

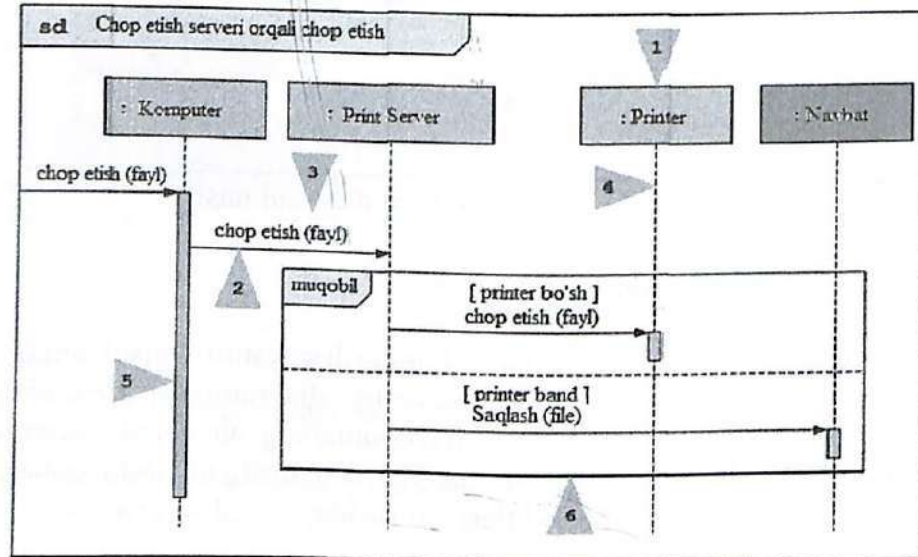
Ketma-ketlik diagrammasi tizimning harakatini "misol orqali" tasvirlash usulidir. Aslida, ketma-ketlik diagrammasi tizimning ma'lum bir bo'lagi yoki bunday qaydnomaning bir qismi qaydnomaning matni (yozuvi)dir. Ob'yektga yo'naltirilgan dasturlashda, eng muhim narsa ob'yektlari o'rasidagi xabarlarini o'zaro

uzatishdan iborat. Bu diagrammada aynan xabarlarni yuborish ketma-ketligi ko'rsatiladi, shuning uchun ham diagrammaning nomi, shu jarayonni ifodalaydi.

Ketma-ketlik diagrammasida, mohiyatning bitta asosiy turidan - o'zaro harakatlanuvchi sinflovchilar (1) (asosan sinflar, tarkiblar va aktyorlar) va munosabatlarning bitta turidan, ya'ni xabarlar (3) almashinadigan aloqalardan (2) foydalaniladi. Xabarlarni yuborishning bir necha usullari mavjud bo'lib, ular grafik belgilarda munosabatga mos keladigan o'q turi bilan farqlanadi.

Ketma-ketlik diagrammasining muhim jihati, vaqt o'tishini aniq ko'rsatishdan iborat. Moslashtirish diagrammasidan tashqari boshqa diagramma turlaridan farqli o'laroq, ketma-ketlik diagrammasida nafaqat elementlar orasidagi grafik aloqalarning mavjudligi, balki diagrammadagi elementlarning nisbiy joylashuvi ham muhimdir. Sukut bo'yicha yuqoridan pastga yo'naltirilgan (ko'rinmas) vaqt o'qi mavjud, deb taxmin qilinadi va keyinroq yuborilgan xabar quyida chiziladi.

11.13- rasmda ketma-ketlik diagrammasida ishlatiladigan asosiy belgilar ko'rsatilgan.



### 11.13- rasm. Ketma-ketlik diagrammasi

O'zaro haqakatlanuvchi ob'yektlarni belgilash uchun standart belgi qo'llaniladi, ya'ni nomli sinflovchi nusxasi yozilgan to'rtburchak. Undan chiqadigan nuqtali chiziq *hayot chizig'i*, deb ataladi (4). Bu modeldagi munosabatlarning ifodasi emas, balki diagrammani o'quvchi inson uchun to'g'ri yo'naltirish uchun mo'ljallangan grafik izoh. Hayot chizig'iga o'rnatilgan tor chiziqlar ko'rinishidagi qiyofalar ham modellashtiriluvchi mohiyatlarning tasvirlari emas.

Bu ob'yektning boshqaruv oqimiga egalik qilish (bajarish sodir bo'lishi) (5) yoki boshqacha aytganda, ob'yektning faollashishi sodir bo'lgan vaqt davrlarini ko'rsatadigan grafik izoh. O'zaro harakat qadamlari (6) ketma-ketlik diagrammasida o'zaro harakat qaydnomsining algoritmik jihatlarini aks ettirishga imkon beradi.

### Aloqa (hamkorlik) diagrammasi

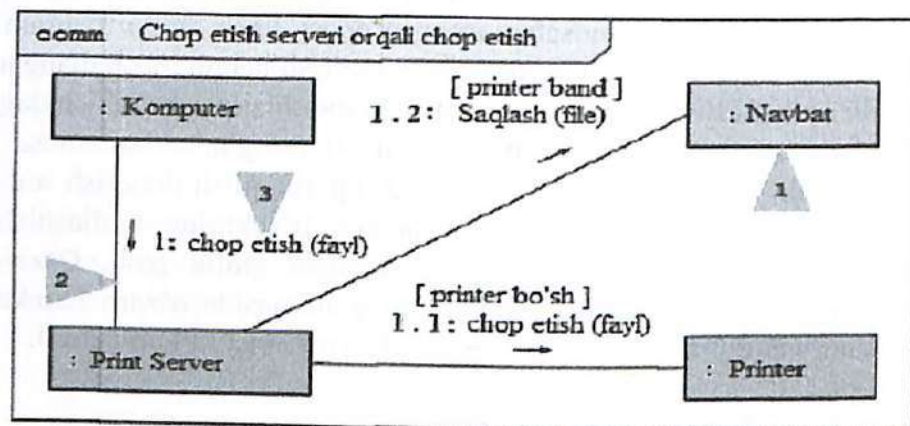
Aloqa diagrammasi – semantik jihatdan ketma-ketliklar diagrammasiga ekvivalent bo'lgan xulq-atvor tavsiflaridan iborat usul. Aslida, bu ham axborotlarni almashish ketma-ketligining xuddi shunday o'zaro harakatlanuvchi sinflovchi nusxalaridan iborat bo'lib, faqat boshqa grafik vositalar bilan ifodalangan. Shunday qilib, aloqa diagrammasida ketma-ketlik diagrammasidagi kabi mohiyatning bitta asosiy turi – o'zaro harakatlanuvchi sinflovchi nusxalaridan (1) va munosabatlarning bitta turi – aloqalar (2) ishlatiladi.

Biroq, bu yerda asosiy e'tibor vaqtga emas, balki aniq nusxalar orasidagi aloqalarning tuzilishiga qaratiladi. 11.13- rasmda aloqa diagrammasida ishlatiladigan asosiy yozuv elementlari keltirilgan.

O'zaro harakatlanuvchi ob'yektlarni belgilash uchun standart belgi - sinflovchi nusxasini nomi bilan to'rtburchak qo'llaniladi.

Hamkorlik diagrammasidagi elementlarning nisbiy joylashishi muhim emas - faqat axborotlar uzatiladigan aloqalar (ko'pincha uyushma nusxalari holatlari) muhim ahamiyatga ega (3). Shajaraviy

o'qli raqamlash xabarlarining vaqt bo'yicha tartibini ko'rsatish uchun ishlatiladi. 11.13- rasm bilan 11.14- rasmni solishtiring (ularda bir xildagi xatti-harakatni tasvirlangan) va sizga hamma narsa aniq bo'ladi.



11.14- rasm. Aloqa diagrammasi

### Joylashtirish diagrammasi

Joylashtirish diagrammasi tizim elementlarining tarkibi va ulanishlarini ko'rsatish bilan birga, ularning ishlashi vaqtidagi jismoniy jihatdan hisoblash zaxiralarda qanday joylashganligini ko'rsatadi.

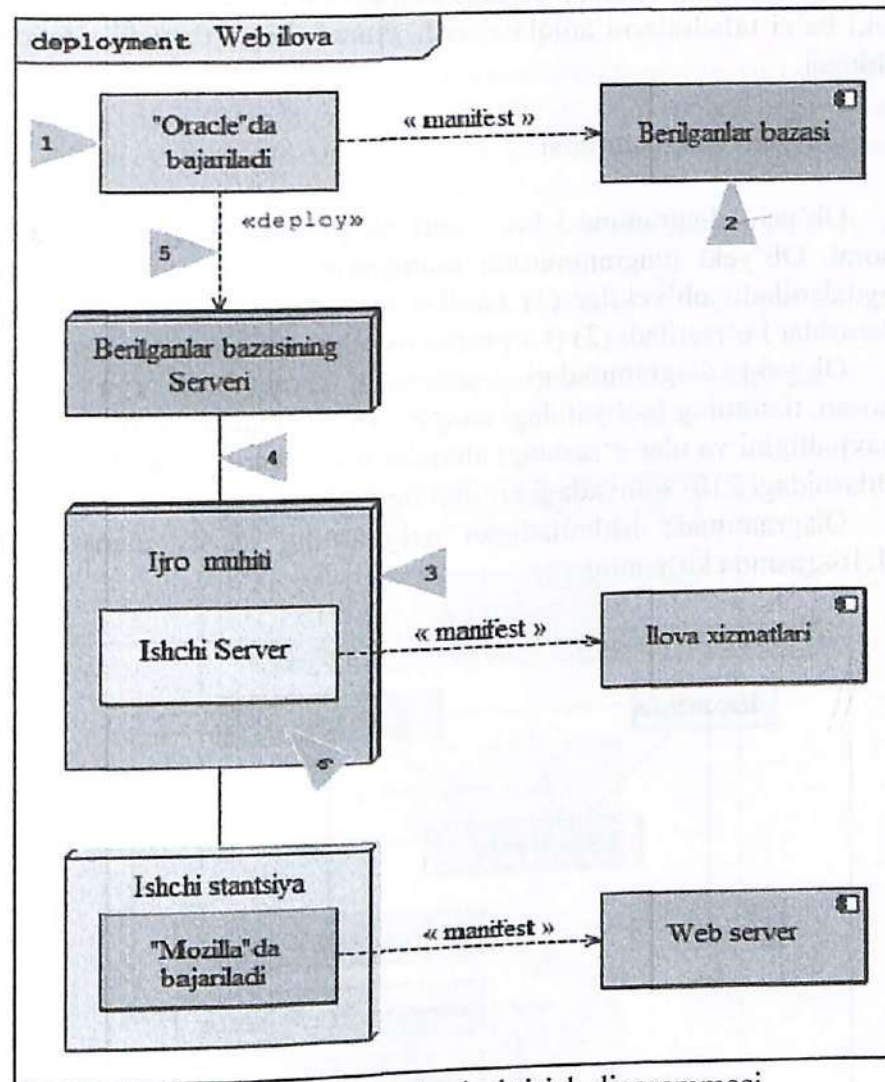
Joylashtirish diagrammasida, tarkiblar diagrammasiga nisbatan, ikki turdagi mohiyatlar qo'shiladi:

- ❖ tarkibning (2) va amalga oshirilishi natijasida paydo bo'lgan artefakt (1),

- ❖ va tugun (3), bu tugun turini yoki ma'lum bir nusxani tavsiflovchi sinflovchi bo'lishi mumkin va tugunlar (4) o'rasidagi uyushma munosabati, bu tugunlarning ish vaqtida jismoniy bog'langanligini ko'rsatadi.

11.15- rasmda joylashish diagrammasida ishlatiladigan belgilarning asosiy elementlari ko'rsatilgan. Bir mohiyat boshqa

mohiyatning bir qismi ekanligini ko'rsatish uchun «o'rnatish» (5) bog'liqlik munosabati qo'llaniladi yoki bir mohiyatning qiyofasi boshqa mohiyatning (6) ichiga joylashtiriladi.



11.15- rasm. Joylashtirish diagrammasi

## 11.6. Maxsus diagrammalar

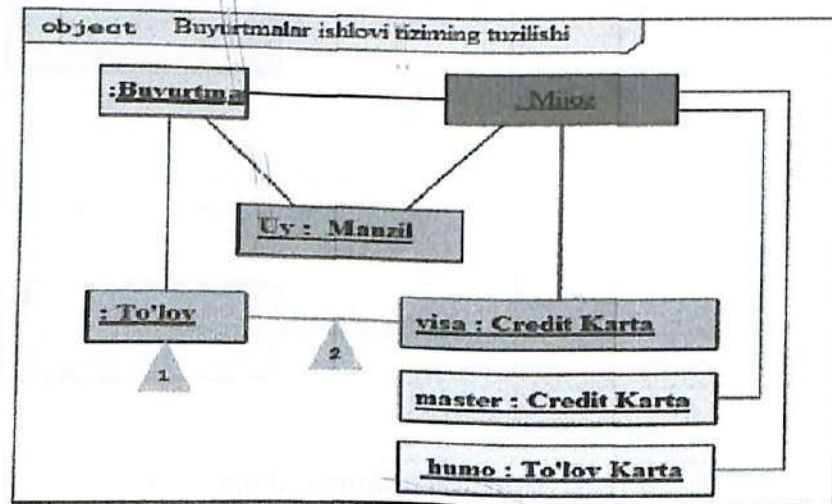
Maxsus diagrammalar ko'pincha umumiy diagrammani to'ldirish uchun xizmat qiladi, masalan, diagrammaning xususiy holatimi yoki ba'zi tafsilotlarni aniqlashtirish shunchaki yordamchi sifatida ishlaydi.

### Ob'yekt diagrammasi

Ob'yekt diagrammasi bu – sinf diagrammasining nusxasidan iborat. Ob'yekt diagrammasida mohiyatning bitta asosiy turidan foydalaniladi: ob'yektlar (1) (sinf nusalari), ular orasida aniq ulanishlar ko'rsatiladi (2) (ko'pincha uyushmalarning nusalari).

Ob'yekt diagrammalari yordamchi xaraktyerga ega – by asosan, tizimning faoliyatidagi aniq bir vaqtda qandaydir ob'yektlar mavjudligini va ular o'rasidagi aloqalarni ko'rsatadi (masalan tizim ishlashidagi 210- soniyadagi xotiraning taqsimlanishi).

Diagrammada ishlatiladigan belgilarning asosiy elementlari 11.16- rasmda ko'rsatilgan.



11.16- rasm. Ob'yekt diagrammasi

## Tarkib diagrammasi

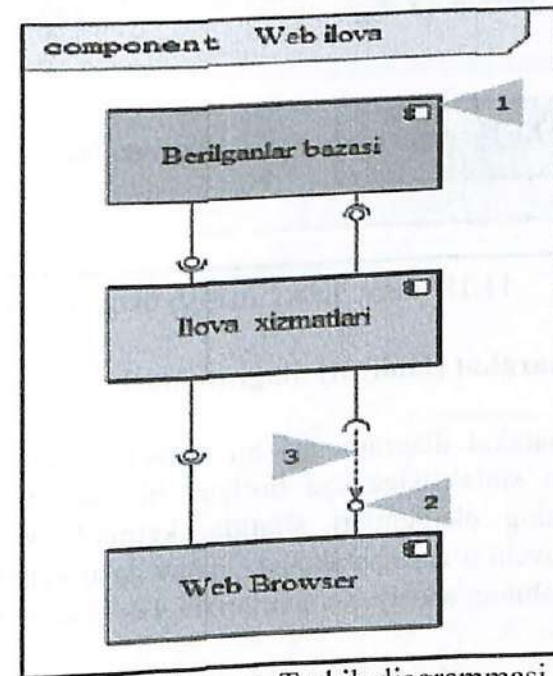
Tarkib diagrammasi – modellashtirilayotgan tizimni tashkil etuvchi sinflar va tarkiblarni, mantiqiy yoki fizik modullar o'rasidagi munosabatlarni ko'rsatadi.

Tarkib diagrammasidagi mohiyatlarning asosiy turi tarkiblarning o'zlari (1), shuningdek interfeyslar (2) bo'lib, ular orqali tarkiblar o'rasidagi o'zaro aloqalar ko'rsatiladi. Tarkiblar diagrammasida quyidagi munosabatlari ishlatiladi:

- tarkiblar va interfeyslar orasidagi amalga oshirishlar (tarkib interfeysni amalga oshiradi);

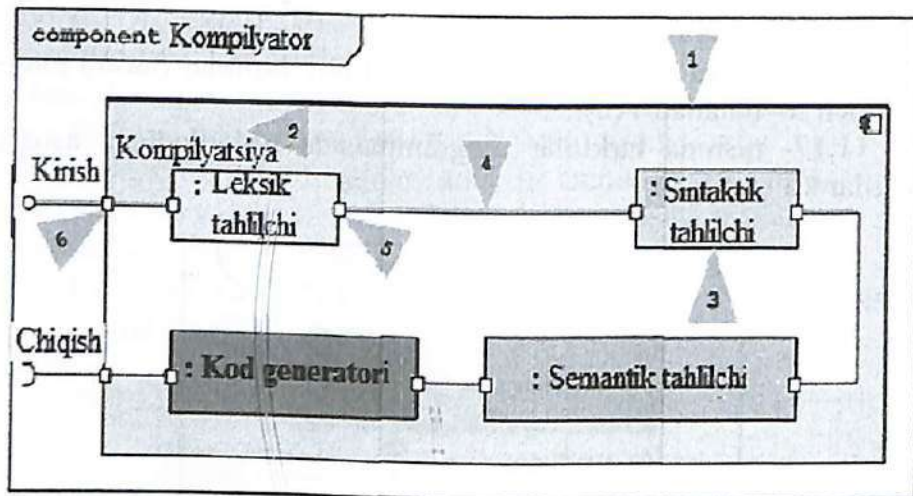
- tarkiblar va interfeyslar o'rtasidagi bog'liqliklar (tarkib interfeysidan foydalanadi) (3).

11.17- rasmda tarkiblar diagrammasida ishlatiladigan asosiy belgilar ko'rsatilgan.



11.17- rasm. Tarkib diagrammasi

To'rtburchak (1) sifatida tasvirlangan tuzilmaviy sinflovchi-ning nomi, uning tepa qismida yoziladi (2). Ichida bitta yoki bir nechta qismlar (3) mavjud bo'lishi mumkin. Qismlar bir-biri bilan o'zaro harakatlanishi mumkin. Bu har xil turdagi *ulagichlar* (4) bilan ko'rsatiladi. Ulagich biriktirilgan qismning tashqi chegarasidagi joy *port* (5), deb ataladi. Portlar tuzilmaviy sinflovchining (6) tashqi chegarasida ham joylashishi mumkin. Ichki tuzilish diagrammasida ishlatiladigan yozuvning asosiy elementlari 11.18-rasmda ko'rsatilgan.

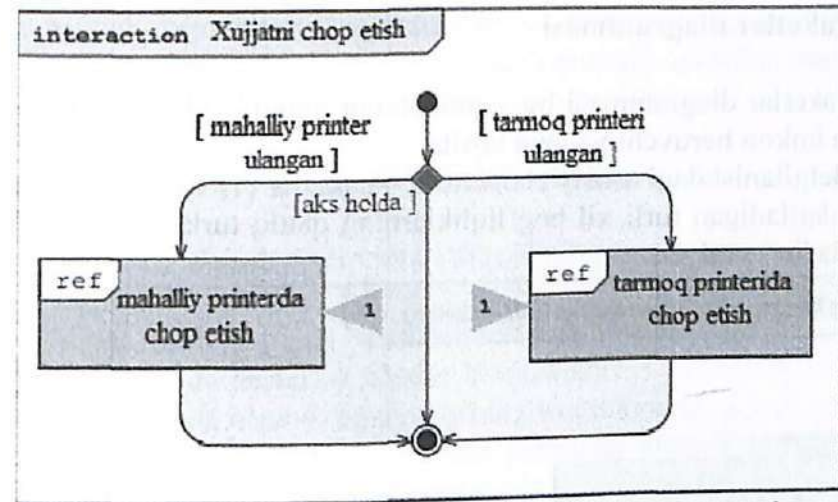


11.18- rasm. Ichki tuzilish diagrammasi

### O'zaro harakat (faoliyat) diagrammasi

O'zaro harakat diagrammasi bu – faoliyat diagrammasining kengaytirilgan sintaksisiga ega bo'lgan bir turi. O'zaro harakat diagrammasining elementlari sifatida, ketma-ktlik diagrammasi bilan aniqlanuvchi o'zaro harakatga (1) havolalar kelishi mumkin.

Belgilanishning asosiy elementlari 11.19- rasmda ko'rsatilgan.

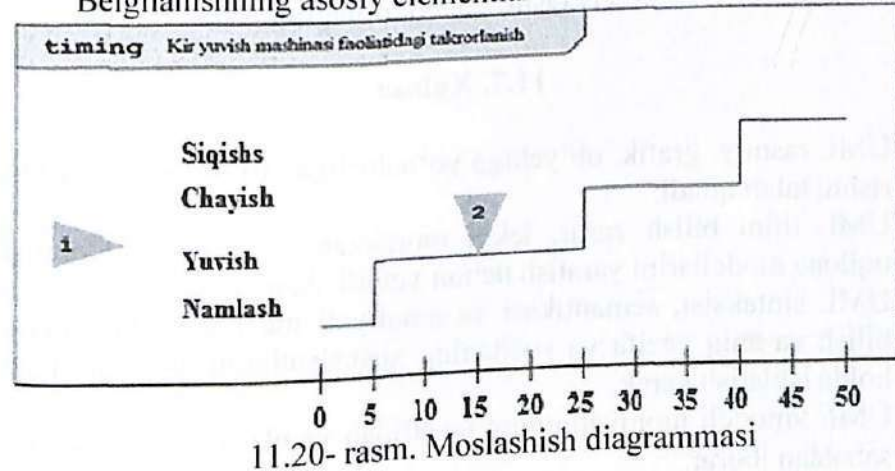


11.19- rasm. O'zaro harakat diagrammasi

### Moslashish (vaqt) diagrammasi

Moslashish diagrammasi ketma-ketlik diagrammasining maxsus shakli bo'lib, unda sinflovchi nusxalarining turli holatlardagi (1) o'zgarishlariga va ularning vaqt bo'yicha moslashishiga (2) alohida e'tibor beriladi.

Belgilanishning asosiy elementlari 11.20- rasmda ko'rsatilgan.

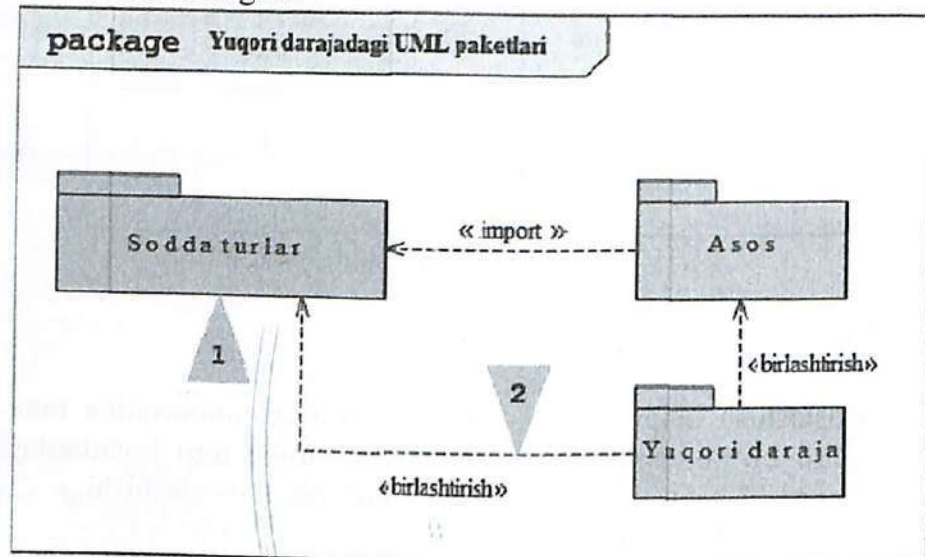


11.20- rasm. Moslashish diagrammasi

## Paketlar diagrammasi

Paketlar diagrammasi bu – modelning murakkabligini boshqarishga imkon beruvchi yagona vosita.

Belgilanishdagi asosiy elementlar – paketlar (1) va diagrammada ishlatiladigan turli xil bog'liqliklarning qattiq turlari (2) 11.21-rasmida ko'rsatilgan.



11.21- rasm. Paketlar diagrammasi

## 11.7. Xulosa

1. UML rasmiy, grafik, ob'yektga yo'naltirilgan til bo'lib, o'zlashtirishni talab qiladi.
2. UML tilini bilish zarur, lekin murakkab dasturiy tizimlarning oqilona modellarini yaratish uchun yetarli shart emas.
3. UML sintaksisi, semantikasi va amaliyoti mavjud bo'lib, ularni bilish va aniq vazifa va vositaning xususiyatlarini hisobga olgan holda ishlatish kerak.
4. UML 9modeli mohiyatlarning tavsifidan va ular orasidagi munosabatdan iborat.

5. Model elementlari diagrammalar va taqdimotlarga guruhlangan holda, modellashtirilgan tizimni turli nuqtayi nazardan eng yaxshi tavsiflash uchun xizmat qiladi.

## 11.8. Nazorat sabollari

1. Axborot tizimlarini modellashtirishning asosiy maqsadi nimadan iborat?
2. UML tilidagi asosiy jihatlar nimalardan iborat?
3. UML tilida necha xil bloklar ifodalanadi?
4. UML tilida nechta mohiyatlarning turi mavjud?
5. UML tilida mohiyatlar va ular orasidagi aloqalarni belgilashga misol keltiring.
6. UML tilida zaif va kuchli (ega) mohiyatlar va ular orasidagi aloqalarni misol keltiring.
7. UML tilida nechta turdagi diagrammalarni yaratish mumkin?
8. UML tilida xizmat qiluvchi instrumental vositalarni aytib bering.
9. Axborot tizimlarini modellashtirishda ishlatiladigan eng so'nggi til qaysi?

## XII- BOB. TIZIM TUZILMASINI UMLda MODELLASHTIRISH

### 12.1. Tuzilmani ob'yektga-yo'naltirilgan modellashtirish

Tuzilmani modellashtirish orqali biz tizimning tarkibiy qismlarini va ular orasidagi munosabatlarni tasvirlaymiz. UML ko'p hollarda ob'yektga-yo'naltirilgan modellashtirish tili sifatida ishlatiladi, shuning uchun ushbu yondashuv asosida tizimni tashkil etuvchi tarkibiy qismlarning turi sinflar va ular o'rasidagi munosabatlar ekanligi ajablanarli emas.

#### Tuzilmani modellashtirish maqsadi

Qanday tuzilmalarni va nima uchun modellashtirish kerakligini batafsil ko'rib chiqaylik. Biz quyidagi tuzilmalarga e'tibor qarata-miz:

- dasturlarning bajarilishi vaqtida ob'yektlar orasidagi aloqalardagi tuzilishga;
- berilganlarni saqlashdagi tuzilishga;
- dastur kodining tuzilishiga;
- ilovadagi tarkiblarning tuzilishiga;
- o'zaro harakatlanuvchi qismlardan iborat murakkab ob'yektlarning tuzilishiga;
- loyihadagi artefaktlarning tuzilishiga;
- ishlatiladigan hisoblash zaxiralarining tuzilishiga.

Bu sinflash to'liq va ortogonal bo'lmasligi mumkin, chunki yuqorida keltirilgan tuzilmalar mustaqil emas, ular bir-biri bilan bog'liq, lekin umuman olganda, dasturlarni ishlab chiqish amaliyotiga mos keladi, chunki u loyihalash va amalga oshirish jarayonida qabul qilingan asosiy yechimlarni qayd qilishga imkon beradi.

Ushbu bobda biz sanab o'tilgan tuzilmalarning maqsadini qisqacha muhokama qilamiz va ularni modellashtirish uchun mo'ljallangan UML vositalarini, dasturning ish vaqtida ob'yektlar orasidagi bog'lanishlarning tuzilishini ko'ramiz.

Ob'yektga-yo'naltirilgan namunada dasturlash dasturni bajarish jarayoni shundan iboratki, dasturiy ta'minot ob'yektlari xabar almashish orqali bir-biri bilan o'zaro harakatlanadi. Axborotning eng keng tarqalgan turi – bu bir sinf ob'yekti usulidan boshqa sinf ob'yekti usulini chaqirish. UMLda aloqalar tuzilishini modellashtirish uchun sinf diagrammasidagi yushma munosabatlari qo'llaniladi.

*Berilganlarni saqlashdagi tuzilish.* Kompyuter xotirasida saqlanadigan berilganlarni dasturlarda qayta ishlov beriladi. Ob'yektga-yo'naltirilgan dasturlash namunasida, dasturni bajarish paytida berilganlarni saqlash uchun sinf atributlari mo'ljallangan. Biroq, avtomatizatsiyalashtirish dasturlarining aksariyatida ish yuritish shunday tuzilganki, ma'lum berilganlar (barchasi emas) kompyuter xotirasida nafaqat dastur ishlashi paytida, balki doimiy ravishda, ya'ni ishlamagan paytlarda ham saqlanishi kerak.

Berilganlarni saqlashdagi tuzilish masalasi berilganlar bazasi ilovalari uchun juda muhimdir. Ushbu masalani hal qilishning ishonchli usullari ma'lum, masalan, «mohiyat-aloqa» diagrammasi. Xuddi shu usullar (belgilardagi aniqlik bilan) UMLda qutblarning karraligini ko'rsatadigan yushmalar shaklida qo'llaniladi.

*Dastur kodining tuzilishi.* Hech kimga sir emaski, dasturlar hajmi jihatidan sezilarli darajada farq qiladi – katta va kichik dasturlar mavjud. Ajablanarlisi shundaki, bu farqlarning qanchalik kattaligida – yuzlab kod satrlaridan (va undan kam), to yuz millionlab qatorlargacha (yoki undan ko'p). Bunday katta miqdoriy farqlar sifat darajasiga albatta ta'sir qiladi. Darhaqiqat, kichik dasturlar uchun kod tuzilishi deyarli ahamiyatsiz, katta dasturlar uchun aksincha, deyarli hal qiluvchi ahamiyatga ega. UML dasturlash tili bo'lmaganligi sababli, model kod tuzilishini to'g'ridan-to'g'ri aniqlamaydi, ammo bilvosita model tuzilishi kod tuzilishiga sezilarli

ta'sir qiladi. Ko'pgina vositalar ob'yektga yo'naltirilgan dasturlash bir yoki bir nechta tillar uchun, yarim avtomatik kod ishlab chiqarishni qo'llab-quvvatlaydi. Ko'pgina hollarda, model sinflari, maqsadli til sinflariga (yoki unga tenglashtirilgan tuzilishlarga) tarjima qilinadi. Bundan tashqari, ko'plab vositalar modeldagi paket tuzilishini hisobga oladi va uni maqsadli dasturlash tizimining tegishli «sinf yuqorisi»dagi tuzilmalariga uzatadi. Shunday qilib, agar kodni avtomatik yaratish vositasi ishtirok etsa, unda modeldagi sinflar va paketlarning tuzilishi amalda ilova kodidining tuzilishini to'liq modellashtiradi.

*Ilovadagi tarkiblarning tuzilishi.* Bitta tarkibdan tashkil topgan dastur sodda tarkibiy tuzilishga ega bo'lib, uni modellashtirishga hojat yo'q. Ammo loyihalash bosqichidagi zamonaviy ilovalarning aksariyati, hatto taqsimlanmagan bo'lsa ham, ko'plab tarkiblarning o'zaro bog'liqligini ifodalaydi. Tarkib tuzilishi ikki jihatning tavsifini o'z ichiga oladi – birinchidan, sinflar tarkibiy qismlarga qanday taqsimlangani, ikkinchidan, tarkibiy qismlar bir-biri bilan qaysi interfeyslar orqali qanday aloqa qilishini. Ushbu ikkala jihat ham UML tarkiblari diagrammalarida modellashtirilgan.

*O'zaro harakatlanuvchi qismlardan tashkil topgan murakkab ob'yektlarning tuzilishi.* Ushbu tuzilmani modellashtirish uchun yangi UML 2 vositasi – sinflovchining ichki tuzilishi diagrammasi qo'llaniladi. Ushbu diagramma sinflar va tarkibiy qismlarning ichki tuzilishini tavsiflash uchun ishlatiladi. Ko'p qismlarning o'zaro ta'sirini tasvirlashga imkon beradigan yana bir mohiyat mavjud. Ushbu mohiyat hamkorlik, deb ataladi va o'zaro harakatni ma'lum bir holatda tasvirlashga xizmat qiladi. Ichki tuzilish nuqtayi nazari-dan, hamkorlikning sinf va tarkibdan asosiy farqi shundaki, hamkorlik o'z qismlarining egasi emas, va undagi qismlarning ulagichlari uyushma sifatida aniq ifodalanmasligi mumkin. Biroq, sinflar va tarkibiy qismlar singari, hamkorlikda ham ijro paytida ishlaydigan nusxalarga ega bo'lishi mumkin.

*Loyihadagi sun'iy dalil (artefakt)larning tuzilishi.* Faqat eng oddiy ilovalar bitta artefakt – dasturning bajariluvchi kodidan

iborat. Aniq dasturlarning aksariyati o'nlab, yuzlab va minglab turli xil tarkibiy qismlarni o'z ichiga oladi – bajariluvchi ikkilik fayllar, zaxiralar fayllari, boshlang'ich kod fayllari, berilganlar fayllari, turli kuzatuv hujjatlari, ma'lumotnomalar fayllari va x.k. Katta dastur uchun nafaqat barcha artefaktlarning aniq va to'liq ro'yxatiga ega bo'lish, balki qaysi biri tizimning ma'lum bir nusxasiga kiritilganligini ham ko'rsatish muhimdir. Gap shundaki, katta dasturlar uchun loyihada bir xil artefaktning turli xil naqllari mavjud. Bu har xil turdagi artefaktlarni tavsiflash uchun standart qat'iy turlarni ta'minlaydigan UMLdagi tarkiblar va joylashtirish diagrammalari tomonidan to'liq modellashtiriladi.

*Amaldagi hisoblash zaxiralarning tuzilishi.* Ko'pgina artefakt-lardan tashkil topgan dastur odatda taqsimlovchi bo'ladi, ya'ni turli xil artefaktlar turli xil kompyuterlarga joylashtiriladi. Joylashtirish diagrammalari ushbu tuzilmaning ham tavsifini modelga kiritishga imkon beradi.

## 12.2. Sinflovchilar va ularning eng muhim xususiyatlari

Tavsiflovchilarning eng muhim turi sinflovchilardir. Sinflovchi bu – bir xil turdagi ob'yektlar to'plamining tavsiflovchisidir. Bu ta'rifdan to'g'ridan-to'g'ri tavsiflovchining asosiy va tavsifli xususiyati kelib chiqadi, ya'ni sinflovchi (to'g'ridan-to'g'ri yoki bilvosita) nusxalarga ega bo'lishi mumkin. UMLda juda ko'p tavsiflovchilar aniqlangan. Biz ularni qismlarga ajratib ko'rib chiqamiz. Yuqorida biz shulardan ikkitasini, ya'ni harakatlanuvchi shaxs (aktyor) va fofdalanish variantlarini (use case) batafsil ko'rib chiqdik.

Endi biz quyidagi sinflovchilarni ko'rib chiqamiz:

- artefakt (artifact);
- berilganlar turi (data type);
- uyushma (association);
- uyushma sinfi (association class);
- интерфейс (interface);
- sinf (class);

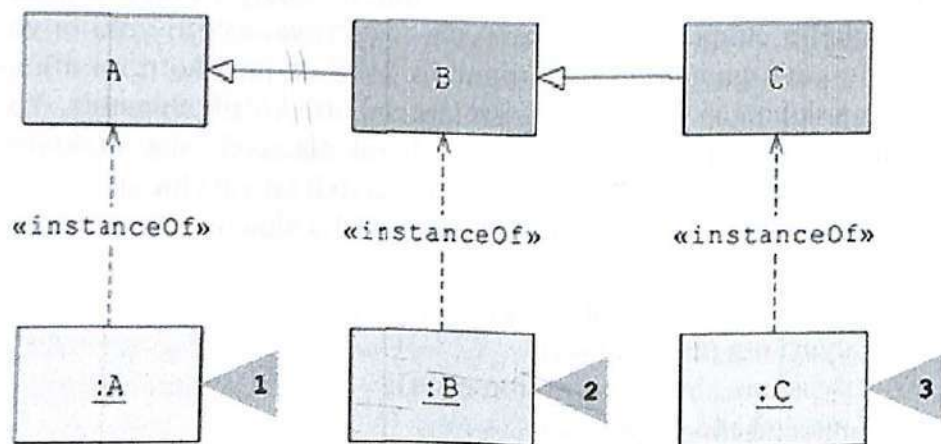
- hamkorlik (collaboration);
- tarkib (component);
- tugun (node).

Barcha sinflovchilar quyidagi muhokamada qo'llaniladigan ba'zi umumiy xususiyatlarga ega. Ushbu bo'limda biz birinchi navbatda kerak bo'ladigan sinflovchilarning ettita eng muhim xususiyatini tasvirlab beramiz.

*Birinchi*dan, barcha model elementlari kabi, sinflovchilar ham o'z nomlariga ega. Ism, model elementini aniqlash uchun xizmat qiladi va shuning uchun shu nomlar maydonida noyob bo'lishi kerak.

*Ikkinchi*dan, yuqorida aytib o'tilganidek, sinflovchining nusxalari bo'lishi mumkin. Nusxalarlar to'g'ridan-to'g'ri yoki bilvosita bo'lishi mumkin. Agar biron-bir ob'yekt to'g'ridan-to'g'ri A sinflovchining konstruktoriga yaratilgan bo'lsa, u holda bu ob'yekt A sinflovchining to'g'ridan-to'g'ri nusxasi deyiladi (12.1-rasm).

Agar A sinflovchi B sinflovchining umumlashmasi bo'lsa, yoki xuddi shunday, B sinflovchi A sinflovchining ixtisoslashuvi bo'lsa, B sinflovchining barcha nusxalari A sinflovchining bilvosita nusxalaridir (12.1-rasm).



12.1- rasm. A sinflovchining bevosita va bilvosita nusxalari

Bu xususiyat tranzitivdir – agar A sinflovchi B sinflovchining umumlashmasi bo'lsa va B sinflovchi C sinflovchining umumlashmasi bo'lsa, C sinflovchining barcha nusxalari ham A ning bilvosita nusxalari hisoblanadi.

*Uchinchi*dan, sinflovchi mavhum yoki aniq bo'lishi mumkin. Mavhum (abstract) sinflovchi to'g'ridan-to'g'ri nusxalarga ega bo'lishi mumkin emas va bu holda uning nomi kursiv bilan yoziladi. Aniq sinflovchi to'g'ridan-to'g'ri nusxalarga ega bo'lishi mumkin, va bu holda uning nomi oddiy shriftda yoziladi.

Abstrakt sinflovchi bu – ob'ektlar to'plamining shunday tavsiflovchi hisoblanadiki, unda to'plam elementlarining to'g'ridan-to'g'ri tavsifi bo'lmaydi, lekin bu sinflovchi boshqa sinflovchilar bilan umumlashtirish munosabati bilan bog'langan bo'ladi va ular nusxalari to'plamlarining birlashishi ushbu mavhum sinflovchi nusxalarining to'plami hisoblanadi. Boshqacha qilib aytganda, to'plam to'g'ridan-to'g'ri emas, balki qism-to'plamlar jamlamasi orqali aniqlanadi. Masalan, mavhum sinf bo'lgan interfeys to'g'ridan-to'g'ri nusxalarga ega bo'lolmaydi, lekin uni amalga oshiradigan sinf to'g'ridan-to'g'ri nusxalarga ega bo'lishi mumkin, shuning uchun interfeys sinflovchi bo'la oladi.

*To'rtinchi*dan, sinflovchi (modelning boshqa elementlari kabi) ko'rish chegarasiga (visibility) ega. Ko'rish chegarasi sinflovchi tarkibidagilardan birining boshqa sinflovchida ishlatilishini mumkinligini aniqlaydi. Boshqacha qilib aytganda, agar ma'lum bir kontekstda biror narsa mavjud, va unga kirish va uni qandaydir tarzda ishlatish mumkin bo'lsa, uni ko'rish chegarasi ana shu kontekst doirasida bo'ladi. Agar u ko'rinmasa, uni ishlatib bo'lmaydi.

Garchi ko'rish chegarasi xususiyati barcha elementlar uchun muhim bo'lmasa ham, barcha model elementlarining xususiyatidir. Ko'rish chegarasi to'rtta qiymatdan birini qabul qilishi mumkin:

- *ochiq* (+ belgisi yoki **public** kalit so'z bilan ko'rsatiladi);
- *himoyalangan* (# belgisi yoki **protected** kalit so'z bilan ko'rsatiladi);

- *yopiq* (- defies belgisi bilan yoki *private* kalit soʻz koʻrsatiladi);

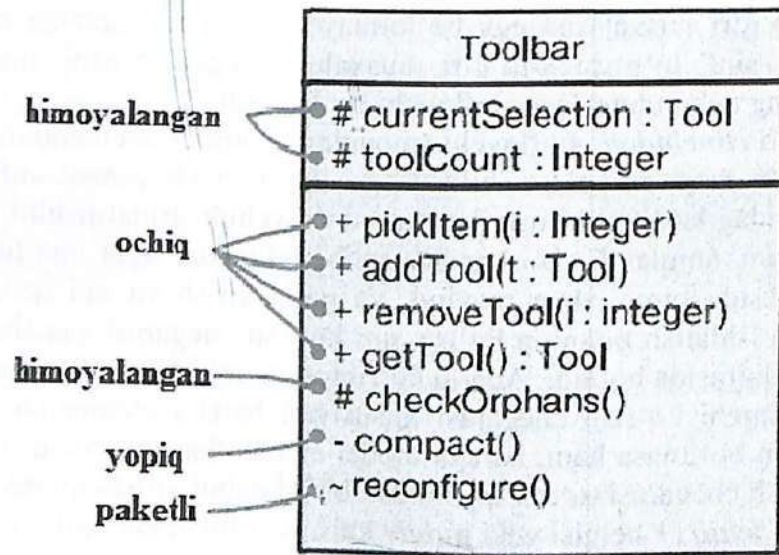
- *paketli* (~ tilde belgisi bilan yoki *package* kalit soʻz bilan koʻrsatiladi).

Ochiq model elementi hamma joyda koʻrinadi, uni oʻz ichiga olgan element ham koʻrinadi. Masalan, sinfnig ochiq atributi sinfnig oʻzi koʻrinadigan hamma joyda koʻrinadi (12.2- rasm).

Modelning himoyalangan elementi uni oʻz ichiga olgan element (konteyner)da boʻlgani kabi, konteyner umumlashma boʻlgan barcha elementlarda ham koʻrinadi. Masalan, sinfdagi himoyalangan atribut uni oʻz ichiga olgan sinfda va barcha qism-sinflarga koʻrinadi (12.2- rasm).

Yopiq model elementi faqat tegishli boʻlgan elementda koʻrinadi. Masalan, sinfdagi yopiq atribut faqat shu sinfda koʻrinadi (12.2- rasm).

Koʻrish chegarasi paketli xususiyati ega model elementi, faqat oʻzi aniqlangan paketdagi elementlarga koʻrinadi (12.2- rasm).



12.2- rasm. Koʻrish chegarasi

*Sinflowchi vositalarining koʻrinish chegarasini koʻrsatganimizda, biz odatda uni amalga oshirishdagi tafsilotlarni yashirasiz va faqat mavhumlik vazifalarini bajarish uchun zarur boʻlgan xossalarni koʻrinadigan qilasiz. Bu maʼlumotni yashirishning asosi boʻlib, mustahkam va barqaror tizim qurilishini taʼminlaydi. Agar biz xossani aniq koʻrinadigan belgi bilan taʼminlamasak, u holda oshkormas holatda public turi qabul qilinadi.*

*Beshinchidan, sinflowchining barcha tarkiblari qoʻllanish sohasi (scope)ga ega. Qoʻllanish sohasi, sinflowchi nusxalarining tarkibi oʻzini qanday namoyon qilishini belgilaydi, yaʼni nusxalar oʻz qiymatiga egami, yoki hammasi bitta qiymatdan foydalanadimi.*

Qoʻllanish sohasi, ikkita mumkin boʻlgan qiymatga ega:

- *nusxa* (instance) - hech qanday tarzda maxsus belgilanmaydi, chunki u oshkormas ravishda ishlatiladi;

- *sinflowchi* (classifier) - sinflowchi tarkibining tavsifi tag chiziq bilan belgilanadi.

Agar tarkibning qoʻllanish sohasi nusxa boʻlsa, y holda sinflowchining har bir nusxasi tarkibdagi oʻz qiymatiga ega. Masalan, atributning qoʻllanish sohasi oshkormas ravishda nusxadan iborat. Bu shuni anglatadiki, sinfnig nusxasi boʻlgan har bir obʼyekt - oʻz atributining qiymatiga ega boʻlishi, va u shu sinf nusxalari boʻlgan boshqa obʼyektlarning ushbu atributining qiymatlaridan qatʼi nazar oʻzgarishi mumkin.

Agar tarkibning qoʻllanish sohasi sinflowchi boʻlsa, u holda sinflowchining hamma nusxalari tarkibning bitta qiymatidan birgalikda foydalanadi. Masalan, konstruktorning odatdagi qoʻllanish sohasi sinflowchi (sinf) doirasiga ega, chunki bu maʼlum bir sinfnig barcha nusxalari uchun umumiy protsedura.

*Oltinchidan, sinflowchi karralik xususiyatiga ega, yaʼni toʻplam kabi, sinflowchi nusxalarining soni boʻyicha cheklov mavjud. Karralikni, nusxalardagi elementlar soni bilan aralashtirib yubormaslik kerak. Modelda koʻrsatilgan toʻplam ish vaqtida turli sondagi elementlarga ega boʻlishi va elementlar soni dinamik*

ravishda o'zgarishi mumkin. Karralik, bu o'zgarishlarning chegaralarini belgilaydi.

To'plamning karraligi (multiplicity) – bu sonlar to'plamidan iborat bo'lib, ular shu to'plam quvvatining mumkin bo'lgan barcha qiymatlarini belgilaydi. Sintaktik jihatdan karralik o'z ifodasi bilan beriladi, ya'ni bo'sh bo'lmagan, vergul bilan ajratilgan, elementlar ketma-ketligi, ularning har biri quyidagi formatga ega:

**Pastki chegara .. Yuqori chegara**

Yuqori va pastki chegaralar sifatida natural sonlar yoki nol qo'llaniladi. Bundan tashqari, \* belgisi yuqori chegara sifatida ishlatilishi mumkin. Agar pastki chegara belgilanmagan bo'lsa, u .. belgisi (ikki nuqta) bilan birga olib tashlanadi. 12.1- jadvalda ko'plik ifodalarga ba'zi misollar keltirilgan.

12.1- jadval. Ko'plik ifodalari

Karralik ifodasi	To'plamda bo'lishi mumkin
0..*	Ixtiyoriy sondagi elementlar
*	Ixtiyoriy sondagi elementlar (oldingiga ekvivalent)
1..*	Bir yoki undan ortiq elementlar
0..1	Bittadan ortiq bo'lmagan elementlar
1..10	Birdan o'gacha bo'lgan elementlar
1..3, 5, 7..10	Bir, ikki, uch, besh, etti, sakkiz, to'qqiz, toki o'ta elementlar
5..3	Noto'gri karralik. Quyi chegara yuqori chegaradan katta
-1..3	Noto'gri karralik. Manfiy sonlar mumkin emas

Va nihoyat, yettinchidan, sinflovchilar (va faqat ular!) umumlashtirish munosabatida ishtirok etishlari mumkin.

Modellashtirish jarayonida biz haqiqiy dunyo mohiyatlari va bizning qarorimizning tarkibiy qismlari bo'lgan mohiyatlarni ifodalovchi mavhumlarni tadqiq o'lamiz. Misol uchun, agar biz Web-ga

yo'naltirilgan buyurtmalarni qayta ishlash tizimini qurmoqchi bo'lsak, unda bizning loyihamiz lug'atimizda *Mijozlar (xaridor)* va *Tranzaksiya* sinflari bo'lishi mumkin. Birinchisi, mahsulotlarga buyurtma byergan shaxslarni, ikkinchisi esa zarracha harakatni ifodalovchi artefaktni tasvirlaydi. Kengaytirilgan tizimda biz har bir mijoz kodidagi nusxalar bilan *Pricing (narxlar)* tarkibini kiritishimiz mumkin. Ushbu mavhumlarning har biri o'z nusxalariga ega bo'ladi. Mohiyatlar va ularning nusxalarini ajratish modellashtirishning muhim qismidir.

UMLdagi ba'zi elementlar, masalan paketlar yoki ulanishlarni umumlashtirishning nusxalari yo'q. Nusxalarga ega bo'lgan modelashtirish elementlari esa – sinflovchidir. Ammo ularning eng muhim atomi, atributlar shaklidagi tarkibiy xossalarga va amallar shaklidagi xulq-atvorning mavjudligi. Sinflovchining barcha nusxalari umumiy xususiyatlarga ega, ammo ularning har biri har bir atribut uchun o'ziga xos qiymatga ega.

UMLdagi sinflovchilarning eng muhim turi bu sinf, bir xil atributlar, amallar, aloqalar va semantikaga ega bo'lgan ob'yektlar termasining tavsifi. *Biroq, sinflar sinflovchilarning yagona turi emas.* UML – modellashtirish uchun muhimligi kam bo'lmagan boshqa ko'plab turlarni taqdim etadi:

➤ *interfeys* – sinf yoki tarkib xizmatini tasnif qilish uchun ishlatiladigan amallar termasi;

➤ *berilganlar turi* – qiymatlari o'zgarmas tur. Misollar, ibtidoiy joylangan turlar (raqamlar, satrlar va boshqalar), sanaluvchi turlar (boolean va boshqalar).

➤ *uyushma* – havolalar termasining tavsifi, ularning har biri bir qator ob'yektlarni birlashtiradi;

➤ *signal* – nusxalar o'rasida uzatiladigan asinxron xabarning tasnifi;

➤ *tarkib* – tizimning modulli qismi bo'lib, o'z faoliyatini tashqi interfeyslar termasidan keyingi o'ringa yashiradi;

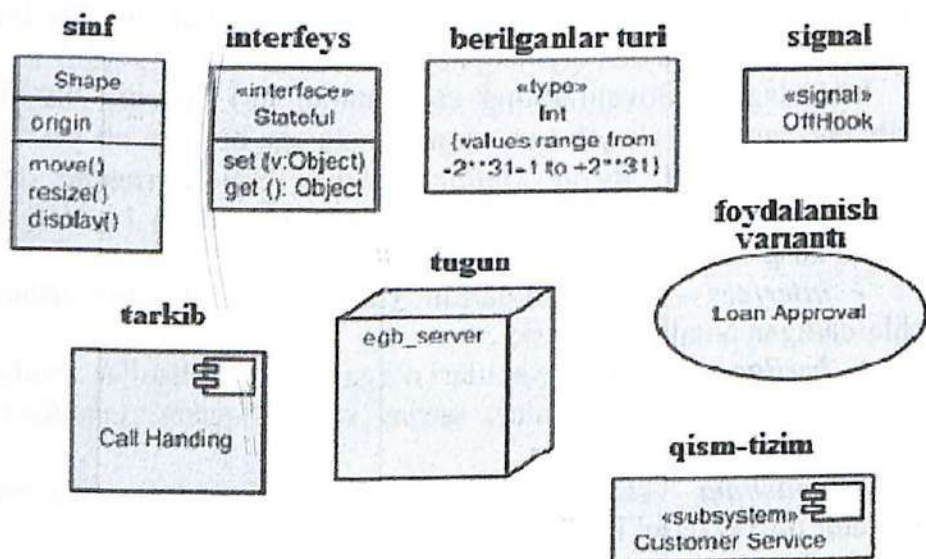
➤ *tugun* – bu ijro paytida mavjud bo'lgan va hisoblash manbasini ifodalovchi jismoniy element bo'lib, odatda kamida xo-

tira va ko'pincha hisoblash qobiliyatiga ega bo'ladi;

➤ **foydalanish variant** – tizim tomonidan amalga oshiriladigan va ma'lum bir aktyor uchun muhim natijani keltirib chiqaradigan harakatlar (shu jumladan ularning turlarini) ketma-ketligini tavsiflash;

➤ **quyi tizim** – tizimning asosiy qismni ifodalovchi tarkib.

Odatda, sinflovchilarning har bir turi tuzilmaviy va xulq-atvorlik xossalari ega bo'lishi mumkin. Bundan tashqari, tizimni ushbu sinflovchilardan biri bilan modellashtirishda, biz mavhumlik ma'nosini etkazish va etarli tafsilotlar darajasini ta'minlash uchun ushbu bobda keltirilgan barcha kengaytirilgan xossalardan foydalanishimiz mumkin. Grafik jihatdan UML ushbu turdagi sinflovchilarning barchasini 12.3- rasmda keltirilgan.



12.3- rasm. UMLdagi sinflovchilar

### 12.3. Nusxaning harakatlanish sohasi va statik faoliyat doirasi

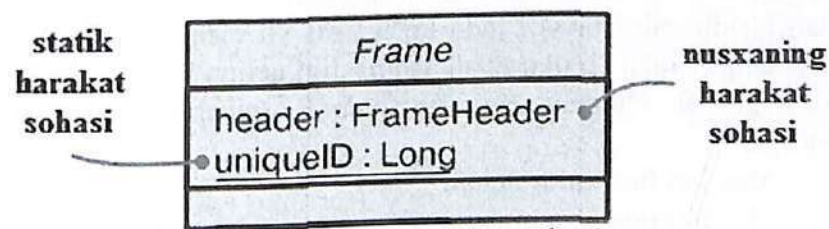
Sinflanuvchining atributlari va amallari uchun muhim bo'lgan

yana bir tafsilot – bu harakatlanish sohasidir (scope). Xossaning harakat doirasi shuni ko'rsatadiki, sinflovchining har bir nusxasi ushbu xossaning o'ziga xos qiymatiga ega bo'lishi mumkinligini yoki sinflovchining barcha nusxalari uchun yagona qiymat mavjud bo'lib, u sinflovchining hamma nusxalari orasida taqsimlanishi kerakligini ko'rsatadi. UMLda ikki xildagi soha mavjud:

✓ **instance** – bu nusxaning harakatlanish sohasi. Sinflovchining har bir misoli o'ziga mos xossaning qiymatiga ega. Bu variant oshkormas ravishda qabul qilingan va qo'shimcha belgilarni talab qilmaydi;

✓ **static** – statik harakat sohasi yoki sinfnining harakat sohasi. Faqat bitta qiymat taqdim etiladi Sinflovchining barcha nusxalari uchun faqat bitta xossa taqdim qilinadi.

12.4- rasmda statik maydonga ega bo'lgan atribut ko'rsatilgan, bu uning nomi tag chiziq bilan ta'kidlangan. Nusxaning harakat sohasini akslantirishda hech qanday qo'shimchalar ishlatilmaydi.



12.4- rasm. Harakat sohasi

Umuman olganda, aksariyat xossalar (atributlar va amallar) biz modellashtirishimiz kerak bo'lgan sinflovchilar nusxasi doirasiga ega bo'ladi. Statik sohada yopiq atributlar mavjud bo'lib, ular sinf nusxalari o'rasida taqsimlanishi kerak. Masalan, yangi sinf nusxalarining noyob aniqlovchilarini yaratish uchun.

**Eslatma.** Statik harakatlanish sohasi, C++ va Java tillaridagi statik atributlar va amallarga mos keladi.

Amallarga nisbatan statik harakatlanish sohasi biroz boshqacha harakat qiladi. Nusxa amali, oshkormas ishlov beriladigan

ob'yektni ko'rsatadigan yashirin parametrğa ega. Statik amalda bunday parametr yo'q, u maqsadli ob'yeksiz an'anaviy global protseduraga o'xshaydi. Statik amallar nusxalarni yaratish yoki statik atributlarni harakatlantirish uchun ishlatiladi.

#### 12.4. Sinf diagrammasidagi mohiyatlar

Sinf diagrammalarida mohiyatlar sifatida birinchi navbatda sinflar o'zining umumiy shaklida, hamda ko'p sonli qat'iy turdagi shakllar sifatida va hususiy holatlarda – interfeyslar, berilganlarning turlari, faol sinflar va boshqalar sifatida qo'laniladi. Shuningdek, sinflar diagrammasida, hamma joydagi kabi, paketlar va izohlardan foydalanish mumkin.

#### Sinflar

Sinf – UMLdagi modellashtirishning eng "boy" elementlaridan biridir. Sinf tavsifi juda ko'p turli xil elementli to'plamlarni o'z ichiga olishi va ular chalkashmasligi uchun UMLda keltirilgan sinf tavsifi elementlarini *bo'limlash* (*compartment*) nazarda tutilgan.

Standart bo'limlar uchta:

- bo'lim nomi va undan tashqari qat'iy turni, karralik va nomlangan qiymatlar ro'yhatini o'z ichiga olishi mumkin;
- atributlar bo'limi – sinfdagi atributlar tavsiflarining ro'yxatini o'z ichiga oladi;
- amallar bo'limi – sinfdagi amallar tavsiflari ro'yxatini o'z ichiga oladi.

Barcha asosiy UML mohiyatlari singari, sinf ham albatta nomga ega bo'lishi va shuning uchun bo'lim nomi ham yozilishi kerak. Boshqa bo'limlar bo'sh yoki umuman yo'q bo'lishi mumkin. Standart bo'limlar bilan bir qatorda, sinf tavsifini va o'z ichiga qo'shimcha ixtiyoriy sondagi bo'limlarni olishi mumkin. Qo'shimcha bo'limlar ma'no jihatidan izohga tenglashtirilgan.

Sinf tasviri juda oddiy — u har doim to'rtburchak shaklida bo'ladi. Agar bo'limlar bir nechta bo'lsa, to'rtburchakning ichki qismi gorizontaal chiziqlar bilan bo'limlarga mos keladigan qismlarga ajratiladi (12.5- rasm).

Sinf Nomi
+attribute -privateAttr -fio:String="Novikov" -array:char[10]
+operationName() -staticOperation() +function():int

12.5- rasm. Sinfning namunaviy tasviri

Har qanday holatda ham bo'lim matndan iborat bo'ladi. Matn standart bo'limlar ichida ma'lum bir sintaksisga ega bo'lishi kerak. Sinf nomi bo'limi umumiy tarzda quyidagi sintaksisga ega:

«qat'iy tur» NOMI {xossasi} karrasi

Sinf nomidan oldin qat'iy tur kelishi mumkin. 12.2- jadvalda sinflarning standart qat'iy turlari ro'yxati keltirilgan.

12.2- jadval. Sinflardagi standart qat'iy turlar

Qat'iy turlar	Tavsifi
actor	Ishtirokchi shaxs
enumeration	Berilganlarning sanoq turi
exception	Istisno (faqat UML 1 uchun)
interface	Tarkibdakilarning hammasi mavhum
metaclass	Nusxalari sinflardan iborat
signal	Nusxalari signallardan iborat sinf
qat'iy tur	Mavjudga asoslangan yangi element

<b>dataType</b>	Berilganlar turi
<b>utality</b>	Nusxalari yo'q, xizmat

Sinfning majburiy nomi kursiv bilan ifodalanisi va bu holda shu sinf mavhum bo'ladi, ya'ni bevosita nusxalari bo'maydi. Sinf va uni tavsiflovchi alohida elementlari foydalanuvchi tomonidan qo'yilgan ixtiyoriy chegaralar va nomlangan qiymatlarga ega bo'lishi mumkin. Sinfning karraligi umumiy qoida bo'yicha beriladi.

### Atributlar

Atribut – bu qiymati saqlanishi mumkin nomlangan joy (yoki uya).

Sinf atributlari atributlar bo'limida keltiriladi. Umuman olganda, atribut tavsifi quyidagi sintaksisga ega:

**ko'rinish chegarasi NOMI karraligi: turi = boshlang'ich\_ iymati {xossasi}**

Ko'rinish chegarasi, odatdagidek, +, -, #, ~ belgilari bilan ko'rsatiladi. Agar ko'rinish chegarasi ko'rsatilmagan bo'lsa, unda oshkormas ravishdagi ko'rinish chegarasi qiymati nazarda tutilmaydi.

Agar atribut nomi tagiga chizilgan bo'lsa, ushbu atributning faoliyat chegarasi odatdagidek sinf nusxasi emas, balki sinfdan iborat bo'ladi. Boshqacha qilib aytganda, barcha ob'yektlari – ushbu sinfning nusxalari, nusxalar uchun umumiy bo'lgan ushbu atributning bitta qiymatidan foydalanadi. Oddiy vaziyatda (tagi chizilmaganda) sinfning har bir nusxasi atributning shaxsiy qiymatini saqlaydi.

Karralik, agar mavjud bo'lsa, berilgan atributni aniq yoki noaniq o'lchamdagi massiv sifatida belgilaydi.

**Atribut turi** – bu joylangan sodda tur yoki foydalanuvchi tomonidan belgilangan turdir. Sinfning nusxasi yaratilganda atribut o'zining berilgan boshlang'ich qiymatini oladi.

Agar boshlang'ich qiymat ko'rsatilmagan bo'lsa, oshkormas ravishdagi hech qanday qiymat nazarda tutilmaydi. Agar atribut

qiymatga ega bo'lishi zarur bo'lsa, unda sinf konstruktori bunga g'amxo'rlik qilishi kerak.

Boshqa har qanday model elementi singari, atribut ham cheklovlar va nomlangan qiymatlar shaklidagi qo'shimcha xossaga ega bo'lishi mumkin.

Masalan, "Xodimlar bo'limi" axborot tizimida Shaxs sinfida xodimning ismini saqlovchi atribut (xususiyat) mavjud bo'lishi kerak.

12.3- jadvalda shunday atributning tavsiflariga misollar keltirilgan. Tavsifda hamma sintaktik mumkin bo'lgan holatlar keltirilgan bo'lib, ularni model tavsilotlarining mos darajalarida foydalanish mumkin.

12.3- jadval. Atributlar tavsiflariga misollar

Misol	Tavsifi
<b>name</b>	mumkin bo'lgan eng kichik tavsif – faqat atribut ismi ko'rsatilgan
<b>+name</b>	nomi va ochiq ko'rinish chegarasi ko'rsatilgan – ismi bilan harakatlari bevosita bajariladi
<b>-name : string</b>	nomi, turi va yopiq ko'rinish chegarasi ko'rsatilgan – ismiga bo'g'liq harakatlari amallar bilan bajariladi
<b>-nami[1..3] : string</b>	oldingisini qo'shimcha ravishda karralik berilgan (3ta tarkib – familiyasi, ismi va otasining ismi)
<b>-name : string="Soliey"</b>	qo'shimcha ravishda boshlang'ich qiymat berilgan
<b>+name : string {Read Onli}</b>	atribut boshlang'ich qiymatni olgandan keyin qiymatini o'gartirmaydigan va tarzda tavsiflangan

## Amallar va usullar

Amal – ob'yekt bilan ishlash tasnifi bo'lib, atributlar qiymatini o'zgartirish, ob'yektda saqlangan xabarga nisbatan yangi qiymatni hisoblash va boshqa ishlarni amalga oshiradi. Sinfda ma'lum bir amalni e'lon qilish, xuddi shu sinfda usulning mavjudligi nazarda tutadi. Agarda amal mavhum deb e'lon qilingan va uni amalga oshirish qism-sinflarda bajarilishi – istisno holat sifatida ko'riladi.

Usul – bu amalnu bajarishdan, yani bajariluvchi algoritmdir. Amal bilan aniqlangan harakatlarni bajarish – usulni chaqirish tashabbuskori orqali amalga oshiriladi. Usulni chaqirishda, o'z navbatida, xuddi shu va boshqa sinflardagi usullar ham chaqirilishi mumkin. Sinf amallarining tavsiflari amallar bo'limida keltiriladi va quyidagi sintaksisga ega bo'ladi:

**ko'rinish chegarasi NOMI (parametrlar) : turi {xossalari}**

Bu yerda parametr so'zi amal parametrlaridagi tavsiflar ketma-ketlikni bildiradi. Ularning har biri quyidagi formatga ega bo'lishi kerak:

**yo'nalish PARAMETR : tur = qiymat**

Keling, birma-bir tahlil qilamiz. Ko'rinish chegarasi, odatdagidek, +, -, #, ~ belgilardan yoki *private, public, protected, package* kalit so'zlari bilan ifodalanadi. Ismdagi tag chiziq, amalning harakat sohasi ob'yekt emas, balki – sinf ekanligini anglatadi. Masalan, konstukturarning harakat sohasi sinf. Ismning kursiv bilan ifodalanishi amalning mavhum ekanligini anglatadi, ya'ni shu sinfda uni amalga oshirish ko'rsatilmagan va shu sinfning qism-sinflarida ko'rsatilishi kerak. Ismdan keyin parametr tavsiflari ro'yxati qavs ichida ko'rsatilishi mumkin. Ro'yxatdagi parameterlarning tavsiflari vergul bilan ajratilishi kerak. Har bir parameterning ismi, uzatish yo'nalishi, turi va argument qiymati oshkormas ravishda ko'rsatilishi kerak. UMLga parametrni uzatish yo'nalishi, parametrlarning semantik tayinlanishi orqali tavsiflanadi, muayyan uzatish mexanizmi ko'rsatilmaydi.

Modelda ko'rsatilgan parametr-larning uzatish yo'nalishlarini aniq talqin qilish qo'llaniladigan tizimli dasturlashga bog'liq. Uzatish yo'nalishining mumkin bo'lgan qiymatlari, o'tkazish yo'nalishini tavsiflash uchun kalit so'zlar 12.4- jadvalda keltirilgan.

12.4- jadval. Kalit so'zlar va uarning tavsifi

Kalit so'z	Parametrning vazifasi
<b>in</b>	kiruvchi parametr – amalda bajariladigan, o'zgar-mas qiymatli argument bo'lishi kerak.
<b>out</b>	chiquvchi parametr – argument saqlovchi bo'lishi, amal esa unda qiymat saqlashi kerak.
<b>inout</b>	kiruvchi va chiquvchi parametr – argument saq-lovchi bo'lishi, unda qiymat saqlanishi kerak. Amal argument uzatgan qiymatdan foydalanadi va natijani saqlovchiga joylashtiradi.
<b>return</b>	Amal qaytaruvchi qiymat. Qaytariluchi qiymat uchun avtomatik tarzda yo'nalish o'natiladi.

Amal parametrining turi, shuningdek, amal tomonidan qay-tarilgan qiymat turi modelda belgilangan ixtiyoriy joylangan tur yoki sinf, interfeys yoki berilganlar turi bo'lishi mumkin. Ham-masi birgalikda (amal nomi, parametrlari va natija turi) odatda *amal belgisi* deb ataladi.

12.3- jadvaldagi "Xodimlar bo'limi" axborot tizimining *Shaxs* sinfining mumkin bo'lgan amallarini tavsiflash misollarini ko'rib-chiqamiz (12.5- jadval). Mavhum bo'lishi mumkin bo'lgan amal-dan farqli o'laroq, ya'ni amalga oshirish usuli va usul aniqlangan o'ziga xos xususiyatga ega bo'lish uchun UMLda usulning o'zini tavsiflash uchun alohida belgi taqdim etilmagan. Belgilasgda aks ettirilmagan boshqa shunga o'xshash holatlarda bo'lgani kabi, eslatmadan foydalanish almasish o'rnida ishlatilishi mumkin.

12.5- jadval. Amallarni tavsiflashga misollar

Misol	Tushuntirish
move	mumkin bo'lgan eng kichik tavsif – faqat amal ismi ko'rsatilgan
+move (in from, in to)	ko'rinish chegarasi, uzatish yo'nalishi va parametrlarning ismi ko'rsatilgan
+move (in from : Department, in to: Department)	batafsil belgilash tavsifi ko'rsatilgan – amalning ko'rinish chegarasi, uzatish yo'nalishi, parametrlarning ismi va turi
+getName () :String {isQuery}	atributning qaytarilayotgan qiymati va samarali natijaga ega funksiya

### Interfeyslar va berilganlar turlari

UMLda sinflorvchilarning bir nechta maxsus holatlari mavjud bo'lib, ular sinflar kabi tuzilmani modellashtirish uchun mo'ljallangan va bir qator o'ziga xos xususiyatlarga ega. Ulardan eng muhimi interfeyslar va berilganlar turlaridir. Interfeys – bu xizmat ko'rsatuvchi provayderlar va iste'molchilar o'rasidagi shartnomani tavsiflovchi nomli tarkiblar termasi. Boshqacha qilib aytganda, interfeys mavhum sinf bo'lib, undagi barcha tarkiblar – atributlar va amallar mavhumdir. Interfeysi mavhum sinf bo'lgani uchun uning bevosita nusxalari bo'lishi mumkin emas.

Muhokama uchun keyingi mavzu – berilganlar turlari. Berilganlar turi – ikki narsaning birlashmasidir – bular qiymatlar to'plami (juda katta yoki hatto potentsial cheksiz bo'lishi mumkin) va ushbu qiymatlarga tegishli cheklangan amallar to'plami. UML modelida uch turdagi berilganlar turlaridan foydalanish mumkin. UMLda oldindan belgilangan, deb hisoblanuvchi quyidagi turlar mavjud:

- sodda turlar (PrimitiveType);
- butun sonlar turi (Integer);
- Bul turi (Boolean);
- satr turi (String);
- butun sonlar to'plamini (cheksiz bo'lishi mumkin) tavsiflovchi UnlimitedNatural turi.

Oxirgi tur asosan mohiyatning karraligini ko'rsatish uchun ishlatiladi. Bositalar ushbu to'plamni kengaytirish va boshqa mos nomlardan foydalanish huquqiga ega.

*Asbob tomonidan qo'llab-quvvatlanadigan dasturlash tilida aniqlangan berilganlar turlar.* Bular joylangan turdagi nomlar yoki til tomonidan ruxsat etilgan bo'lsa, turni etkazib beradigan murakkab iboralar bo'lishi mumkin.

Modelda foydalanuvchi tomonidan aniqlangan berilganlar turlari. Ushbu turlar «enumeration» yoki «dataType» qat'iy turi sifatida sinflorvchilarga taqdim etiladi.

Berilganlarning sanaluvchi turi alohida e'tiborga loyiqdir. Masalan, Bul turi UMLda haqiqiy va yolg'on qiymatlari bilan sanaluvchi tur sifatida aniqlanadi. Agar loyihalashtirilayotgan ilovada biz oddiy ikki qiymat mantiqdan emas, balki uch qiymatli mantiqdan foydalanmoqchi bo'lsak, unda tegishli turni 12.6- jadvalda ko'rsatilganidek aniqlash mumkin.

12.6- jadval. Berilganlarning “uch qiymatli mantiq” turi

«enumeration» 3Logic
yes no unknown
+add(a:3Logic, b:3Logic) <u>+or(a:3Logic, b:3Logic)</u>

## 12.5. Sinf diagrammasidagi munosabatlar

Sinf diagrammalaridagi mohiyatlar asosan uyushma munosabatlari, shu jumladan yig'ish va kompozitsiya, va umumlashtirish orqali bog'lanadi. Bog'liqlik va amalga oshirish munosabatlari sinf diagrammalarida kamroq qo'llaniladi, ammo shunga qaramay, ular ham qo'llaniladi va biz ularni oddiyroq, deb hisoblab tahlilni shulardan boshlaymiz.

### Bog'liqlik va amalga oshirish o'rasidagi munosabatlar

Umuman olganda, UML bir nechta guruhlariga bo'linishi mumkin bo'lgan juda ko'p standart qat'iy turlardagi tobelik munosabatlarini belgilaydi:

- ❖ sinflar diagrammasidagi sinflar va ob'ektlar o'rtasida;
- ❖ paketlar orasida;
- ❖ foydalanish holatlari orasida;
- ❖ boshqalar.

Bu yerda biz 12.7- jadvalda keltirilgan birinchi guruh bog'liqliklarini ko'rib chiqamiz.

12.7- jadval. Sinf diagrammasidagi standart tobelik qat'iy turlari

Qat'iy turlar	Tavsifi
call	2 ta amal orasidagi tobelikni ko'rsatadi – tobe sinf amali erkin sinf amalini chaqiradi
derive	“... bo'yicha hisoblash mumkin” ma'nosida ishlatiladi. Ushbu qat'iy turlardagi tobelik nafaqat sinflarga, modeldagi boshqa elementlar – atributlar, uyushmalar va boshqalarga qo'llaniladi. Buning mohiyati shundan iboratki, tobe elementni erkin elementdagi ma'lumot orqali tiklash mumkinligini anglatadi. Ushbu tobelik shuni ko'satadiki, tobe element umuman olganda ortiqcha, modelga esa u qulaylik, yaqqollik va x.k. nuqtayi nazaridan kiritilgan.

instanceOf	Tobe ob'yekt yoki sinf, erkin sinf (metasinf)ning nusxasi ekanligini ko'rsatadi.
instantiate	Erkin sinf amallari, tobe sinf nusxalarini yaratishini ko'rsatadi.
povertype	Tobe sinf nusxalari, erkin sinf qism-sinflaridan iborat ekanligini ko'rsatadi. Ushbu holatda, tobe sinfning metasinfligi kelib chiqadi.
refine	Tobe sinf, erkin sinfni aniqlashini ko'rsatadi. Ushbu tobelik, o'zaro bo'g'langan sinflar konseptual jihatdan ustma-ust tushushini, lekin mavhumlikning turli darajasida turishini ko'rsatadi.

Endi, amalga oshirish munosabatlarini ko'rib chiqamiz. Interfeyslar va boshqa sinflorvchilar, xususan, sinflar o'rasida sinf diagrammasida ikkita munosabat qo'llaniladi:

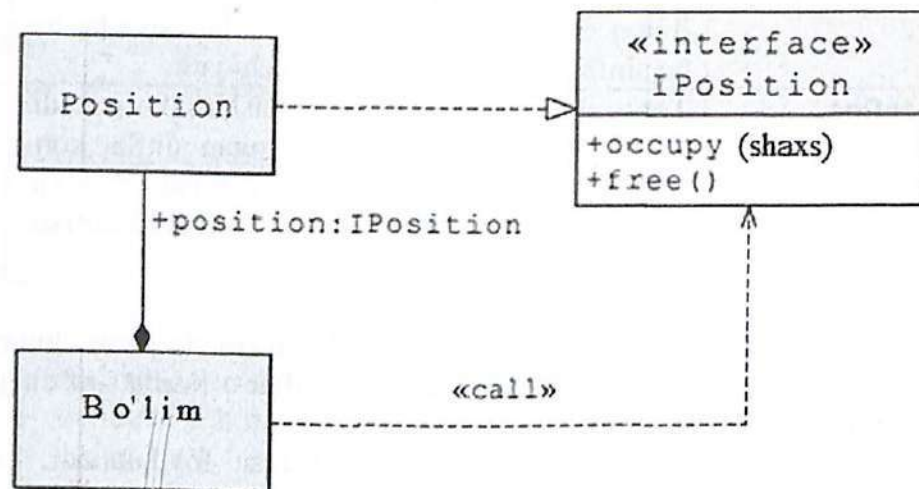
- sinflorvchi (xususan, sinf) interfeysidan foydalanadi, bu tobelik qat'iy turi «call» yordamida ko'rsatiladi;
- sinflorvchi (xususan, sinf) interfeysni amalga oshiradi, bu amalga oshirish munosabati yordamida ko'rsatiladi.

Amalga oshirish munosabatlaridan foydalanishda hech qanday cheklovlar yo'q. Sinf ko'plab interfeyslarni amalga oshirishi mumkin va aksincha, interfeys ko'plab sinflar tomonidan amalga oshirilishi mumkin. «call» tobelik qat'iy turidagi bog'liqlikdan foydalanishda ham hech qanday cheklovlar yo'q, har qanday ko'rinish chegarasidagi interfeyslarning har qanday amallarini chaqirishi mumkin. «call» tobelik qat'iy turidagi mazmun juda oddiy – bu bog'liqlik, erkin qutbda joylashgan sinfning amallarida tobe qutbda joylashgan sinfning amallari chaqirilishini ko'rsatadi.

Keling, “Xodimlar bo'limi” axborot tizimidan bir misolni ko'rib chiqamiz.

Faraz qilaylik, *Bo'lim* sinfi kadrlar harakati bilan bog'liq amallarni amalga oshirish uchun *Position (lavozim)* sinfning lavozimni egallash va bo'shatish imkonini beruvchi amallaridan

foydalanadi, Bo'lim sinfiga boshqa amallar kerak emas. Buning uchun 12.6-rasmda ko'rsatilganidek, tegishli IPosition interfeysini aniqlashi va uni ushbu sinflar bilan munosabatlarini bog'lash mumkin.



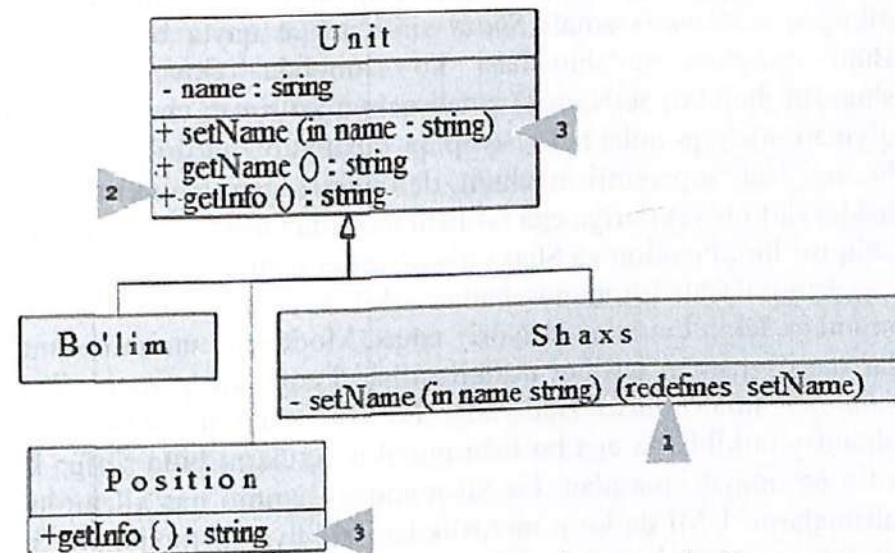
12.6- rasm. Amalga oshirish munosabati va interfeyslardan foydalanish

### Umumlashtirish munosabati

Umumlashtirish munosabati ko'pincha sinf diagrammasida qo'llaniladi. Bitta tizimdagi sinflar o'rtasida umumiylik bo'lmagan vaziyatni tasavvur qilish qiyin. Qoida bo'yicha, umumiylik bor va bu umumiylikni alohida sinfga ajratish tavsiya etiladi. Bunday holda, supersinfda to'plangan umumiy tarkiblar avtomatik ravishda qism-sinflar tomonidan meros qilib olinadi.

Shunday qilib, tavsiflarning umumiy soni kamayadi, ya'ni xato qilish ehtimoli kamayadi. Umumlashtirishlardan foydalanish tizim loyihachisining erkinligini cheklamaydi, chunki meros bo'lib qolgan tarkiblarni qism-sinflarga qayta aniqlash mumkin. Qism-sinfda tarkib qayta aniqlanganligini ko'rsatish uchun UML

2 da paydo bo'lgan qayta ta'mirlash (redefines) qo'shimchasidan foydalanish kerak (12.7- rasm). Oshkormas ravishda, umumlashmalar almashtiriluvchi (substitutativ), ya'ni almashtirish tamoyilini qondiruvchi sifatida qaraladi.



12.7- rasm. Umumlashtirish munosabati

*Misol.* Korxonaning har bir tarkibiy bo'linmasi (bo'lim, lavozim) o'z nomiga ega bo'lishi kerak. "Xodimlar bo'limi" axborot tizimida biz *Position*, *Bo'lim* va *Xodim (Shaxs)* sinflarini aniqladik. Ushbu sinflarning barchasi uchun ob'yektning o'z nomini o'z ichiga olgan atribut ko'rsatilishi mumkin, bu uni bir xildagi turlardan ajratib turadi. Oddiylik uchun bunday atribut String turiga ega, deb faraz qilamiz. Bunday holda, berilgan atributni saqlash va u bilan ishlash uchun mas'ul bo'lgan supersinfni aniqlash mumkin, boshqa sinflarni esa umumlashtirish munosabati orqali supersinf bilan bog'lash mumkin. Biroq, xodimlar bo'limi sohasini chuqurroq tahlil qilish shuni ko'rsatadiki, ajratilgan sinflarning o'z nomi bilan ishlashi mutlaqo bir xil tarzda amalga oshirilmaydi. Darhaqiqat, bo'linmalar va lavozimlarga o'z

nomlarini tayinlash va o'zgartirish "xodimlar bo'limi" axborot tizimining mas'uliyati doirasida, ammo xodimning o'z ismini tayinlash (ayniqsa, o'zgartirish) bu chegaradan tashqariga chiqadi. Ana shu mulohazalardan kelib chiqib, biz 12.7- rasmda keltirilgan umumlashmalar tuzilishini keltirib chiqaramiz. *Unit* sinfida e'lon qilingan *setName()* amali *Shaxs* sinfi uchun qayta belgilangan. Buni *redefines* qo'shimchasi ko'rsatmoqda. Qayta aniqlash shundan iboratki, *setName()* amali uchun ko'rinish chegarasining qiymati «ochiq» holatdan – «yopiq» holatga o'zgartirildi. E'tibor bering, *Unit* supersinfi mavhum, deb aniqlangan, chunki tizimda ushbu sinf ob'yektlariga ega bo'lishi nazarda tutilmagan. Nusxalar aniq Bo'lim, Position va Shaxs qism-sinflar uchun mavjud.

Umumlashtirish munosabatlari UMLda juda erkin o'rnatilishi mumkin, lekin butunlay ixtiyoriy emas. Modeldagi umumlashmalar qat'iy qisman tartibni tashkil qilishi kerak. **Ko'p meroslikda** (*multiple inheritance*) ziddiyatlar bo'lishi mumkin, supersinflar shunday tarkiblarga ega bo'lishi mumkinki, ularni bitta sinfga kiritib bo'lmaydi, masalan, bir xil nomdagi, ammo har xil turdagi atributlarni. UMLda ko'p meroslikdagi ziddiyatlar modelning izchillik qoidalarini buzish, deb hisoblanadi. Agarda modelda ziddiyatlar bo'lmasa, UMLda ko'p meroslik nafaqat taqiqlanmaydi, balki rag'batlantiriladi. Xususan, standartdagi metamodellarga anchagina ko'p meroslik misollari keltirilgan.

### 12.6. Xulosa

1. Ob'yektga yo'naltirilgan namunada dasturlash dasturni bajarish jarayonida, dasturiy ta'minot ob'yektlari xabar almashish orqali bir-biri bilan o'zaro harakatlanadi.
2. Axborotning eng keng tarqalgan turi – bu bir sinf ob'yekti usulidan boshqa sinf ob'yekti usulini chaqirish.
3. UMLda aloqalar tuzilishini modellashtirish uchun sinf diagrammasidagi yushma munosabatlari qo'llaniladi.

4. Sinflovchilar murakkab tizimning tuzilishi darajasida tavsiflanadi.
5. Sinf diagrammalari sinflarning tuzilishini va ular o'rtasidagi munosabatlarni modellashtiradi.
6. Sinflar ko'rilayotgan sohani tahlili, model elementlari va umumiy nazariylardagi mulohazalarning o'zaro mos kelishi asosida tanlanadi.
7. Sinflar o'rtasidagi munosabatlar, birinchi navbatda, umumlashtirish va uyushma munosabatlaridan foydalanish bilan tavsiflanadi. Tobelik va amalga oshirish munosabatlari yordamida esa, kamroq tavsiflanadi.

### 12.7. Nazorat sabollari

1. Modellashtirishdagi tuzilmalarning turlarini ayting.
2. Sinflovchining asosiy vazifasi nimadan iborat?
3. Sinflovchining turlarini ayting.
4. Sinflovchilarning ettita eng muhim xususiyatini tasvirlab bering.
5. Karralik qanday ifodalanadi?
6. UML – modellashtirish uchun sinflovchining hechta turlarni taklif qiladi?
7. Nusxaning harakatlanish sohasi nimadan iborat?
8. Sinfni UMLda tasvirlashdagi standart nechta bo'limdan iborat?
9. Sinflardagi standart qat'iy turlarning soni nechta?
10. Sinf diagrammasining standartdagi tobelik bo'yich qat'iy turlarni ayting.
11. Umumlashtirish munosabatining mohiyati nimadan iborat?

### XIII- BOB. UYUSHMALAR

#### 13.1. Uyushmalar va ularning qo'shimchalari

Uyushma munosabati, sinf diagrammasida eng muhim, deb hisoblansa kerak. Umuman olganda, sinflarni bog'lovchi belgilanishi uzluksiz chiziqdan iborat uyushma – bir sinfning nusxalari boshqa sinf nusxalari bilan bog'liqligini anglatadi. Nusxalar ko'p bo'lishi va har biri bir nechta nusxalar bilan bo'g'lanishi mumkinligi sababli, uyushma bir-biriga bog'langan ob'yektlarning ko'p termalarini tavsiflovchi tasnifchi, ekanligi aniq. UMLda uyushma sinflovchi bo'lib, uning nusxalari *bog'lovchilar (aloqachilar)*, deb ataladi.

*Aloqa (link)* — bu uyushmaning (yoki ulagichning) nusxasidir, bu uyushma qutblaridagi sinflovchilarning nusxalariga havo-lalarning tartiblangan termasidan (kortej, yozuv) iborat.

Yuqorida aytib o'tilganidek, uyushmaning asosiy belgisi uz-lik-siz chiziq bog'lani uchun, u uyushma sinflaridagi ob'yektlar, ish vaqtida o'zaro harakat qilishi mumkinligini ko'rsatadi. Ammo bu uyushma munosabatlari yordamida modellashtirilishi mumkin bo'lgan narsalarning faqat kichik bir qismidir. Uyushma uchun UMLda eng ko'p turli xil qo'shimchalar keltirilgan. Bu qo'shim-chalar, odatdagidek, majburiy emas, kerak bo'lganda, turli vaziyatlarda turli yo'llar bilan qo'llash mumkin. Agar biz bir vaqtning o'zida barcha qo'shimchalardan foydalansak, diagramma shunchalik haddan tashqari yuklanadiki, uni o'qish qiyinlashadi. Shunday qilib, uyushma uchun quyidagi qo'shimchalar aniqlangan:

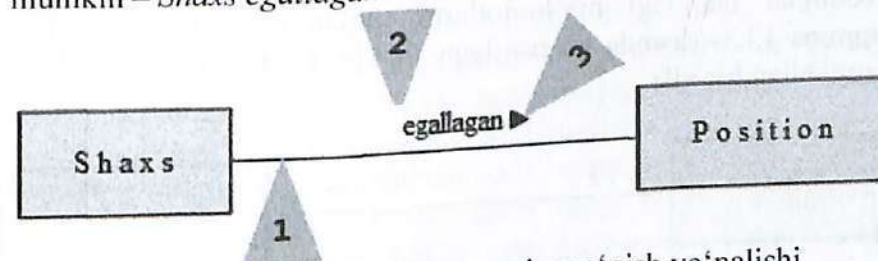
- uyushma nomi;
- uyushma qutbining karraligi;
- yig'ish va kompozitsiya;

- uyushma qutbi uchun yo'nalishni tanlash imkoniyati;
  - uyushma qutbining roli;
  - uyushma sinfi;
  - uyushma qutbini malakalovchisi.
- Ularni tartib bilan ko'rib chiqamiz.

#### 13.2. Uuyushma nomi. Uyushma qutbining karraligi

Uuyushma nomi matn tarzida yuqori (yoki ostki yoki yon) qator sifatida ko'rsatiladi. Nom qo'shimcha ma'noga ega emas va shunchaki modeldagi uyushmalarni farqlash imkonini beradi. Nom odatda, uyushmalarning ko'p qutbli holatlarida yoki bir xil sinflar guruhi bir nechta turli uyushmalar bog'langanda ko'rsati-ladi. Biroq, bu holatlar uchun qat'iy qoida yo'q.

Misol uchun, agar "Xodimlar bo'limi" axborot tizimida xodim lavozimda o'tirgan bo'lsa, u holda *Shaxs* va *Position* sinflarining nusxalari o'zaro bog'langan bo'lishi kerak, yani sinflar orasida uyushma munosabati (1) va uni tayinlashdagi tushuntiruvchi nom bo'lishi kerak. Qo'shimcha ravishda uyushma nomini o'qish uchun yo'nalishni ko'rsatish mumkin (3). 13.1-rasmdagi grafik modelning bo'lagini, tovushni chiqarib o'qish mumkin – *Shaxs egallagan Position (lavozim)*.

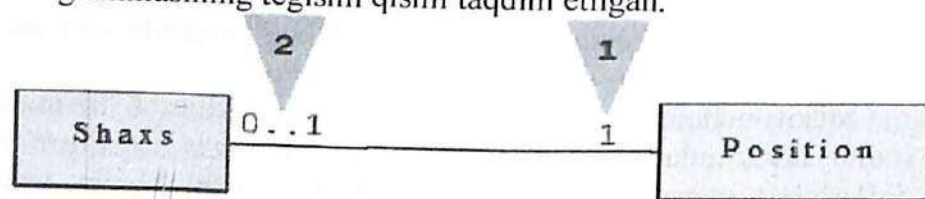


13.1- rasm. Uyushma nomi va o'qish yo'nalishi

*Uyushma qutbining karraligi*, shu sinfdagi qancha ob'yekt, shu qutb tomonidan aloqada ishtirok etayotganini bildiradi. Karralik aniq bir son sifatida ko'rsatilishi mumkin, u holda

berilgan qutb tomonidagi har bir bog'lanishda ko'rsatilgan sondagi ob'yektlar ishtirok etadi. Keng tarqalgan holat, bu karralik mumkin bo'lgan qiymatlar oralig'i sifatida ko'rsatilishi, bunda aloqada ishtirok etuvchi ob'yektlar soni belgilangan oraliqda bo'lishi kerak. Karralikni ko'rsatishda, biz noaniq raqamni bildiruvchi \* belgisidan ham foydalanishimiz mumkin.

Misol uchun, agar "Xodimlar bo'limi" axborot tizimining stavkalarini bo'lish va lavozimlarni birlashtirishni nazarda tutilmagan bo'lsa, u holda ishlaydigan xodimga bitta (1) lavozim to'g'ri keladi va lavozim bitta xodimga yoki bitta ham xodimga mos kelmaydi (2) (lavozim bo'sh). 13.2- rasmda UML diagrammasining tegishli qismi taqdim etilgan.



13.2- rasm. Uyushma qutblarining karraligi

Keyinchalik murakkab holatlar ham qutblarning karraligi yordamida osongina modellashtiriladi. Misol uchun, agar biz lavozimlarni birlashtirishni ta'minlamoqchi va hatto ishlamaydigan xodimlar haqidagi ma'lumotlarni saqlamoqchi bo'lsak, diagramma 13.3- rasmda ko'rsatilgan shaklni oladi (\*yozuvi 0..\* yozuvi bilan bir xil).



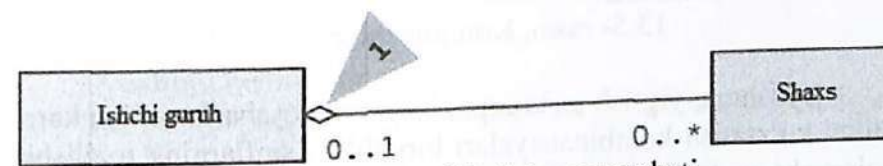
13.3- rasm. Noaniq karralikdan foydalanish

### 13.3. Yig'ish va kompozitsiya

UML uyushma munosabatlarining ikkita alohida, ammo juda

muhim holatidan foydalanadi, ular *yig'ish* va *kompozitsiya*, deb ataladi. Ikkala holatda ham biz "qism – butun" munosabatlarini modellashtirish haqida gapiramiz. Ko'rinib turibdiki, bu turdagi munosabatlar uyushma munosabatlariga tegishli bo'lishi kerak, chunki qismlar va butun odatda o'zaro harakatlanadi.

Yig'ish – bu A (qism) va B (butun) sinf o'rasidagi uyushma (bog'liqlik) bo'lib, A sinfining (bir yoki bir nechta) nusxalari B sinf nusxalariga kiritilganligini anglatadi. Bu maxsus qo'shimcha grafik yordamida "butun"ga birlashtirilgan uyushma qutbida bo'yalmagan romb tarzida tasvirlanadi (1). Masalan, 13.4- rasmda xodim ishchi guruh (Workgroup) a'zosi ekanligi ko'rsatilgan.



13.4- rasm. Yig'ish munosabati

Bunday holda, qo'shimcha cheklovlar qo'yilmaydi. *Shaxs* sinfining nusxasi (qismi) boshqa ob'yektlar bilan bog'lanishi mumkin (ya'ni, *Shaxs* sinfi bir nechta yig'ilishlarda ishtirok etishi mumkin), "*ishchi guruh*" sinfining (butun) nusxalaridan mustaqil ravishda yaratilishi va yo'q qilinishi mumkin.

Kompozitsiya – bu A sinf (qism) va B sinf (butun) o'rasidagi uyushma (bog'liqlik) bo'lib, u agregatsiyaga nisbatan yanada kuchliroq cheklovlarni qo'yadi – kompozitsion jihatdan A qism faqat bitta butun B ga kirishi mumkin, qism faqat butun mavjud bo'lganda mavjud bo'ladi va butun bilan birga mavjudligini to'xtatadi. Biroq qismni, butun yoq qilingunga qadar, undan ajratib olish mumkin. Bunday holda, kompozitsiya buziladi.

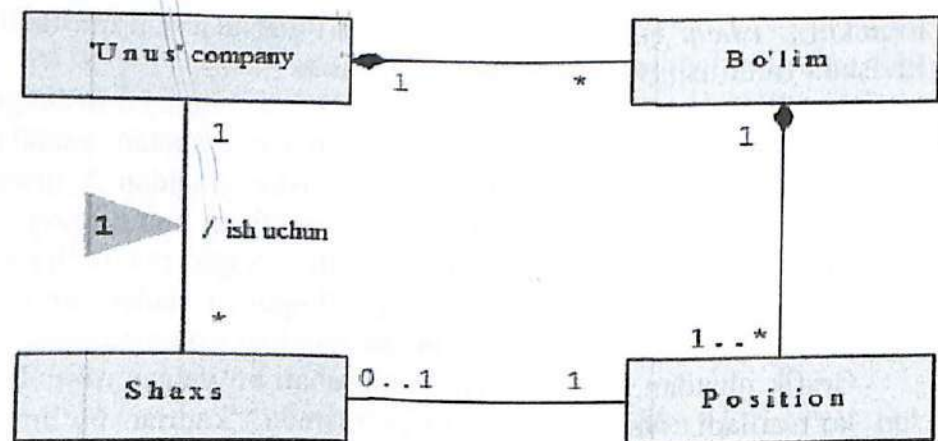
Grafik jihatdan kompozitsiya munosabati bo'yalgan romb bilan ko'rsatiladi. Misol uchun 13.5- rasmda "kadrlar bo'limi" axborot tizimidagi ishchi guruhlar va xodimlar o'rasidagi munosabatlariga yana bir nazar tashlangan.

Bunday holda, tashkilot qat'iy ("armiya") tuzilmasini qabul qilganligi – har bir xodim aniq bitta ishchi guruhga kiradi va har bir ishchi guruhda kamida bitta xodim bo'ladi. Ushbu tuzilmani modellashtirish uchun kompozitsiyadan foydalaniladi. Agar tuzilma mo'rt bo'lsa: "havoda osilgan" xodimlar, "bo'sh" ishchi guruhlar va boshqalar mavjud bo'lsa, unda bunday tuzilmani modellashtirish uchun yanada mos vosita bu yig'ishdir (13.5-rasmga qarang).



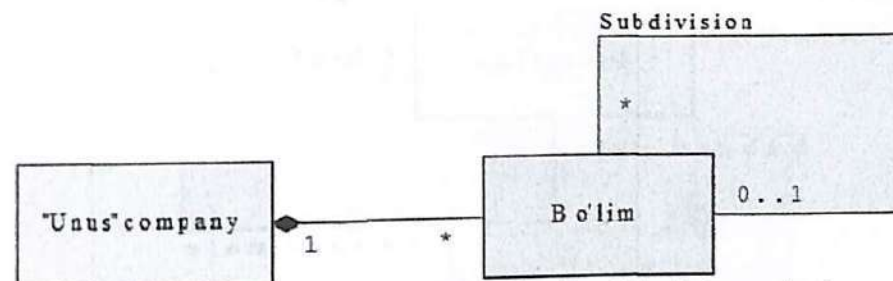
13.5- rasm. Kompozitsiya munosabati

Uyushma, yig'ish va kompozitsiya munosabatlari bilan karralikni ko'rsatish kombinatsiyalari birgalikda sinflarning tuzilishini qisqacha va to'liq ko'rsatishga imkon beradi – nimadan iborat va u qanday bog'langanligini tasvirlaydi. 13.6- rasmda "Xodimlar bo'limi" axborot tizimi uchun bunday tuzilmaning naqlaridan biriga misol keltirilgan.



13.6- rasm. "Xodimlar bo'limi" axborot tizimi sinflari o'rasidagi aloqalarning tuzilishi

Faraz qilaylik, "Xodimlar bo'limi" axborot tizimi korxonaga bo'linmalarining shajaraviy tuzilmasini qo'llab-quvvatlashi kerak bo'lsin. Bu talabni hisobga olgan yechim 13.7- rasmga keltirilgan.



13.7- rasm. Kompozitsiyadan foydalanishga misol

### Uyushma qutbining roli

Rol (role) – bu shu uyushmadagi sinflovchi taqdim etadigan interfeysdir.

**Uyushma qutbi** – bu uyushma chizig'ining sinf to'rtburchagi bilan tutashgan nuqtasidir. Aynan shu nuqta yaqinida uyushma qutblariga ko'plab qo'shimchalar joylashadi.

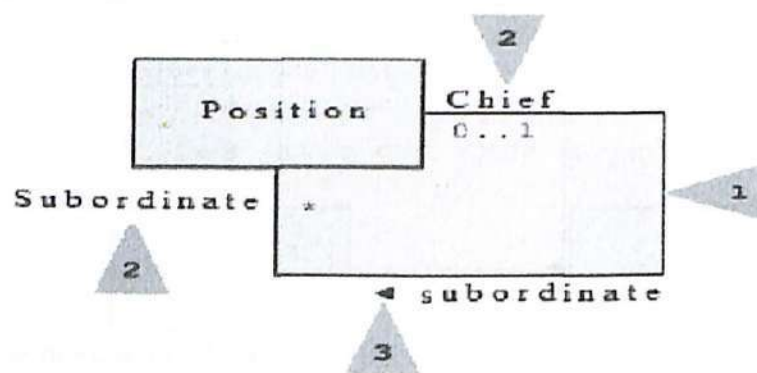
Uyushma qutbining roli, shuningdek, interfeysni tasniflagichi, deb ham ataladi, bu (shu uyushma qutbiga ulangan) tasniflagichning uyushmada qanday ishtirok etishini aniq ko'rsatishning bitta usuli.

Ushbu qo'shimchaning yozuvi – uyushma qutbida ko'rsatilgan matndan iborat. Umuman holda, uyushma qutbining roli quyidagi sintaksisga ega:

### ko'rinish chegarasi NOMI : turi

Ism majburiydir, u rol nomi bilan ataladi, bu aslida qutblarni farqlash imkonini beruvchi uyushma qutbining nomidir. Agar ikki xil sinfni bog'laydigan bitta uyushma ko'rib chiqilsa, unda rol nomlarining hojati yo'q, chunki uyushma qutblarini ular birlashtirilgan sinf nomlari bilan osongina ajratish mumkin. Biroq, agar bunday bo'lmasa, ya'ni agar ikkita sinf bir nechta uyushmalar bilan bog'langan bo'lsa, yoki uyushma sinfni o'zi bilano'zini

bog'lasa, u holda albatta uyushma qutblarining rollarini ko'rsatish kerak. Yana bir misol keltiramiz (13.8- rasmga qarang).

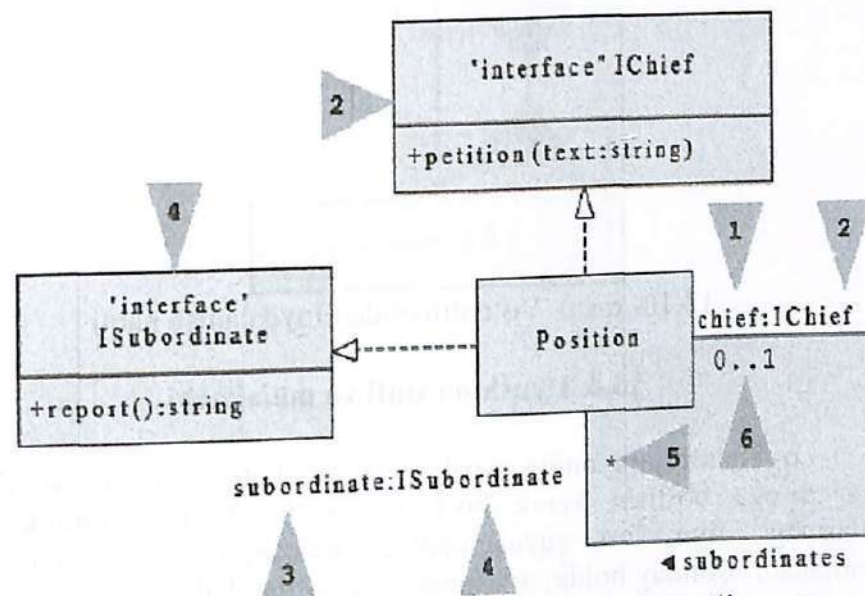


13.8- rasm. Lavozimlar shajarasining tavsifi

13.8- rasmda *Position* sinfining o'zi bilan o'zining aloqasi ko'rsatilgan (1). Uyushma qutblarida rollar ko'rsatilgan (2). O'qish yo'nalishini ko'rsatadigan belgi (3) (qora uchburchak) ushbu uyushmani – “Chief subordinates Subordinate” kabi o'qish imkonini beradil. Ushbu uyushma tashkilotdagi lavozimlarning bo'ysunish shajarasi mavjudligini aks ettirish uchun mo'ljallangan. Biroq, 13.8- rasmdan, faqat *Shaxs* sinfidagi ob'yektlar ma'lum bir shajarani tashkil etishi aniq, ya'ni har bir ob'yekt shajaradagi bir nechta asosiy ob'yektlar va ko'pi bilan bitta yuqori ob'yekt bilan bog'liqligini, lekin boshqa hech narsani bildirmaydi.

Rollardan va shu bilan birga amalga oshirish munosabatlari-dan foydalanib, “xodimlar bo'limi” axborot tizimidagi bo'ysunishni juda ixcham, lekin aniq tasvirlash mumkin. Masalan, 13.9- rasmda ko'rsatilishicha, bo'ysunish shajarasida har bir lavozim ikkita rol o'ynashi mumkin. Bir tomondan, lavozim bosh (1) (chief), deb hisoblanishi mumkin, bu holda u *ICchief* (2) boshliqqa xizmat xati (*petitsiya*)ni taqdim etadi (lavozim boshligi eslatmasini topshirishi mumkin). Boshqa tomondan, lavozimni bo'ysu-

nuvchi (3) (subordinate), deb hisoblash mumkin, bu holda u bo'ysunuvchidan hisobot talab qilinishi mumkin bo'lgan *report* amaliga ega *ISubordinate* (4) interfeysini taqdim etadi. Rahbarning qo'l ostida ixtiyoriy sondagi bo'ysunuvchilar (5) ishlasi mumkin, shu jumladan bitta ham bo'lmaslihi mumkin, bo'ysunuvchida bittadan ortiq boshliq bo'lmasligi mumkin (6).

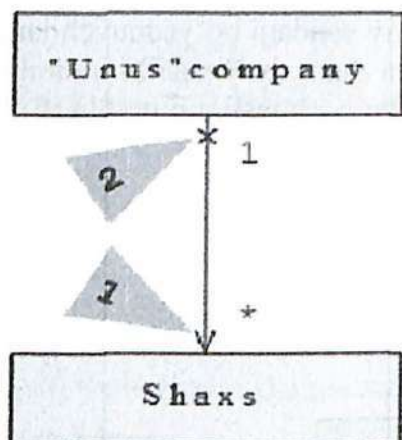


13.9- rasm. Uyushma qutblarining roli

*Uyushma qutbi uchun yo'naltirish qobiliyati* – bu Boolean turidagi qiymatga ega bo'lgan qutb xossasi bo'lib, ushbu uyushma qutbga biriktirilgan holda, ushbu uyushma yordamida sinf ob'yektlariga samarali kirish mumkinligini aniqlaydi.

Muayyan uyushma qutbi uchun yo'naltirish imkoniyatini akslantirish mumkin yoki mumkin emasligini ko'rsatish uchun quyidagi belgilash qo'llaniladi – agar ma'lum bir qutb uchun yo'naltirish mumkin bo'lsa, u holda bu qutb uyushma chizig'ining (1) oxiri mil bilan belgilanadi, agar yo'naltirish mumkin bo'lmasa, unda uyushma chizig'ining oxirida qiya xoch (2) chiziladi. 13.10-

rasmda keltirilgan misolda, yo'naltirish faqat *Kompaniyadan Shaxsgacha* bo'lgan yo'nalishda mumkin, lekin aksincha emas.



13.10- rasm. Yo'naltirishdan foydalanish naqli

### 13.4. Uyushma sinfi va malakachi

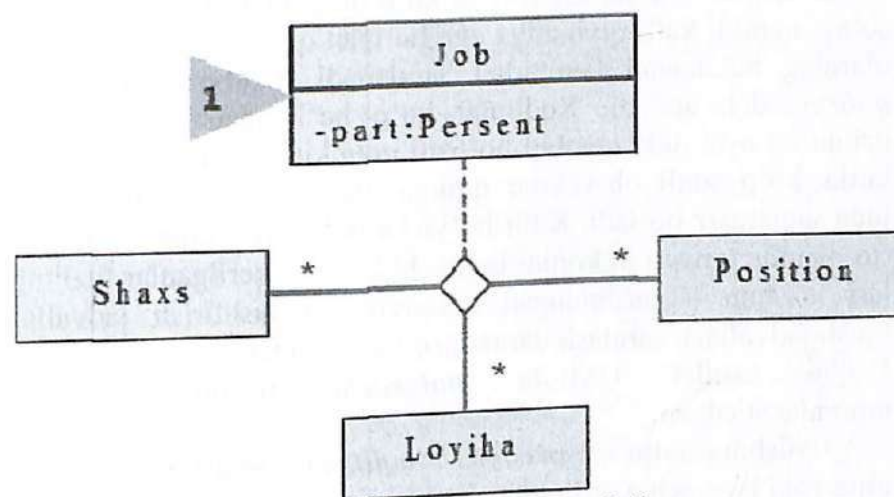
Loyihalash jarayonida uyushma o'z atributlari va hatto amallariga ega bo'lishi kerak bo'lgan holatlar bo'lishi mumkin, ularning qiymatlari, uyushmaning nusxalari – aloqalarda saqlanadi. Bunday holda, uyushma sinfi qo'llaniladi.

**Uyushma sinfi** – uyushma bo'lgan mohiyatdan iborat bo'lib, o'z tarkibida sinfnning tarkibiy qismlarini ham saqlaydi.

Xodim turli loyihalarda bir nechta lavozimlarda ishlashi mumkin bo'lgan vaziyatga ruxsat beriladi va bitta loyihada bir xil lavozimni bir nechta xodimlar egallashi mumkin (stavkani bo'lish). Ish haqi miqdori har bir xodimning ushbu loyihada ma'lum bir lavozimda qancha vaqt ishlaganiga bog'liq.

Shunday qilib, nafaqat lavozimlarni birlashtirishga ruxsat beriladi (bitta xodim turli loyihalarda bir nechta lavozimlarda ishlashi mumkin), balki stavkalarni bo'lish (bitta lavozimni bir nechta yarim, chorak va h.k. stavkalik bir nechta xodimlar egallashi mumkin).

Uyushmaning qismlarga ajratilgan belgilanishidan foydalanib, biz *Shaxs*, *Position* va *Loyiha* sinflari o'rasida "ko'pdan ko'pga" aloqali uyushma mavjudligini aytishimiz mumkin. Biroq, bu etarli emas, ma'lum bir xodimbibg ushbu lavozimning qaysi qismini egallashini ko'rsatish kerak. Ushbu ma'lumotni na lavozimga, na xodimga, na loyihaga bog'lash mumkin emas - bu ularning barchasini bog'laydigan uyushmaning atributidir. 13.11-rasmda ushbu muammoni hal qilish uchun uyushma sinfidan (1) foydalanish usuli ko'rsatilgan.



13.11- rasm. Uyushma sinfi

Ay'na bir bor *Company* va *Shaxs* sinflari o'rtasidagi munosabatlarning o'ziga xosligiga e'tibor qaratamiz. 13.10- rasmda ular uyushma qutblarining "birdan-ko'pga" karraligi bilan bog'liqligi ko'rsatilgan. Bunday uyushma, "bir" karrali qutbda joylashgan *Company* sinfining har bir nusxasini, "ko'p" (ba'zan katta) karralik qutbda joylashgan *Shaxs* sinfidagi ob'yektlar to'plamini etkazib beradi. Biroq, ko'pincha *Shaxs* bilan bog'liq ob'yektlarning to'liq to'plamini emas, balki kichik qism-to'plamini, ko'pincha bitta aniq ob'yektni olishimiz kerak bo'lgan

holatlar yuzaga keladi. To'plamdan bitta aniq ob'yektni ajratish uchun biz ushbu ob'yektni noyob tarzda aniqlaydigan berilganlarga ega bo'lishingiz kerak. Bunday berilganlar odatda *kalit*, deb ataladi.

Masalan, "Xodimlar bo'limi" axborot tizimidagi xodim uchun shaxsiy soliq to'lovchi raqami kalit, deb hisoblanishi mumkin.

Muayyan xodimni topish muammosini hal qilish uchun (*Shaxs* sinfining o'ziga xos nusxasi) har doim quyidagi oson yechimni qo'llash mumkin – kalitni *Shaxs* sinfi ob'yektining o'zida atribut sifatida saqlash va ko'p ob'yektlarni olganingizdan so'ng, kerakli kalit qiymatiga ega bo'lgan qiymat topilmaguncha, ularning barchasini ketma-ket saralanadi. Ushbu usul *chiziqli qidiruv*, deb ataladi. Xodimlar ko'p bo'lmagan kompaniyalar uchun bu usul juda maqbul bo'lishi mumkin. Ammo boshqa hol-larda, ko'p sonli ob'yektlar qutbga ulanganda, chiziqli qidiruv juda samarasiz bo'ladi. Kalit bo'yicha ob'yektni samarali ajratish (to'plamda topish) imkonini beruvchi ko'plab berilganlar tuzilmalari ma'lum – tartiblangan massivlar, joylashtirish jadvallari (xesh-jadvallar), saralash daraxtlari, tashqi indekslar va boshqalar. Ushbu usullar UMLda *malakachi*, tushunchasi bilan umumlashtirilgan.

Uyushma qutbi *malakachisi (qualifier)* – uyushma qutbining bitta yoki bir nechta atributlari bo'lib, uning qiymati uyushmaning boshqa qutbiga biriktirilgan sinfdagi bir (yoki bir nechta) ob'yektlarni ajratish imkonini beradi.

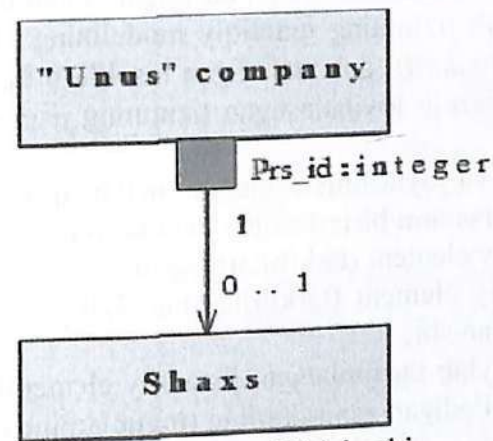
Malakachi, sinf to'rtburchagiga ulashgan uyushma qutbida kichik to'rtburchak sifatida tasvirlanadi.

Ushbu to'rtburchak ichida (yoki uning yonida) malakachi atributlarning nomlari va ehtimol turlari ham ko'rsatilishi mumkin. Uyushmaning malakachi atributining tavsifi oddiy sinf atributining tavsifi bilan bir xil sintaksisga ega, faqat u boshlang'ich qiymatni ololmaydi.

*Malakachining asosiy maqsadi*, uyushmadagi qarama-qarshi qutbning karraligini kamaytirishdir, shuning uchun u asosan "bir-

dan ko'pga" yoki "ko'pdan ko'pga" qutb karraliklariga ega bo'lgan uyushmalarda qo'llaniladi va u qutb yonida "ko'p" karralikdagi qarama-qarshi qutbda turadi.

*Malakachidan* foydalanganda qarama-qarshi qutbning karraligi kamayadi va bu diagrammada akslanadi. 13.10- rasm bilan 13.12- rasmni solishtiring.



13.12- rasm. Malakachi

*Shaxs* sinfining qutb chastotasi \* dan 0..1 gacha o'zgardi, chunki ma'lum bir kalit uchun *Shaxs* sinfining nusxasi topilishi yoki kalit qiymati noto'g'ri bo'lsa, topilmasligi mumkin.

Shunday qilib, agar karralik malakachi qutbiga qarama-qarshi bo'lgan uyushma qutbida berilgan bo'lsa, u holda u bog'lanish qutbiga ulangan ob'yektlar to'plamining ruxsat etilgan quvvatini emas, balki malakachi atributlarining qiymatlarini belgilashda aniqlanadigan qism-to'plamning ruxsat etilgan quvvatini ko'rsatadi.

### 13.5. Amalga oshirish diagrammalari

Ushbu bo'limda biz ikkita amalga oshirish – tarkiblar va joylashtirish diagramlarini ko'rib chiqamiz. Bu diagrammalar amalga

oshirish va yetkazib berish bosqichlarida, ya'ni rivojlanishining keyingi bosqichlarida alohida ahamiyatga ega. Amalga oshirish nuqtayi nazaridan, loyihalashtirilgan tizim hisoblash tugunlari bo'ylab taqsimlangan tarkiblardan iborat (joylashtirish diagrammalarida ko'rsatilgan).

UML 2da UML 1ga nisbatan sezilarli o'zgarish yuz bergan, ya'ni "tarkib" tushunchasi ikkiga bo'lingan: mantiqiy va jismoniy. Mantiqiy tarkib tizimning mantiqiy modelining elementi bo'lsa, *artefakt (sun'iy dalil)*, deb ataladigan hisoblash tugunida joylashgan jismoniy tarkib loyihalashgan tizimning jismoniy elementini ifodalaydi.

Tarkiblar va joylashtirish diagrammalari quyidagi, eng yaqin bog'langan narsalarni birlashtirgan juda ko'p umumiyliklarga ega:

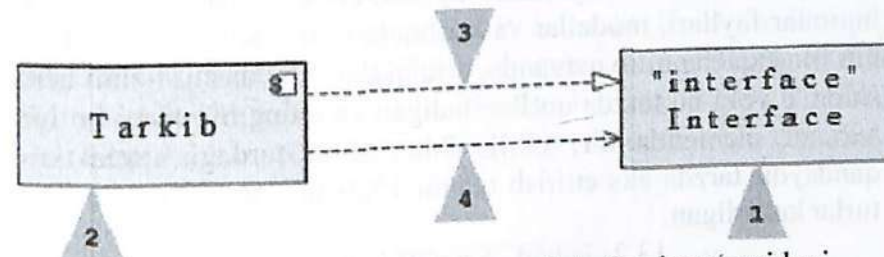
- mantiqiy element (tarkib)larning tuzilishi;
- mantiqiy element (tarkib)larning fizik elementlarga (artefaktlarga) akslanishi;
- ular bo'ylab taqsimlangan jismoniy elementlar (artefaktlar) bilan foydalaniladigan zaxiralarning (tugunlarning) tuzilishi.

### Interfeys

Sinflowchilarga, masalan, tarkiblarga (2) nisbatan interfeyslar (1) ikkita rolni, ya'ni *ta'minlangan va talab qilinuvchi bo'lishi* mumkin (13.13- rasimga qarang). Lekin shuni ta'kidlash kerak-ki, interfeysning o'zi shunchaki shartnomaning tavsifi bo'lib, ta'minlangan va talab qilinuvchi bo'lishi, undan qanday foydalanishga bog'liq. Agarda sinflowchi interfeysni amalga oshirsa, u shu sinflowchi uchun u *ta'minlangan* bo'ladi va bu dalil amalga oshirish munosabati yordamida ko'rsatiladi (13.13- rasmdagi, 3). Agarda sinflowchi interfeys amalini chaqirsa, u holda bu sinflowchi uchun u *talab qilingan* interfeys hisoblanadi va bu dalil qaramlik munosabati yordamida ko'rsatiladi (13.13- rasmdagi, 4).

### Tarkiblar, artefaktlar va tugunlar

*Tarkib* (component) – bu tizimning mantiqiy tasvirining modulli qismi bo'lib, uning o'zaro ta'siri *ta'minlangan va talab qili-*



13.13- rasm. Sinflowchilar va interfeyslar o'rasidagi munosabatlar

*uvchi* interfeyslar termasi bilan tavsiflanadi.

UMLdagi bu tarkib modelning bir qismi bo'lib, faqat loyihalash vaqtida mavjud bo'lgan mantiqiy mohiyatni tavsiflaydi, garchi kelajakda u ish vaqtida bajarilishi bilan jismoniy amalga oshirish (artefakt) bog'liq bo'lishi mumkin. UML standartida tarkiblar uchun 13.1- jadvalda keltirilgan qat'iy turlar taqdim etilgan.

13.1- jadval. Tarkiblarning standart qat'iy turlari

Qat'iy tur	Tavsifi
<b>buildComponent</b>	Ilovalarni yaratish uchun xizmat qiluvchi tarkib
<b>entity</b>	Ma'lum bir sohadagi tushunchani doimiy saqlovchi axborot tarkibi
<b>service</b>	Holatsiz funksional tarkib, so'raluvchi qiymatlarni aniq qaytaruvchi
<b>subsystem</b>	Katta tizimni bolaklash jarayonidagi shajaraviy birlik

Dasturlarni jamlashdagi tarkib ma'nosiga o'xshash, UMLdagi artefakt tushunchasidir, hamma artefaktlar emas, balki faqat uning ba'zi qat'iy turlari o'xshaydi. *Artefakt* – bu dastur tizimidagi sun'iy ravishda yaratilgan har qanday elementdir. Dastur tizimidagi elementlar va demak artefaktlarga – axborotning jismoniy elementlari bo'lgan bajariladigan fayllar, dasturning boshlang'ich kodlari, veb-sahifalar, yordam fayllari, qo'shimcha hujjatlar, ma'lumotlar fayllari, modellar va boshqalarni o'z ichiga olishi mumkin. Boshqacha qilib aytganda, artefaktlar – bu dastur tizimi ishlashida, u yoki bu tarzda qo'llaniladigan va uning bir qismi bo'lgan axborot elementlaridir. UML 2da ushbu turdagi artefaktlarni qandaydir tarzda aks ettirish uchun 13.2- jadvalda standart qat'iy turlar keltirilgan.

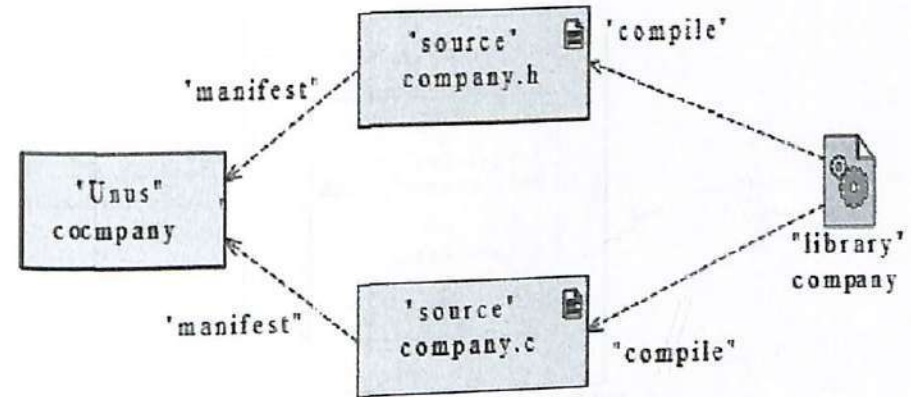
13.2- jadval. Artefaktlarning standart qat'iy turlari

Qat'iy tur	Tavsifi
file	Fayl tizimidagi ixtiyoriy saqlanuvchi fayl
document	Boshlang'ich math fayli, yoki bajariluvchi fayl ham bo'lmagan, hujjat fayldan iborat – artefakt
executable	Hech qanday qat'iy tur ko'rsatilmasa, oshkormas ravishda bajariluvchi ixtiyoriy ko'rinishdagi dastur
library	статик yoki динамик kutubxona
script	Mos dasturiy vositalar bilan sharhlanish imkoniyatini yaratuvchi matndan iborat fayl
source	Dasturning boshlang'ich kodidan iborat fayl

Biroq, haqiqiy artefaktlar yuqorida sanab o'tilganlarga qaraganda ancha xilma-xildir. Ushbu holatni qandaydir tarzda hisobga olish uchun ko'plab vositalar, standart qat'iy turlardan tashqari, artefaktlarning qo'shimcha qat'iy turlarini qo'llab-quvvatlaydi. Buning uchun ko'pincha diagrammalardagi yuqori ko'rinishni ta'minlash uchun maxsus belgi va qiyofalardan foydalaniladi.

UMLda artefakt tushunchasidan foydalanishning eng muhim jihati shundan iboratki, artefakt manifestatsiyada ishtirok etishi mumkin.

*Manifestatsiya* – bu «manifest» qat'iy turiga bog'langan tobe munosabat bo'lib, modelning elementini (masalan, sinf yoki tarkib) bilan uning artefakt shaklida jismoniy amalga oshirilishidagi bog'lovchidir. 13.14- rasmda Company sinfining manifestatsiya munosabati («manifest» qat'iy turi) bilan bo'lanishi keltirilgan. Manifestatsiya munosabati ikkita «source» qat'iy turdagi artefakt bilan bog'langan, ular esa o'z navbatida, bajarish vaqtini belgilovchi artefaktni – Company sinfining dinamik kutubxonasini («library» qat'iy turi) aniqlaydi.



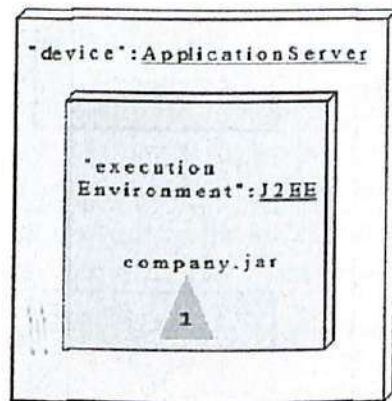
13.14- rasm. Artefaktlar

Umuman olganda, manifestatsiya munosabati – bu "ko'pdan ko'pga" turidagi munosabatdir, modelning bitta elementi ko'plab artefaktlar tomonidan amalga oshirilishi va bitta artefakt modelning ko'plab elementlarini amalga oshirishda ishtirok etishi mumkin. Manifestatsiyani grafik jihatdan amalga oshirilishi mumkin bo'lgan mohiyatdagi artefaktning «manifest» qat'iy turidagi tobe munosabat sifatida tasvirlashadi. Manifestatsiya "ko'pdan ko'pga" turdagi munosabat bo'lganligi sababli, uni to'liq tavsiflash uchun

modelda bir nechta qaramlik munosabatlari talab qilinishi mumkin.

Muhokama qilmoqchi bo'lgan oxirgi mohiyatimiz – bu tugundur. *Tugun (node)* – bu tizimning ishlashida ishtirok etadigan jismoniy hisoblash zaxirasidir.

UML 2 – tugunlar uchun ikkita qat'iy turlar sifatida «execute-onEnvironment» va «device»ni taqdim etadi. «ExecutionEnvironment» qat'iy turiga ega tugun bizga dastur bajariladigan asbob va dasturiy platformani modellashtirish imkonini beradi. «Device» qat'iy turiga ega tugun asbob va dasturiy platformani ham modellashtiradi, lekin 13.15- rasmda ko'rsatilganidek, bir tugunni boshqasiga joylashtirish imkoniyatini ham beradi.

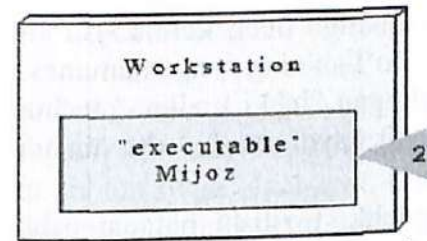
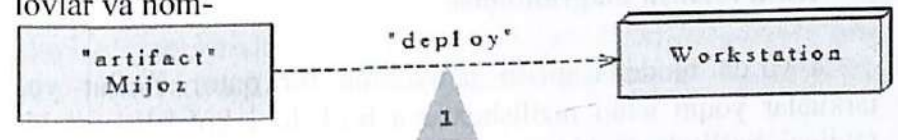


13.15- rasm. Tugun belgisi

Tizimning artefaktlari uning ishlashi davomida tugunlarga joylashtiriladi, bu grafik tarzda ularning tugun ichidagi ro'yxati bilan ifodalanadi (13.15- rasmdagi 1), yoki artefakt va tugun o'rasidagi «deploy» qat'iy turiga tobe munosabat (13.16- rasmdagi 1) yoki tugun tasviri ichidagi artefakt tasviri (13.16- rasmdagi 2). Barcha belgilar tengdir.

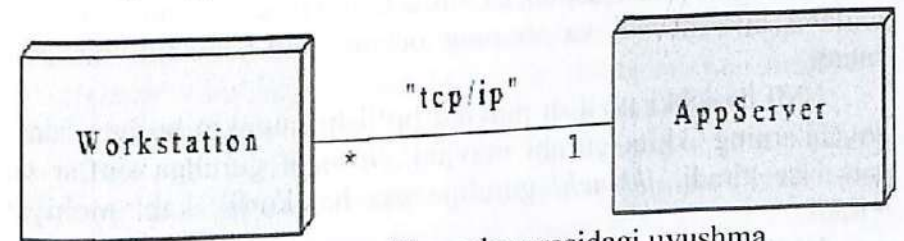
Ushbu – band doirasida biz ko'rib chiqishimiz kerak bo'lgan oxirgi narsa tugunlar o'rasidagi uyushma munosabatidir. Agar tugunlar o'zaro uyushma munosabati bilan bog'langan bo'lsa, bu

boshqa kontekstlarda ham kabi bir xil narsani, ya'ni xabar almashish qobiliyati mavjudligini anglatadi. Tugunlardan tashkil topgan hisoblash tarmoqlariga nisbatan uyushmaning borligi, aloqa kanalining mavjudligini bildiradi. Agar siz kanalning xossalari haqida qo'shimcha ma'lumot berish kerak bo'lsa, buni umumiy mexanizmlar – qat'iy turlar (13.17- rasmdagi «tcp/ip»na), cheklovlar va nom-



13.16- rasm. Tugunda artefaktni joylashtirish belgilari

nomlangan qiymatlar yordamida amalga oshirish mumkin.



13.17- rasm. Tugunlar orasidagi uyushma

### 13.6. Sinflorlardagi rollar va nusxalar darajasida modellashtirish

Yuqorida muhokama qilingan tuzilmaviy diagrammalarda

mohiyatlar asosan sinflovchilardan iborat edi. Sinflovchilarning nusxalari, agar ular paydo bo'lsa, yordamchi rol o'ynaydi. Biroq, modelni batafsilroq, ob'yekt nuqtayi nazardan ko'rib chiqish zarur bo'lgan vaziyatlar mavjud. Ushbu bo'limda bunday hollarda qo'llaniladigan UML vositalarini ko'rib chiqamiz.

### Ichki tuzilish diagrammasi

UMLda modellashtirish jarayonida bir qator sinflar yoki tarkublar yoqin ichki tuzilishga ega bo'lishini ko'rish mumkin. Oldingi bo'limlarda biz, masalan, "butun" va uning "qismlari" o'rasidagi munosabatni ifodalash uchun xizmat qiladigan sinflar o'rasidagi kompozitsiyaning munosabatiga duch keldik. Bu sinf (kompozit) ichki tuzilishga ega bo'lishining eng namunaviy misoli bo'lsa-da, UMLda qabul qilingan "ichki tuzilma" tushunchasining mohiyatini to'liq aks ettirmaydi va bu shu tushunchaning faqat bitta jihati.

Gap shundaki, sinflovchining ichki tuzilishi nafaqat ushbu sinflovchini qandaydir konteyner shaklida taqdim etish qobiliyatini, balki ma'lum bir jumlaning ta'minlash qobiliyatini ham anglatadi, ya'ni ichki tuzilish – bu mantiqiy tushuncha bo'lib, bu ma'lum bir vazifani bajarish uchun izchillik tamoyiliga nisbatan sinflovchilarni birlashtiradi va shuning uchun xatti-harakatni nazarda tutadi.

UMLda ichki tuzilish mavjud bo'lishi mumkin bo'lgan sinflovchilarning ikkita guruhi mavjud. *Birinchi* guruhga sinflar va tarkiblar kiradi, *ikkinchi* guruhga esa hamkorlik kabi mohiyat kiradi.

UMLdagi guruhlarning har biri uchun ichki tuzilishini ifodalash uchun o'z vositalari mavjud va ushbu bo'limning dastlabki ikki bandini ushbu vositalarni muhokama qilishga bag'ishlaymiz. Ichki tuzilma diagrammasi UML 2 ning eng muhim yangiliklaridan biridir. U tuzilmaviy bo'laklash mexanizmini tuzilmaviy sinflovchilarga (sinflar va tarkiblar) kengaytiradi. Amaliy

modellashtirish nuqtayi nazaridan bu juda muhimdir. Diagrammalarda murakkab sinflovchining tarkibiy qismlari qanday o'zaro harakat qilishini to'g'ridan-to'g'ri ko'rsata olmaslik UML 1 da murakkab modellarni tuzishni qiyinlashtirgan. UML 2 da bu cheklov olib tashlandi.

Ichki tuzilish diagrammasi – bu sinflovchining ichki tuzilishini va ushbu tuzilmani tashkil etuvchi elementlar (qismlar) o'rasidagi o'zaro harakat yo'llarini ochib beradigan tuzilmaviy diagramma. Ichki tuzilish diagrammalarida quyidagi asosiy ob'yektlar qo'llaniladi:

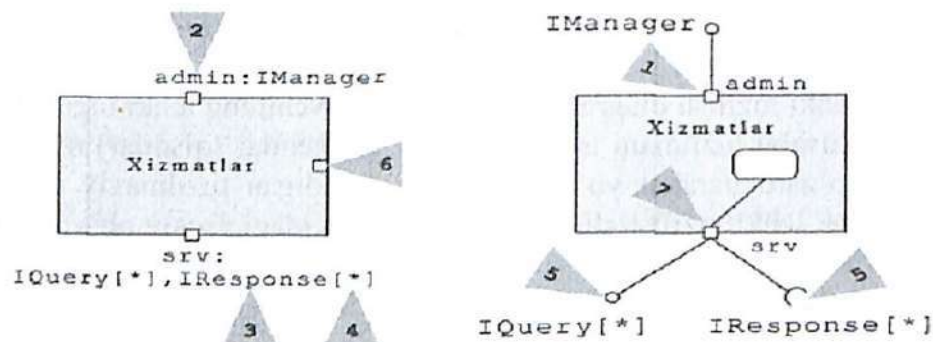
- tuzilmaviy sinflovchi (structured classifier);
- qism (part);
- port (port);
- ulagich (connector).

**Tuzilmaviy sinflovchi** – ichki tuzilishi ichki tuzilish diagrammasi bilan tavsiflangan sinflovchidir (sinf yoki tarkib). Ushbu ta'rif sun'iy ga o'xshaydi, lekin rasmiy ravishda u to'g'ri, chunki tuzilmaviy sinflovchi tushunchasining o'zi sun'iydir. Tuzilmaviy sinflovchi sinflovchilar uchun odatiy bo'lgan to'rtburchaklar shaklida tasvirlangan, uning ichida va chegaralarida boshqa ob'yektlarning qiyofalari va belgilari joylashtirilgan. Odatda, bitta diagrammaning ichki tuzilishi bitta sinflovchining tuzilishini ochib beradi, faqat shu sinflovchining qismlarini ko'rsatadi.

**Qism** – bu tuzilma tarkibi bo'lib, uning nusxasi tuzilmaviy sinflovchining nusxasida o'ynaydigan rolini tavsiflaydi.

**Port** – bu tuzilmaviy sinflovchi va uning qismlarining tashqi mohiyatlar bilan o'zaro harakatidagi, shaxsiy nuqta. Port tuzilmaviy sinflovchi yoki uning qismi (1) chegarasida kichik kvadrat sifatida tasvirlanadi. Portning nomi (2), shuningdek turi (3) va karraligi (4) ko'rsatilishi mumkin. Agar turi ko'rsatilgan bo'lsa, u holda bu port orqali o'zaro aloqa sodir bo'ladigan interfeys bo'lishi kerak. Biroq, tur ko'rsatilmaligi mumkin, lekin bir yoki bir nechta taqdim etilgan va/yoki kerakli interfeyslarni (5) portga aniq ulash mumkin, ular orqali tashqi muhit bilan o'zaro aloqa

amalga oshiriladi. Port mohiyati to'g'risidagi umumiy yozuv 13.18- rasmda keltirilgan.



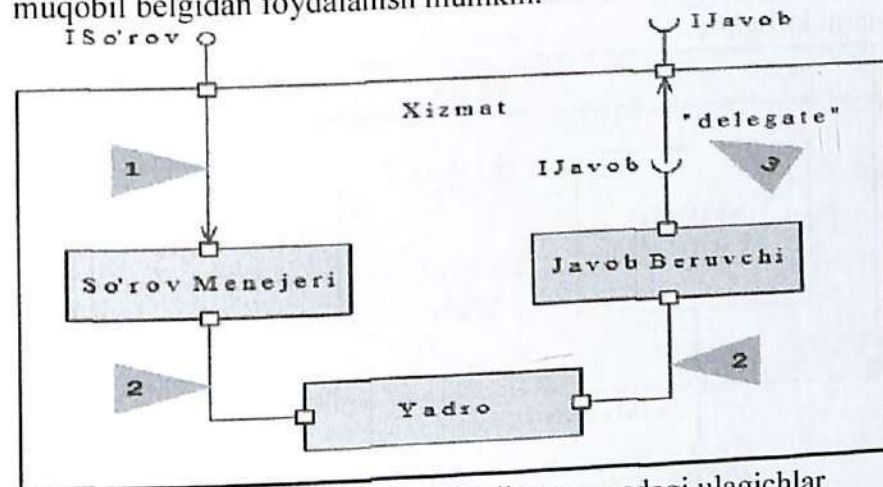
13.18- rasm. Port belgisi

Yuqorida aytib o'tilganidek, tuzilmaviy sinflovchining qismlarini, agar ular qism o'ynagan rolga mos kelsa, boshqa sinflovchilarning barcha nusxalari uchun uyalar, deb hisoblash mumkin. Sinflovchilarning bu nusxalari bir-biri bilan aniq (masalan, uyushmalar) yoki oshkormas (masalan, tobe) munosabatlar orqali bog'lanishi mumkin. Tuzilmaviy sinflovchi doirasida bu munosabatlar ulagichlar shaklida taqdim etiladi.

**Ulagich** tuzilmaviy sinflovchining qismlarini bir-biriga ulash uchun ishlatiladi. Ulagich tuzilmaviy sinflovchi portini uning bir qismiga ulashi yoki ununing qismlarni bir-biriga ulashi mumkin. Bunday holda, qismlarning chegaralaridagi portlar ko'rsatilishi yoki ko'rsatilmaslgi mumkin. Shuni ta'kidlash kerakki, ulagichlar nafaqat ichki tuzilish diagrammalarida, balki hamkorlik diagrammasida ham qo'llaniladi.

**Ulagichlar** ikki xil bo'ladi – vakil va yig'iluvchi. Tuzilmaviy sinflovchi portini uning ichki qismiga bog'laydigan ulagichga **vakil ulagichi** (*delegation connector*), deyiladi. Tuzilmaviy sinflovchining ikkita qismini birlashtiradigan ulagich **yig'iluvchi ulagich** (*assembly connector*), deb ataladi. 13.19- rasmda ulagichlarning ikkala turi ko'rsatilgan: vakil (1) va yig'iluvchi (2). Vakil

ulagichlar uchun «delegate» (3) qat'iy turidan foydalangan holda muqobil belgidan foydalanish mumkin.



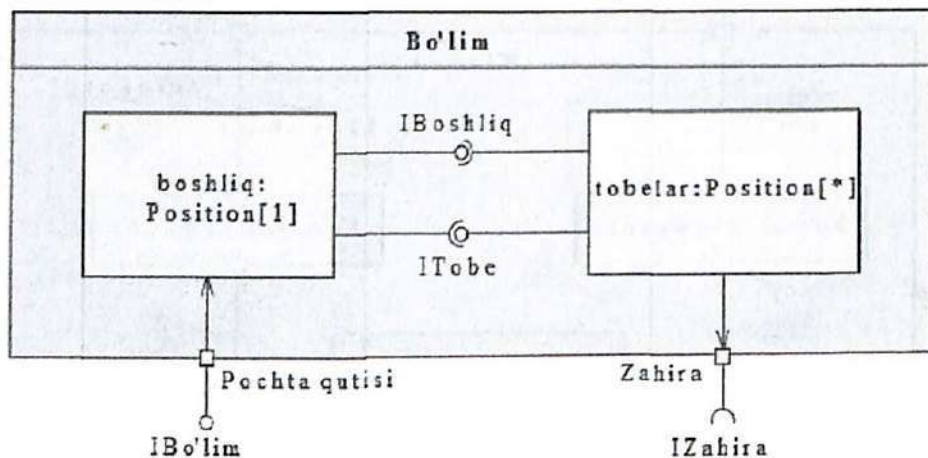
13.19- rasm. Ichki tuzilmaviy diagrammadagi ulagichlar

“Xodimlar bo'limi” axborot tizimi modelidan ichki tuzilmaviy diagramma misolini ko'rib chiqamiz (13.20- rasimga qarang). Ushbu diagrammada biz bo'limda (bo'lim sinfida) bitta boshliq va bir nechta xodimlardan iborat oddiy ichki tuzilmani ajratamiz. Tobelar (subordinates) va boshliq (chief) o'zaro aloqada bo'lib, axborot uzatish yo'nalishiga qarab turli xil o'zaro harakat interfeyslari taqdim etiladi (IChief va ISubordinate). Bundan tashqari, ushbu qismlar tashqi muhit bilan o'z portlari (Inbox va Resource) orqali o'zaro haratlanadi.

### 13.7. Hamkorlik

**Hamkorlik** – bu ichki tuzilishga ega bo'lgan yana bir sinflovchidir. Hamkorlikni sinflovchining bir turi, deb raqash mumkin, uning farqi shundaki, hamkorlik hech qachon o'z qismlariga ega-lik qilmaydi. Qismlar bir-biriga jismoniy kirish orqali emas, balki hamkorlikda ishtirok etish orqali bog'langan. Hamkorlikning ichki tuzilishini aniqlashda tuzilmaviy sinflovchilarni tavsiflash

bilan bir xil mohiyatlar qo'llaniladi, shuning uchun biz oldingi bandning mazmunini takrorlamaymiz, faqat kerakli tushuntirishlarni kiritamiz.



13.20- rasm. Bo'lim sinfining ichki tuzilish diagrammasi

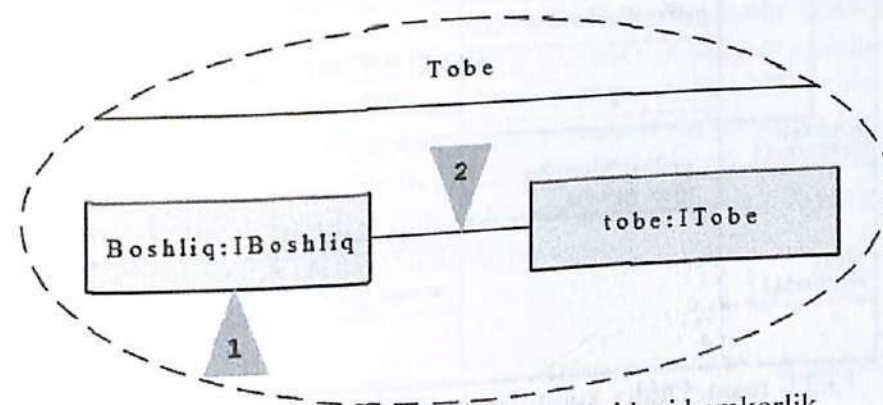
Hamkorlik muammoni hal qilish uchun zarur bo'lgan rollar ko'rinishidagi hamkorlik qiluvchi ishtirokchilar termasini belgilaydi. Har bir alohida holatda, bu rollarni sinflovchilarning o'ziga xos nusxalari o'ynaydi. Masalani yechishdagi rollar orasidagi muhim munosabatlar diagrammada ulagichlar tomonidan ko'rsatiladi va shu bilan kerakli aloqalarni aniqlaydi.

Aniqlik uchun hamkorlikda sinflovchilarning faqat hal qilina-yotgan masala uchun muhim bo'lgan jihatlarini tavsiflash va qolganlarini istisno qilish foydalidir. Natijada, bir xil ishtirokchi bir vaqtning o'zida turli hamkorliklarda turli rollarni o'ynashi mumkin va har bir hamkorlik ushbu ishtirokchining faqat muhim jihatlarini ifodalaydi. Hamkorlik tuzilmasidan alohida uning xatti-harakati, masalan, o'zaro harakat diagrammalari orqali tavsiflanadi.

Yana, "Xodimlar bo'limi" axborot tizimiga murojaat qilaymiz. 13.20- rasmda bo'limning ichki tuzilishi ko'rsatilgan bo'lib, unda boshliq va tobelar o'rasidagi o'zaro munosabatlar tasvirlangan. Keling, boshliqning bir yoki bir nechta o'rinbosarlari bo'lsin,

u orqali u tobelar bilan o'zaro aloqada bo'lgan biroz murakkabroq holatni ko'rib chiqaylik. Agar biz bunday holat uchun ichki tuzilmaning diagrammasini tuzsak, boshliqning undagi o'rinbosarlari bilan o'zaro harakatining tavsifi o'rinbosarlarning tobelar (boshliqning dastlabki diagrammadagi tobelar bilan) o'zaro harakatini tavsifini takrorlaydi.

Hamkorlik bunday ortiqchalikdan qochish imkonini beradi, uning yordamida o'zaro harakat chizmasini bir marta tasvirlash va turli o'zaro harakat qiluvchi qismlarga qo'llash mumkin. Hamkorlik nomini o'z ichiga olgan nuqtali ellips sifatida tasvirlangan.



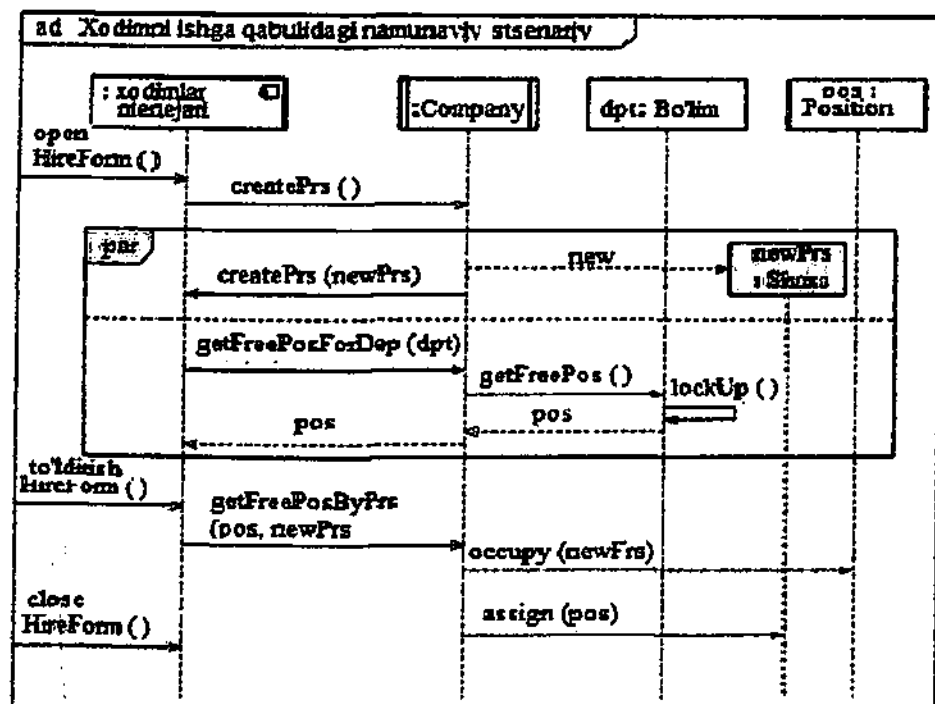
13.21- rasm. Boshliq va tobe o'rasidagi hamkorlik

Rollar (1) va ulagichlar (2) ko'rinishidagi hamkorlikning ichki tuzilishi alohida diagrammada ellips ichida ko'rsatilishi mumkin. Masalan, 13.21- rasmda boshliq va tobelar o'rasidagi munosabatlarni tavsiflovchi hamkorlik ko'rsatilgan.

### 13.8. Loyihalash namunasi

UMLda hamkorlikdan foydalanish loyihalash namunalari kabi tushuncha bilan chambarchas bog'liq bo'lib, ular o'z navbatida naqshlarning bir turi hisoblanadi. Ushbu jihatni o'rganish uchun "Xodimlar bo'limi" axborot tizimini modellashirishga qaytamiz

va foydalanish variantini standart ssenariysini amalga oshiramiz (13.22- rasm).



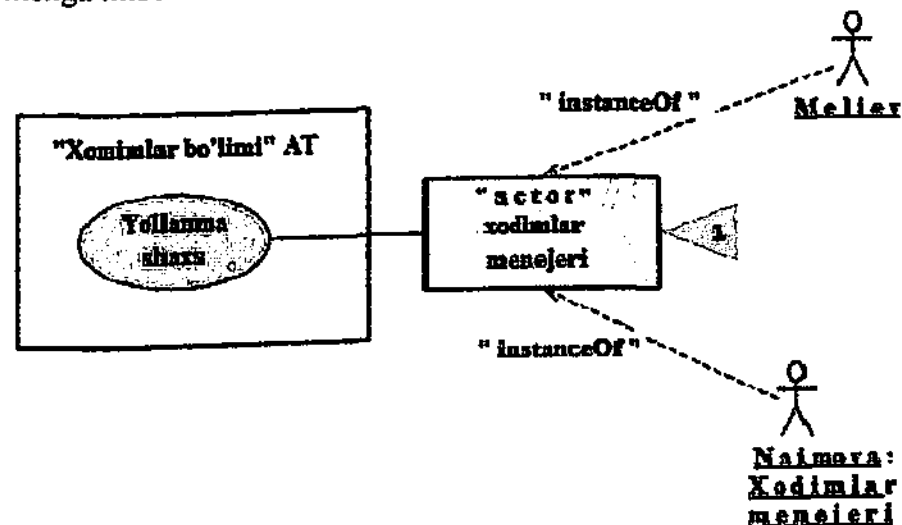
13.22- rasm. Oddiy xodimlarni yollash ssenariysi uchun ketma-ketlik diagrammasi

### 13.9. Sinflovchilarning nusxalari

Oldingi mavzularda biz juda ko'p turli sinflovchilarni ko'rib chiqdik. Biroq, shu bilan birga, biz har doim sinflovchilarning tasnifchi ekanligini ta'kidlaganmiz, ya'ni bir xil turdagi ob'yektlarning tavsifchisi, lekin hech qachon sinflovchilarning nusxalari haqida maqsadli tahlil o'tkazmadik. Endi bu bo'shliqni to'ldirish uchun biz yetarli materiallar to'pladik.

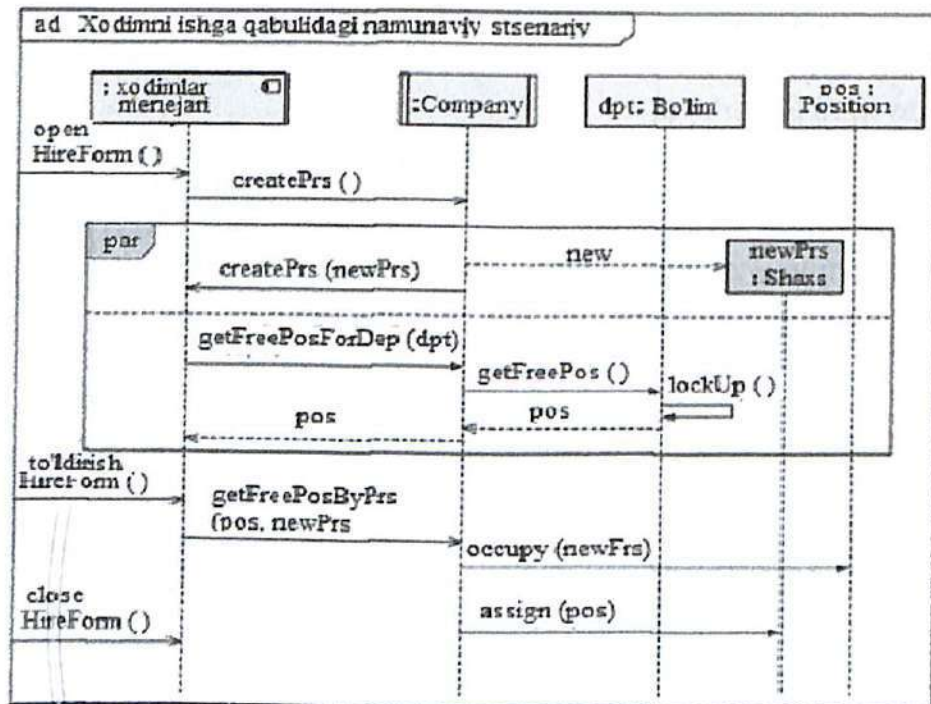
Diagrammadagi element sinflovching o'zi emas, balki sinflovchining nusxasi ekanligini ko'rsatish uchun – uning nomi tag

chiziq bilan belgilanadi. Mavhum sinflovchilar, masalan, interfeys («interface» qat'iy turi) va {abstract} turidagi cheklovchiga ega bo'lgan boshqa ixtiyoriy sinflovchilar, aniq nusxalarga ega bo'lishi mumkin emas. Agar uyushmaning nusxasining nomi - aloqa bo'lsa, u qolganlardan ajralib turadi, biz faqat ulanishning nomi yo'qligini va shuning uchun tagiga chizish uchun hech narsa yo'qligini ta'kidlaymiz. Garchi uyushma va aloqni belgilash bir xil bo'lsa-da (uzluksiz chiziq), bu aloqani – uyushmasi bilan chalkashtirib yuborish mumkin degani emas, ular turli xil foydalanish usullariga ega. "Berilganlar turi" («dataType» qat'iy turi) sinflovchisining nusxalari va uning ixtisoslashuvidagi "oddiy tur" («primitive» qat'iy turi) diagrammalarda ko'p hollarda boshqa sinflovchilar atributlarining boshlang'ich yoki doimiy qiymatlari sifatida taqdim etiladi. "Berilganlarning sanoq turi" sinflovchisining mumkin bo'lgan nusxalari uchun («enumeration» qat'iy turi) maxsus belgisi taqdim etilgan (13.2- jadvalga qarang). "Xodimlar bo'limi" axborot tizimi uchun aktyorning nusxalaridan foydalanishga misol 13.23- rasmda keltirilgan.



13.23- rasm. Haralatlanuvchi shaxs nusxalaridan foydalanishga misol

va foydalanish variantini standart ssenariysini amalga oshiramiz (13.22- rasm).



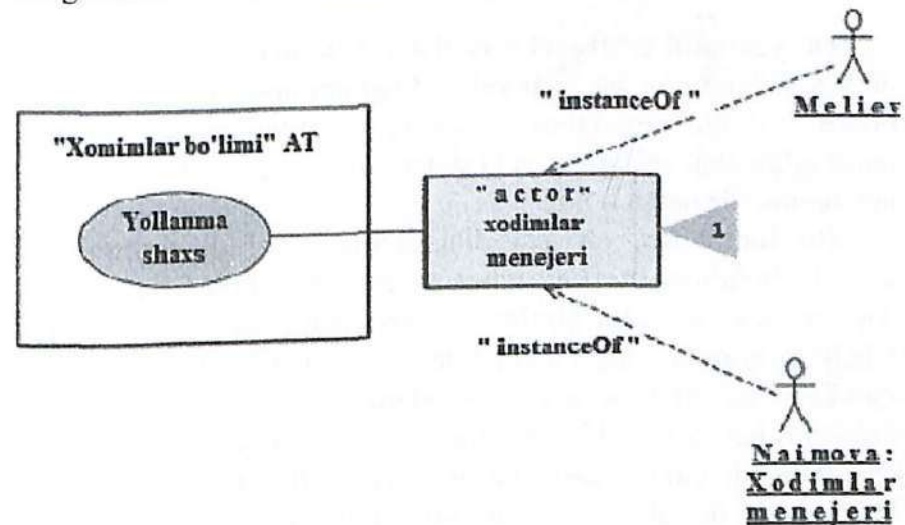
13.22- rasm. Oddiy xodimlarni yollash ssenariysi uchun ketma-ketlik diagrammasi

### 13.9. Sinflovchilarning nusxalari

Oldingi mavzularda biz juda ko'p turli sinflovchilarni ko'rib chiqdik. Biroq, shu bilan birga, biz har doim sinflovchilarning tasnifchi ekanligini ta'kidlaganmiz, ya'ni bir xil turdagi ob'ektlarning tavsifchisi, lekin hech qachon sinflovchilarning nusxalari haqida maqsadli tahlil o'tkazmadik. Endi bu bo'shliqni to'ldirish uchun biz yetarli materiallar to'pladik.

Diagrammadagi element sinflovching o'zi emas, balki sinflovching nusxasi ekanligini ko'rsatish uchun – uning nomi tag

chiziq bilan belgilanadi. Mavhum sinflovchilar, masalan, interfeys («interface» qat'iy turi) va {abstract} turidagi cheklovchiga ega bo'lgan boshqa ixtiyoriy sinflovchilar, aniq nusxalarga ega bo'lishi mumkin emas. Agar uyushmaning nusxasining nomi - aloqa bo'lsa, u qolganlardan ajralib turadi, biz faqat ulanishning nomi yo'qligini va shuning uchun tagiga chizish uchun hech narsa yo'qligini ta'kidlaymiz. Garchi uyushma va aloqni belgilash bir xil bo'lsa-da (uzluksiz chiziq), bu aloqani – uyushmasi bilan chalkashtirib yuborish mumkin degani emas, ular turli xil foydalanish usullariga ega. "Berilganlar turi" («dataType» qat'iy turi) sinflovchisining nusxalari va uning ixtisoslashuidagi "oddiy tur" («primitive» qat'iy turi) diagrammalarda ko'p hollarda boshqa sinflovchilar atributlarining boshlang'ich yoki doimiy qiymatlari sifatida taqdim etiladi. "Berilganlarning sanoq turi" sinflovchisining mumkin bo'lgan nusxalari uchun («enumeration» qat'iy turi) maxsus belgisi taqdim etilgan (13.2- jadvalga qarang). "Xodimlar bo'limi" axborot tizimi uchun aktyorning nusxalaridan foydalanishga misol 13.23- rasmda keltirilgan.



13.23- rasm. Haratlanuvchi shaxs nusxalaridan foydalanishga misol

Fursatdan foydalanib, biz shu misoldan foydalanib, «actor» (1) qat'iy turi bilan sinfdagi harakatlanuvchi shaxsni ifodalash uchun muqobil yozuvni ko'rsatamiz.

Tizimni modellashtirishda tugunlarning nusxalari («*device*», «*executionEnvironment*» qat'iy turlari) va ularga joylashtirilgan artefaktlarning nusxalari («*artifact*» qat'iy turi va 6.2- jadvalda ko'rsatilgan boshqa qat'iy turlar) katta yordam beradi, bular joylashtirish diagrammasida ko'rsatilgan. «Kadrlar bo'limi» axborot tizimiga kelsak, masalan, mavhum hisoblash tugunlarini ko'rib chiqish o'rniga, tashkilotda mavjud bo'lgan aniq kompyuterlarni ko'rib chiqish va dasturning aniq nusxalarini ushbu kompyuterlarga joylashtirish mumkin.

Ushbu band bo'yicha ko'rib chiqiladigan oxirgi sinflovchi – bu *tarkibdir*. Tarkib, ta'rifga ko'ra, nusxalarga ega bo'lishi mumkin emas, chunki u faqat tizimni modellashtirish bosqichida mavjud bo'lib, unda esa nusxalar bo'lishi mumkin emas.

### 13.10. Ob'yektlar va ob'yektlar diagrammasi

Ob'yektlarni UMLda ko'rsatish uchun maxsus diagramma — ob'yekt diagrammasi (ob'yekt diagrammasi) mavjud. Ushbu bobda sinf diagrammalari va uning tarkibiy qismlarini batafsil tahlil qilgandan so'ng, ob'yekt diagrammalariga nisbatan qo'shish uchun juda oz narsa qoldi.

Bir tomondan, ob'yekt diagrammasi sinf diagrammasining xususiy holatidan boshqa narsa emas. Ob'yektlar diagrammasidagi ob'yektlar – bu sinflarning nusxalari, ya'ni ob'yektlardir. Ularning nomlari, tag chiziq bilan ifodalanadi. Ob'yektlar diagrammasidagi munosabatlar aloqalardan, ya'ni uyushmalarning nusxalaridan iborat. Uyushmalar uchun taqdim etilgan qo'shimchalarning hammasi ham ma'noga ega emas va aloqalar uchun ko'rsatilishi mumkin. Xususan, bog'lanish qutblari uchun karralik ma'noga ega emas. Odatda, aloqa hech qanday qo'shimchalarsiz ob'yektlarni bog'laydigan chiziq sifatida ko'rsatiladi.

Boshqa tomondan, ob'yektlar diagrammasini tizimning ma'lum bir bajarilish jarayonidagi xotira ko'rinishi sifatida ko'rib chiqish mumkin (ya'ni, qaysi o'ziga xos ob'yektlar ma'lum bir vaqtda birga bajarilishi va ular o'rasida qanday aloqalar o'rnatilganligiga misol). Ob'yektlar diagrammasi misoldan boshqa narsa emasligi sababli, uning tavsiflash quvvati ham oz. Ob'yektlar diagrammasida ko'rsatilishi mumkin bo'lgan hamma narsani o'zaro harakatlar diagrammasida ko'rsatish va ko'proq ma'lumotga ega bo'lish mumkin. Shuning uchun ob'yektlar diagrammasi nisbatan kam qo'llaniladi.

### 13.11. Xulosa

1. Uyushma munosabatlarida juda ko'p turli xil qo'shimchalar termasi mavjud, ular yordamida sinflar o'rasidagi munosabatlarning alohida xususiyatlarini ko'rsatish mumkin.
2. Sinflar to'plami – «tarkib» nomli mantiqiy tuzilmaga birlashishi mumkin.
3. Har bir tarkib talab qilinuvchi va ta'minlangan interfeyslar termasi bilan tavsiflanadi.
4. Tarkib va sinflar model elementlari sifatida jismoniy mohiyatlar – artefaktlar bilan manifestatsiyalar yordamida bog'lanadi.
5. Tarkib diagrammalari tarkiblarning (artifaktlar) tuzilishini va ular o'rasidagi o'zaro aloqalarni modellashtiradi.
6. Joylashtirish diagrammalari hisoblash zaxiralar va ularga joylashtirilgan artefaktlar tuzilmasini modellashtiradi.
7. Ichki tuzilish diagrammalari murakkab sinflovchi qismlarining o'zaro harakatidagi voqe'likni ko'rsatadi. Bunday qismlar esa, o'z navbatida, ichki tuzilishga ega bo'lishi mumkin.
8. Hamkorlik – bu bir nechta sinflovchilarning o'zaro harakatidagi voqe'likni ko'rsatishning bir usuli hisoblanadi.
9. Ob'yektlar diagrammasi – bu tizim faoliyatining alohidagi vaqtida, dasturiy ob'yektlar orasidagi o'zaro aloqalarga misoldir.

### 13.12. Nazorat savollari

1. Uyushma munosabatining roli nimadan iborat?
2. Uyushma uchun qanday qo'shimchalar aniqlangan?
3. "Qism – butun" munosabatlarini modellashtirish uyushma munosabatining qaysi holatida kuzatiladi?
4. Yig'ish va kompozitsiya munosabatlariga misollar keltiring.
5. Uyushma qutbi, deganda nimani tushunasiz?
6. Uyushma qutbinig roli nimadan iborat?
7. Uyushma qutbinig roliga misol keltiring.
8. Uyushma sinfi, deganda nimani tushunasiz?
9. Uyushma sinfiga misol keltiring.
10. Uyushma qutbining malakachisi deganda nimani tushunasiz?
11. UMLda artefakt (sun'iy dalil) deganda nimani tushunasiz?
12. Tarkib deganda nimani tushunasiz?
13. UML standartidagi tarkiblar uchun qat'iy turlarni aytib bering.
14. Artefaktlarning standart qat'iy turlarini aytib bering.
15. Manifestatsiya deganda nimani tushunasiz?
16. Ichki tuzilish diagrammalarida asosiy ob'yektlarni ayting.
17. Hamkorlikni sinflavchining boshqa sinflavcyilardan farqi nimada?
18. Sinflavchining nusxasi qanday beilgilanadi?
19. Ob'yekt diagrammasining roli nimadan iborat?

## XIV- BOB. IDEF TARKIBIY TAHLIL USLUBIYATLARI

### 14.1. IDEF nima?

IDEF ICAM oilasining uslubiyatlari (Integrated Computer Aided Manufacturing - Ishlab chiqarishdagi aniqlovchilarga integrallashgan kompyuterli yorlam) – murakkab tizimlardagi masalalarni modellashtirish uchun, kehg doiradagi murakkab tizimlarni turli kesimlardagi faoliyat modellarini akslantirish va tahlil qilishdan iborat. Shu bilan birga, tizimdagi jarayonlarni tekshirishning kengligi va chuqurligi ishlab chiquvchi tomonidan belgilanadi, bu esa yaratilgan modelni ortiqcha berilganlar bilan yuklamaslikka imkon beradi.

IDEF uslubiyatlari AQSh harbiy havo kuchlari tomonidan taklif etilgan sanoatni kompyuterlashtirish dasturi – ICAM doirasida yaratilgan bo'lib, uni amalga oshirish davomida ishlab chiqarish (sanoat) tizimlarida o'zaro harakat jarayonlarini tahlil qilish usullarini ishlab chiqish zarurati aniqlangan. Ko'rib chiqilayotgan uslubiyat oilasini ishlab chiqishda asosiy talab ICAM dasturining barcha ishtirokchilari o'rasida samarali ma'lumot almashish imkoniyati mavjud bo'lgan (shuning uchun nomi: Icam DEFINITION – IDEF, boshqa variant Integrated DEFINITION).

IDEF oilasining funktsional modellashtirish uslubiyatlari standartlashtirilgan. Ularning yordami bilan biz turli xil murakkab tizimlarning faoliyat modellarini samarali namoyish va tahlil qilishimiz mumkin. IDEF uslubiyatining asosiy tamiyili har qanday o'rganilayotgan tizimni o'rganilayotgan tizimda sodir bo'layotgan jarayonlar, amallar, harakatlarni aks ettiruvchi o'zaro harakat qiluvchi va o'zaro bog'liq bloklar majmuasi sifatida aks ettirishdir.

Standart nashr etilgandan so'ng, u tijoratning turli sohalarida muvaffaqiyatli qo'llanilib, o'zining tijorat jarayonlarini tahlil qilish, loyihalash va akslantirishning samarali vositasi sifatida ko'rsatdi. IDEF uslubiyatlari bo'yicha quyidagi belgilashlari mavjud:

IDEF0 – Funksiyani modellashtirish (Function Modeling).

IDEF1 – Axborotni modellashtirish (Information Modeling).

IDEF1X (IDEF1 Kengaytirilgani) – Berilganlarni modellashtirish (Data Modeling).

IDEF2 – Loyihalashdagi taqlid modeli (Simulation Model Design).

IDEF3 – Jarayon tavsifiga ta'qib (Process Description Capture).

IDEF4 – Ob'yektga yo'naltirilgan loyihalash (Object Oriented Design).

IDEF5 – Ontologiya tavsifini egallash (Ontology Description Capture) Захват обоснования проекта

IDEF6 – Logihalash asoslarini egallash (Design Rationale Capture)

IDEF7 – Axborot tizimining auditi (Information System Auditing).

IDEF8 – Foydalanuvchi interfeysini modellashtirish (User Interface Modeling).

IDEF9 – Ssenariyga asoslangan loyihalash (Scenario Driven IS Design).

IDEF10 – Modellashtirish arxitekturasini amalga oshirish (Implementation Architecture Modeling).

IDEF11 – Axborot Artefaktini modellashtirish (Information Artifact Modeling).

IDEF12 – Tashkilotni modellashtirish (Organization Modeling).

IDEF13 – Uchta chizmani loyihalashtirishni akslantirish (Three Schema Mapping Design).

IDEF14 – Tarmoqni loyihalashtirish (Network Design).

## 14.2. IDEF0 uslubiyatining asoslari

IDEF0 uslubiyati to'rtta asosiy tushunchaga asoslangan:

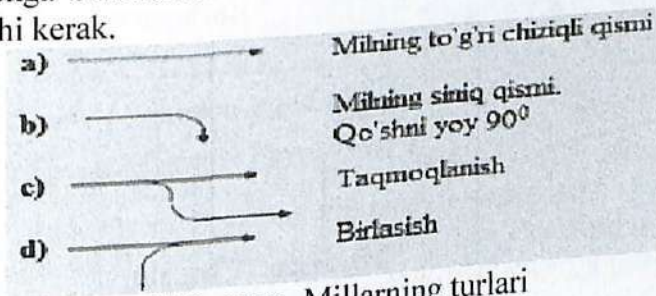
- \* funksional blok (ActivityBox);
- \* interfeys yoyi (Arrow) yoki mil;
- \* dekompozitsiya (Decomposition);
- \* izohli lug'at (Glossary).

Funksional blok misoli 14.1- rasmda keltirilgan. Har bir blok ichida uning nomi va raqami joylashtiriladi. Ism funksiyani tavsiflovchi faol fe'l yoki fe'l burmasi bo'lishi kerak. Blok raqami pastki o'ng burchakda joylashadi. Blok raqamlari ularni diagrammada va tegishli matnda aniqlash uchun ishlatiladi.



14.1- rasm. Funksional blok

Mil (yoy) bir yoki bir nechta to'g'ri chiziq segmentlaridan va bir uchida poyanak bo'ladi (14.2- rasm, a). Millarning segmentlari tekis yoki siniq tarzda bo'lishi mumkin, bunda milning gorizont va vertikal segmentlari 90° burchakka ega bo'lgan yoylar bilan bog'lanadi (14.2- rasm, b). Millarni qurish qoidalari quyidagi 14.2- rasmda keltirilgan. Millarning uchlari funksional blokning tashqi chegarasiga tegishi kerak, lekin uni kesib o'tmasligi kerak. Millar blokga burchaklarda emas, balki uning yon tomonlariga birlashtirilishi kerak.



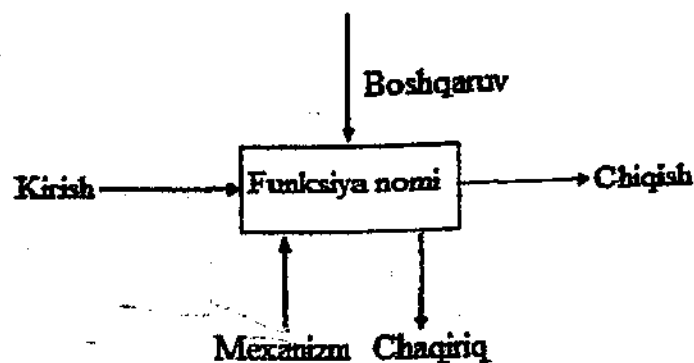
14.2- rasm. Millarning turlari

Funksional blokning har bir tomoni standart qiymatga ega va birlashtirilgan milning rolini aniq belgilaydi. Blokning chap tomoniga kiradigan millar kirishlarni bildiradi. Kirishlar funksiya tomonidan, uning chiqishida paydo bo'luvchini yaratish uchun o'zgartiriladi yoki ishlatiladi. Yuqoridan blokga kiruvchi millar boshqaruvga mos keladi. Blokning o'ng tomonidan chiquvchi millar chiqishni, ya'ni funksiya tomonidan ishlab chiqarilgan berilganlar yoki moddiy ob'yektlarni bildiradi. Blokning pastki qismiga ulangan millar esa mexanizmlarni ifodalaydi.

Yuqoriga yo'naltirilgan millar funksiyaning bajarilishini qo'llab-quvvatlovchi vositalarni aniqlaydi. Boshqa vositalar o'zaro blokidan meros bo'lishi mumkin.

Mexanizmning pastga yo'naltirilgan millari chaqiriq millari hisoblanadi. Chaqiriq millari shu modeldan yoki modelning ma'lum bir qismidan boshqa modelga yoki modelning boshqa qismiga kiritilgan blokga ularning aloqasini ta'minlovchi murojaatni bildiradi, ya'ni turli xil modellar yoki bir xil modelning turli qismlari bir xil element (blok)dan foydalanishlari mumkin. Millarning standart joylashuvi 14.3- rasmda keltirilgan.

Funksiya nomlari (funksional bloklar)ga misollar – qismlarni ishlab chiqarish; zaxiralarni rejalashtirish; bajarilishini nazorat qilish; tizimni loyihalash; tizimdan foydalanish; chizmalarni yaratish; qismni ishlab chiqarish; qismni tekshirish.

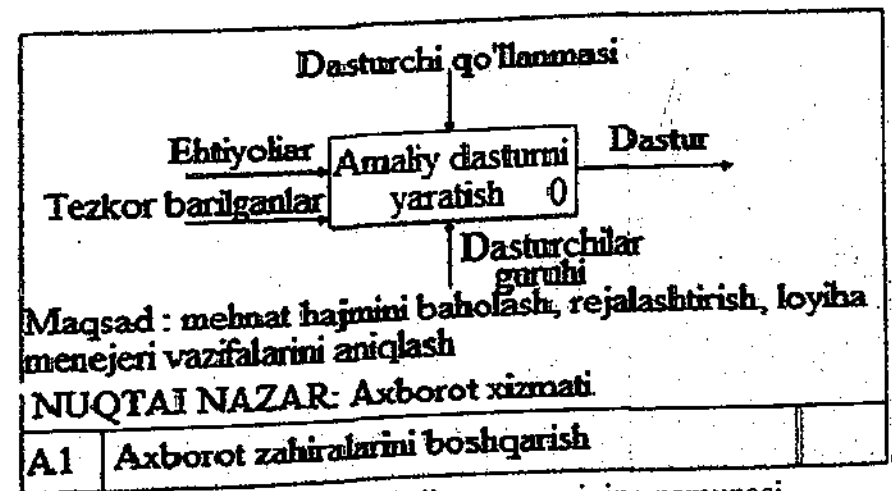


14.3- rasm. Standart funksional blok

Millar funksiyaning bajarish uchun zarur bo'lgan yoki u tomonidan ishlab chiqarilgan berilganlar yoki moddiy ob'yektlarni aniqlaydi. Har bir mil ot yoki ot aylanmasi bilan belgilanishi kerak, masalan: tasniflar; sinov hisoboti; byudjet; loyiha talablari; qism dizayni; ko'rsatma; talablar.

Mil belgilarida funksiya, kirish, boshqaruv, chiqish, mexanizm, chaqiriq kabi umumiy atamalar ishlatilmasligi kerak.

Har bir model yuqori darajadagi kontekstli diagrammaga ega bo'lishi kerak, unda modellashtirish ob'yekti chegaralar millari yagona bo'lgan blok bilan ifodalanadi. Ushbu *diagramma A-0*, deb nomlanadi. Ushbu diagrammadagi millar modellashtirish ob'jektining atrof-muhit bilan aloqalarini aks ettiradi. Yagona blok butun ob'jektning ifodalanganligi sababli, uning nomi butun loyiha uchun umumiydir. Xuddi shu narsa barcha diagramma millari uchun ham amal qiladi, chunki ular ob'jektning tashqi interfeyslarining to'liq majmuasini ifodalaydi. A-0 diagrammasi modellashtirish maydonini va uning chegarasini belgilaydi. A-0 diagrammasining namunasi 14.4- rasmda keltirilgan.



14.4- rasm. A-0 diagrammasining namunasi

A-0 kontekstli diagramma, shuningdek, model yaratilayotgan mansabdor shaxs yoki bo'linmaning nuqtayi nazarini va unga eri-

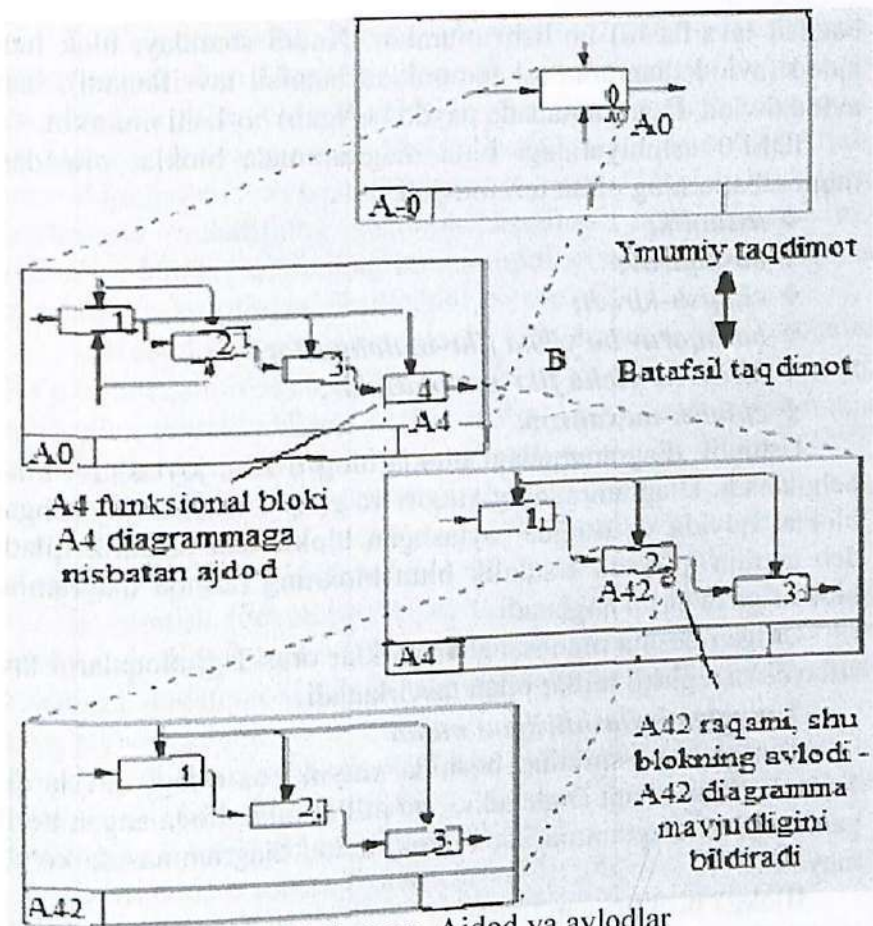
shish uchun mo'ljallangan maqsadni belgilaydigan qisqa bayonetlarni o'z ichiga olishi kerak. Nuqtayi nazarning o'zgarishi ob'jektning boshqa jihatlarini ko'rib chiqishga olib keladi. Bir nuqtayi nazardan muhim bo'lgan jihatlar bir xil ob'jektda boshqa nuqtayi nazardan ishlab chiqilgan modelda paydo bo'lmasligi mumkin. Maqsadni shakllantirish modelni yaratish sababini ifodalaydi, ya'ni model javob berishi kerak bo'lgan savollar ro'yxatini o'z ichiga oladi, bu uning tuzilishini sezilarli darajada belgilaydi.

Ob'jektning eng muhim xususiyatlari odatda shajaraning yuqori darajalarida aniqlanadi. Yuqori darajadagi funktsiya (funktsional blok) bo'linishi va qism-funksiyalarga ajralishi bilan bu xususiyatlar aniqlanadi. Har bir qism-funksiyani keyingi darajadagi elementlarga ajralishini ifodalash mumkin va bu modelashtirish maqsadida tuzilgan savollarga javob berishga imkon beradigan tegishli tuzilma olinmaguncha sodir bo'ladi. Har bir qism-funksiya alohida blok tomonidan modellashtiriladi. Har bir ajdod bloki quyi darajadagi avlod diagrammasi bilan batafsil tavsiflanadi. Barcha avlodlar jadvallari yuqori darajadagi kontekst diagrammasi doirasidan tashqariga chiqmasligi kerak.

### 14.3. Avlodlar diagrammasi

Yuqori darajadagi kontekstli diagrammada ko'rsatilgan funktsiyani avlod diagrammasini yaratish orqali asosiy qism-funksiyalarga ajratish mumkin. O'z navbatida, ushbu qism-funksiyalarning har biri keyingi, quyi darajadagi avlodlar diagrammasini yaratish orqali tarkibiy qismlarga bo'linishi mumkin.

Har bir avlod diagrammasi ajdod blokining qo'shimcha tafsilotlarini ta'minlaydigan avlod bloklari va millarini o'z ichiga oladi. Ajratish paytida yaratilgan avlod diagrammasi ajdod bloki bilan bir xil maydonni qamrab oladi, ammo uni batafsilroq tavsiflaydi. Shuning uchun avlodlar jadvalini ajdod blokiga joylashtirilgan, deb hisoblash mumkin. Ushbu tuzilma 14.5- rasmda keltirilgan.



14.5- rasm. Ajdod va avlodlar

### 14.4. Ajdodlar diagrammasi

**Ajdod diagramma** – bu bir yoki bir nechta ajdod bloklarini o'z ichiga olgan diagrammadir. Har bir kontekstsiz diagramma ham avlodlar diagrammasidir, chunki ta'rifga ko'ra, u ba'zi ajdod bloklarini batafsil bayon qiladi.

Shunday qilib, har qanday diagramma ham ajdod diagramma (ajdod bloklarini o'z ichiga oladi), ham avlod (o'z ajdod bloklarini

batafsil tavsiflashi) bo'lishi mumkin. Xuddi shunday, blok ham ajdod (avlod diagrammasi tomonidan batafsil tavsiflanadi), ham avlod (avlod diagrammadada paydo bo'lgan) bo'lishi mumkin.

IDEF0 uslubiyatidagi bitta diagrammada bloklar orasidagi munosabatlarning oltita turi mavjud:

- ❖ *ustunlik;*
- ❖ *boshqaruv;*
- ❖ *chiqish-kirish;*
- ❖ *boshqaruv bo'yicha fikr-mulohazalar;*
- ❖ *kirish bo'yicha fikr-mulohazalar;*
- ❖ *chiqish-mexanizm.*

Ustunlik diagrammadagi bloklarning o'zaro joylashuvi bilan belgilanadi. Diagrammaning yuqori va chap tomonida joylashgan bloklar quyida va o'ngda joylashgan bloklardan ustunlik qiladi, deb taxmin qilinadi. Ustunlik bitta blokning boshqa diagramma bloklariga ta'sirini anglatadi.

Qolgan beshta munosabatlar bloklar orasidagi aloqalarni tavsiflaydi va tegishli millar bilan tasvirlanadi.

#### ***Tunnelga joylashtirilgan millar***

Tunnel – bu milning boshida va/yoki oxiridagi qavslardir. Tunnel millari shuni anglatadiki, bu millar bilan ifodalangan berilganlar ajdod diagrammasida va/yoki avlod diagrammasida ko'rilmaydi.

Blokga biriktirilgan tunnelga joylashtirilgan mil, bu mil bilan ifodalangan berilganlar keyingi parchalanish darajasida majburiy emasligini anglatadi.

Tunneldagi erkin uchiga joylashtirilgan mil, u bildirgan berilganlar ota-ona diagrammasida yo'qligini anglatadi. Tunnel, diagrammani to'ldirib yubormaslik uchun ishlatiladi.

### **14.5. Diagrammalarni qurish qoidalari**

1. Model faqat bitta blokni o'z ichiga olgan A-0 kontekst diagrammasidan iborat bo'lishi kerak.

2. Diagrammadagi bloklar diagonal ravishda – diagrammaning yuqori chap burchagidan pastki o'ng tomonga tayinlangan raqamlar tartibida joylashtirilishi kerak. Yuqoridagi chap tomonda joylashgan diagrammadagi bloklar pastki o'ng tomonda joylashgan bloklardan ustun. Diagramma varag'idagi bloklarning joylashuvi muallifning ustunlik haqidagi tushunchasini aks ettiradi. Shunday qilib, diagramma topologiyasi qaysi funksiyalar qolganlarga ko'proq ta'sir qilishini ko'rsatadi.

3. Kontekstsiz diagrammalarda kamida uchta yoki oltitadan ko'p bo'lmagan bloklar bo'lishi kerak. Ushbu cheklovlar diagrammalarning murakkabligini o'qish, tushunish va foydalanish mumkin bo'lgan darajada qo'llab-quvvatlaydi.

4. Kontekstsiz diagrammaning har bir bloki pastki o'ng burchakda joylashgan raqamni oladi; raqamlash tartibi yuqori chapdan pastki o'ng blokga (1 dan 6 gacha raqamlar).

5. Ajratish (dekompozitsiya) mo'ljallangan har bir blokning avlod jadvaliga havolasi bo'lishi kerak; havola (masalan, tugun raqami, C-raqam yoki sahifa raqami) blokning pastki o'ng burchagiga joylashtiriladi.

6. Blok nomlari (bajariladigan funksiyalar) va mil belgilari noyob bo'lishi kerak. Agar mil nishohlari bir xil bo'lsa, demak, millar bir xil berilganlarni aks ettiradi.

7. Murakkab topologiyaga ega millar mavjud bo'lganda, uni aniqlash qulayligi uchun nishonni takrorlash tavsiya etiladi.

8. Diagrammani o'qishni osonlashtirish uchun millar bloklari va burilishlari, shuningdek millar bloklari va kesishmalari orasidagi maksimal masofani ta'minlash kerak. Shu bilan birga, ikki xil milni aralashtirish ehtimoli kamayadi.

9. Bloklar har doim kamida bitta boshqaruv va bitta chiqish miliga ega bo'lishi kerak, lekin kirish millari bo'lmashligi mumkin.

10. Agar bir xil berilganlar, boshqarish va kirish uchun xizmat qilsa, faqat boshqarish mili chiziladi. Bu berilganlarning boshqaruv xususiyatini ta'kidlaydi va diagrammaning murakkabligini kamaytiradi.

11. Parallel millar orasidagi maksimal masofa nishonlarni joylashtirishni, ularni o'qishni osonlashtiradi va millarning yo'llarini kuzatishga imkon beradi.

12. Millar o'xshash berilganlarni ifodalasa va ularning manbasi diagrammada ko'rsatilmagan bo'lsa, ular birlashtiriladi.

13. Boshqaruv mulohazalari blokning yuqori va yuqori qismida joylashgan millar bilan ko'rsatilishi kerak. Teskari aloqalar blokning pastki va pastki qismida joylashgan millar bilan ko'rsatilishi kerak. Xuddi shunday teskari aloqa, mexanizm orqali ham ko'rsatiladi. Shunday qilib, chiziqlar va kesishmalarning minimal soni bilan teskari aloqa ko'rsatiladi.

#### 14.6. IDEF0 uslubiyati va tilining asosiy tushunchalari

IDEF0 uslubiyatida, tizim va uning elementlarida sodir bo'ladigan hamma narsa *funksiyalar* deb ataladi. har bir Blok funksiyaga mos keladi. IDEF0 diagrammasida blok to'rtburchakdan iborat. Blok boshqa bloklar bilan yoki modellashtirilayotgan tizimga tashqi muhit bilan o'zaro ta'sir qiladigan interfeyslar blokga kiradigan yoki chiqadigan millar bilan ifodalanadi. Kiruvchi millar, blok bilan tasvirlangan funksiyani bajarish uchun, bir vaqtning o'zida qanday shartlarni bajarish zarurligini ko'rsatadi.

IDEF0 diagrammalari oddiy blok va mil grafikalariga asoslangani bo'lib, bloklar va millarni tavsiflash uchun nishonlar tabiiy tilda bajariladi. Diagrammalarning ketma-ket parchalanishi (ajratilishi) shajaraviy tamoyil asosida qurilgan bo'lib, bunda asosiy funksiyalar yuqori darajada ko'rsatiladi, so'ngra ular maydalashtiriladi va aniqlashtiriladi.

IDEF0da modelni ishlab chiqish bosqichma-bosqich, qadam-qadam protseduradir. Qadamning har bir bosqichida ishlab chiquvchi muhokama qilinadigan, ko'rib chiqiladigan, taqrizlanadigan va keyingi tahrir qilinadigan model variantini taklif qiladi, shundan so'ng sikl takrorlanadi.

IDEF0 uslubiyati va tilining asosiy tushunchalari quyidagilardan iborat:

1. **Blok:** ism va raqamni o'z ichiga olgan va funksiyani tavsiflash uchun ishlatiladigan to'rtburchak.

2. **Tarmoqlanish:** milni ikki yoki undan ortiq segmentlarga bo'lish.

3. **Ichki mil:** kirish, boshqarish yoki chiqish mili, uning uchlari manba va iste'molchini bitta diagrammaning bloklari bilan bog'laydi. Chegara milidan farq qiladi.

4. **Kirish mili:** IDEF0 blokiga kirishini ko'rsatadigan millar sinfi, ya'ni funksiya tomonidan chiqishga aylantiriladigan berilganlar yoki moddiy ob'yektlar. Kirish millari IDEF0 blokining chap tomoniga ulanadi.

5. **Chiqish mili:** IDEF0 blokidan chiqishini ko'rsatadigan millar sinfi, ya'ni funksiya tomonidan ishlab chiqarilgan berilganlar yoki moddiy ob'yektlar. Chiqish millari IDEF0 blokining o'ng tomoniga ulanadi.

6. **Izohli lug'at (glossariy):** tugunlar, bloklar, millar yoki umuman IDEF0 modeli bilan bog'liq kalit so'zlar, iboralar va qisqartmalar uchun ta'riflar ro'yxati.

7. **Chegara mili:** bir uchi manba yoki iste'molchi bilan bog'langan, ikkinchisi diagrammadagi hech qanday blokga ulanmagan mil. Diagrammani, tizimning boshqa bloklari bilan aloqasini aks ettiradi, va ichki mildan farq qiladi.

8. **Parchalanish:** modellashtiriladigan funksiyani tarkiblarga ajratish.

9. **Tugunlar daraxti:** IDEF0 modelidagi ajdod va avlod tugunlari o'rasidagi munosabatlarni daraxt grafi shaklida ifodalash.

10. **A0 diagrammasi:** IDEF0 diagrammasining yuqori darajadagi funksiyani, uning kirishlari, chiqishlari, boshqarilishi va mexanizmlarini modelning maqsadi va model qurilayotgan nuqtayi nazarni shakllantirish bilan birga tavsiflovchi bitta blokdan iborat maxsus (kontekstli) tasvir.

11. **Diagramma:** blokning parchalanishini tavsiflovchi modelning bir qismi.

12. **Namoyish diagrammasi FEO (for exposure only – faqat ekspozitsiya uchun):** IDEF0 diagrammasi haqida aniq faktlarni etkazish uchun ishlatiladigan grafik tavsif. FEO diagrammalarini tuzishda IDEF0 qoidasiga rioya qilmaslik mumkin.

13. **Avlod blok:** avlod (yaratilgan) diagrammasidagi blok.

14. **Avlod diagramma:** ajdod (yaratuvchi) blokini batafsil ko'rsatadigan diagramma.

15. **Blok nomi:** blok ichiga joylashtirilgan va modellashtiriladigan funksiyani tavsiflovchi fe'l yoki fe'l aylanmasi.

16. **Interfeys:** berilganlar yoki moddiy ob'yektlar o'tadigan ajratuvchi chegara; berilganlar yoki moddiy ob'yektlarni bir tarkibdandan ikkinchisiga uzatuvchi ikki yoki undan ortiq model tarkiblari orasidagi aloqa.

17. **ICOM kodi:** qisqartma (Input - kirish, Control – boshqarish, Output - chiqish, Mexanizm - mexanizm), avlod diagrammasining chegara millarining ajdod blokining millari bilan mos kelishini ta'minlaydigan kod; havolalar uchun ishlatiladi.

18. **Kontekst:** funksiya ishlaydigan muhit (yoki diagrammadagi funksiyalar majmuasi).

19. **Kontekst diagramma:** model kontekstini ifodalovchi An ( $n \geq 0$ ) tugun raqamiga ega bo'lgan diagramma, bitta blokdan iborat A0 diagrammasi zarur (majburiy) kontekst diagramma; A1, A2, ... tugun raqamlari bo'lgan diagrammalar qo'shimcha kontekst diagrammalar.

20. **Mil nishoni:** mil yoki mil segmentiga bog'langan va ularning ma'nosini belgilaydigan ot yoki ot aylanmasi.

21. **IDEF0 modeli:** ma'lum bir maqsad va tanlangan nuqtayi nazardan ishlab chiqilgan tizimning grafik tavsifi. Grafik, matn va izohli lug'at yordamida tizim funksiyalarini tasvirlaydigan bir yoki bir nechta IDEF0 diagrammalar to'plami.

22. **Blok raqami:** blokning pastki o'ng burchagiga joylashtirilgan va diagrammadagi blokni noyob tarzda aniqlaydigan raqam (0 – 6).

23. **Tugunlar ro'yxati:** IDEF0 model tugunlarini tartibli ravishda ko'rsatadigan ro'yxat, ko'pincha zina shaklida. Tugunlar daraxti bilan bir xil ma'no va tarkibga ega (9- band).

24. **Modelga eslatma:** IDEF0 diagrammasining bir qismi bo'lgan va o'z grafik tasvirni topmagan faktni yozish uchun ishlatiladigan matnli sharh.

25. **Ajdod diagramma:** avlod blokini o'z ichiga olgan diagramma.

26. **Avlod bloki:** avlod diagrammasi bilan batafsil tavsiflangan blok.

27. **Bog'lanish/ajratish:** millarning qiymatlarini qo'shma qiymatga birlashtirish ("to'plam"ga bog'lash) yoki millarning birlashishi yoki ajralishini sintaksisi bilan ifodalangan millarning qiymatlarini ajratish ("to'plam"ni echish).

28. **Mil segmenti:** blokning yon tomonida, filial yoki qo'shilish nuqtasida yoki chegarada (milning bog'lanmagan uchi) boshlanadigan yoki tugaydigan chiziq segmenti.

29. **Semantika (ma'no):** tilning sintaktik tarkibiy qismlarining ma'nosi.

30. **Sintaksis:** tilning tarkibiy qismlari yoki xususiyatlari va ular o'tasidagi munosabatlarni belgilaydigan qoidalar.

31. **Birlashish:** ikki yoki undan ortiq mil segmentlarini bitta segmentga birlashtirish.

32. **C-nomer:** xronologik tartibda yaratilgan va diagrammani aniqlash va uning tarixini kuzatish uchun ishlatiladigan raqam; Diagrammaning ma'lum bir naqlini aniqlashda mos yozuvlar ifodasi sifatida ishlatilishi mumkin.

33. **Mil:** ochiq kanalni yoki berilganlar yoki moddiy ob'yektlarni manbadan (milning boshlang'ich nuqtasi) iste'molchiga ("uchi" bo'lgan so'nggi nuqta) uzatuvchi kanalni modellashtiradigan bir yoki bir nechta segmentlardan tashkil

topgan yo'naltirilgan chiziq. Millarning 4 sinfi mavjud: kirish mili, chiqish mili, boshqaruv mili, mexanizm mili (chaqiriq milini o'z ichiga oladi).

34. **Chaqiriq mili:** ma'lum bir model blokidan (yoki modelning bir qismidan) boshqa model blokiga (yoki bir xil modelning boshqa qismiga) murojaat qilishni bildiruvchi va modellar o'rasida yoki bitta modelning turli qismlari o'rtasida aloqani ta'minlaydigan mexanizm milining turi.

35. **Mexanizm mili:** IDEF0 mexanizmlarini aks ettiruvchi millar sinfi, ya'ni funksiyani bajarish uchun ishlatiladigan vositalar; chaqiriq milining maxsus holatini o'z ichiga oladi. Mexanizm millari IDEF0 blokining pastki qismiga ulanadi.

36. **Tunnelga joylashtirilgan mil (tunnel mili):** avlod diagrammasidagi har bir mil ajdod diagrammasidagi millarga mos kelishi kerakligi haqidagi odatiy talabni qondirmaydigan mil (maxsus yozuvli).

37. **Matn:** IDEF0 grafik diagrammasiga har qanday matnli (grafik bo'lmagan) sharh.

38. **Tilda:** nishonni milning ma'lum bir segmentiga ulash uchun ishlatiladigan kichik singan (to'lqinli) chiziq yoki diagramma tarkibi bilan model eslatmasi.

39. **Nuqtayi nazar:** model ishlab chiqilayotgan tashkilotning mansabdor shaxsi yoki bo'linmasining havolasi.

40. **Tugun:** avlodlar bloklarini yaratadigan – ajdod blok.

41. **Tugun havolasi:** diagrammani aniqlash va model shajarasidagi o'rnini aniqlash uchun berilgan kod; qisqartirilgan model nomi va qo'shimcha kengaytmali diagrammaning tugun raqamidan hosil bo'ladi.

42. **Diagrammaning tugun raqami:** diagrammaning tugun havolasining ajdod blok raqamiga mos keladigan qismi.

43. **Tugun raqami:** blokga berilgan va uning model shajarasidagi o'rnini belgilaydigan kod; batafsil mos yozuvlar ifodasi sifatida ishlatilish mumkin.

44. **Boshqarish mili:** IDEF0-da boshqaruvni ko'rsatadigan millar sinfi, ya'ni blokning chiqishi to'g'ri bo'lgan shartlar. Boshqaruv sifatida modellashtirilgan berilganlar yoki ob'yektlar mos keladigan chiqishni yaratadigan funksiya tomonidan o'zgartirilishi mumkin. Boshqarish millari IDEF0 blokining yuqori qismiga ulanadi.

45. **Funksiya:** nima qilish kerakligini tavsiflovchi fe'l yoki fe'l shakli bilan aniqlanadigan faoliyat, IDEF0 bloki tomonidan modellashtirilgan, jarayon yoki o'zgartirish.

46. **Maqsad:** modelni yaratish sababini qisqacha shakllantirish.

#### 14.7. IDEF1X uslubiyati

Axborot modelini loyihalashdagi eng muhim maqsad – bu mavjud bo'lgan o'rganilayotgan predmet sohaning bir-biriga qarshi bo'lmagan tuzilishlarini va bu tuzilishdagi tarkiblar orasidagi o'zaro harakatlarni yaratishdan iborat. Berilganlarning konseptual modeli tushunchasi IDEF1X vositasida berilganlarni semantik modellashtirish uslubiyati, ya'ni berilganlarni boshqa berilganlar bilan o'zaro aloqasining tasavvur qilishdan iborat.

IDEF1X standartining asosiy maqsadlari quyidagilardan iborat:

- tashkilotlarning axborot zaxiralarini to'liq tushunish va tahlil qilish vositasi;
- berilganlarning murakkabligini taqdim etish va uzatishning umumiy vositasi;
- korxonaning ishlashi uchun zarur bo'lgan berilganlarni umumiy taqdim etish usullari;
- foydalanuvchilar tomonidan tekshirilishi va jismoniy berilganlar bazasi loyihasiga aylantirilishi mumkin bo'lgan ilovadan mustaqil berilganlarni taqdim etishni aniqlash vositalari;

- mavjud berilganlar manbalaridan berilganlarning moslashgan ta'rifini olish usullari.

IDEFIX-ning asosiy maqsadi moslashishni qo'llab-quvvatlashdir. Moslashish yondashuvi "konseptual chizma" deb nomlangan berilganlar manbasining yagona semantik ta'rifini olish, boshqarish va undan foydalanishga qaratilgan. "Konseptual chizma" korxonada ichidagi berilganlarning yagona moslashish ta'rifini taqdim etadi, bu berilganlarning biron bir qo'llanilishiga nisbatan noxolis emas va bu berilganlarning jismoniy saqlanishi yoki mavjudligiga bog'liq emas. Konseptual chizmaning asosiy maqsadi berilganlar yaxlitligini birlashtirish, almashish va boshqarish uchun ishlatilishi mumkin bo'lgan berilganlar o'rasidagi qiymatlar va munosabatlarni izchil aniqlashni ta'minlashdir. Konseptual chizma uchta muhim xususiyatga ega bo'lishi kerak:

- tijorat infratuzilmasiga mos kelishi va barcha qo'llaniladigan sohalar uchun to'g'ri bo'lishi;
- shunday kengaytirish kerakki, bunda yangi berilganlar ilgari aniqlangan berilganlarni o'zgartirmasdan aniqlanishi lozim;
- kerakli foydalanuvchi ko'rinishlariga va turli xil berilganlarni saqlash va kirish tuzilmalariga o'tish qobiliyatining mavjudligi.

P. Chen yondashuviga asoslangan va T. Ramey tomonidan yaratilgan IDEF1 usuli uchinchi me'yoriy shakldagi berilganlarning relyatsion modeli talablariga mos javob beruvchi ekvivalent model qurish imkoniyatini yaratadi. Hozirgi paytda IDEF1 uslubiyatining takomillashgan naqli IDEFIX ishlatilmoqda. IDEFIX uslubiyati – «mohiyat-aloqa» konsepsiyasiga asoslangan berilganlarni semantik modellashdagi yondashuvlardan biri. Turli tabiatdagi tizimlarni axborot tuzilishini tahlil qilish vositasi. IDEFIX uslubiyati yordamida qurilgan axborot modeli, tizimdagi ob'yektlar xususidagi

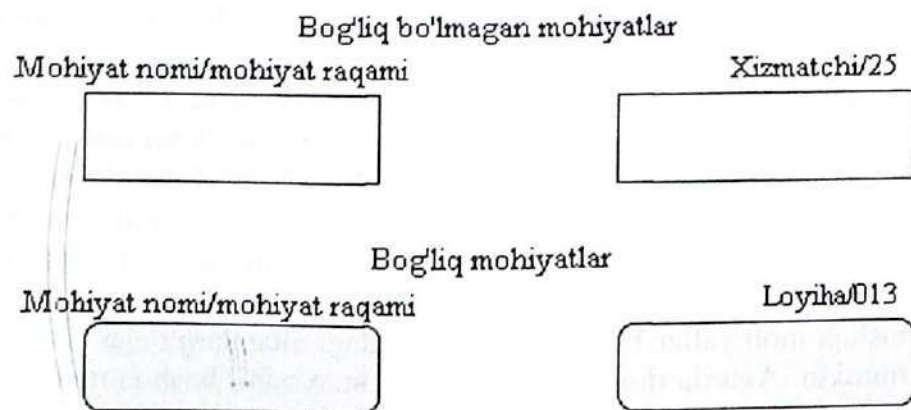
axborotlarning mantiqiy tuzilishini akslantiradi. IDEFIX uslubiyatida o'rganishni soddalashtirish va avtomatizatsiyalashni amalga oshirish imkoniyatlari hisobga olingan. IDEFIX diagrammalari bir qancha keng tarqalgan CASE vositalar tarkibida ishlatilmoqda xususan, ERwin, ERStudio, Design/IDEF tizimlarida. Bu uslubiyat AQSH armiyasi uchun yaratilgan bo'lib, davlat muassasalari, moliya va ishlab chiqarish korporatsiyalarida keng ishlatiladi. Shunday qilib, IDEFIX standartiga mos ravishda taqdim etilgan konseptual model, loyihalashtirilayotgan tizimdagi berilganlar bazasining mantiqiy chizmasidan iborat. IDEFIX konsepsiyasida *mohiyat* – bu modellashtirilayotgan tizimdagi alohida ob'jekt yoki voqea bo'lib, xususiyatlarga ega bo'lgan to'plam - atributlardan iborat. Ushbu to'planning alohidagi elementi «*mohiyatning nusxasi*» deyiladi.

IDEFIX standarti yordamida mohiyatlarning ikki turi – *bog'liq va bog'liqmas* hamda aloqalarning - *aniqlovchi va aniqlovchi bo'lmaganlarini* tasvirlash mumkin. IDEFIX uslubiyatida, mohiyat aniqlovchilarga bog'liq bo'lmaydi yoki mohiyatning har bir nusxasini uning boshqa mohiyat bilan munosabatlari aniqlanmagan holatda bir qiymatli aniqlash mumkin. Har bir mohiyat boshqa mohiyatlar bilan ixtiyoriy sondagi aloqalarga ega bo'lishi mumkin. Agarda mohiyatning har bir nusxasini, boshqa mohiyatlar bilan aloqasini aniqlamagan holda, bir qiymatli aniqlash mumkin bo'lsa, u *erkin (mustaqil, bog'liqsiz) mohiyat* deyiladi.

Agarda mohiyatning nusxasini bir qiymatli aniqlash uning boshqa mohiyatlar bilan aloqasiga bog'liq bo'lsa, u *tobe (avlod) mohiyat* bo'ladi. Mohiyat uni yaratuvchi (ajdod) mohiyatdagi aloqa orqali voris sifatida o'tgan atributlarga ega bo'lishi mumkin. Bu atributlar odatda *tashqi kalit* bo'ladi va mohiyatlar orasidagi aloqalarni tashkillashtirishda ishlatiladi. Agarda mohiyatning tashqi kaliti uning birlamchi kaliti (RK) yoki tarkibli birlamchi kalitning qismi sifatida ishlatilsa, u holda mohiyat *ajdod* mohiyatga *tobe (avlod)* bo'ladi. Agarda tashqi kalit birlamchi kalit

bo'lmasa va u birlamchi kalitning tarkibiy qismi bo'lmasa, u holda mohiyat ajdod mohiyatga nisbatan *erkin* bo'ladi.

Agarda mohiyat aniqlovchilarga bog'liq bo'lsa yoki mohiyat nusxalarini bir qiymatli aniqlash boshqa mohiyat bilan o'zaro munosabatiga bog'liq bo'lsa, *aniqlovchilarga bog'liq mohiyat* yoki *bog'liq mohiyat* deb ataladi. *O'zaro bog'langan (uyushma-viy) mohiyat* deganda esa, ikkita yoki undan ortiq mohiyat bilan bog'langan mohiyat tushuniladi. Har bir mohiyatga noyob nom va yassi chiziq («/») bilan ajratilgan mohiyat raqami yoziladi hamda blok ustida joylashtiriladi. Bu mohiyatlarni belgilashda IDEFIX uslubiyatida qabul qilingan yondashuvlar 14.6- rasmda keltirilgan.



14.6- rasm. IDEFIX uslubiyatidagi belgilashlar

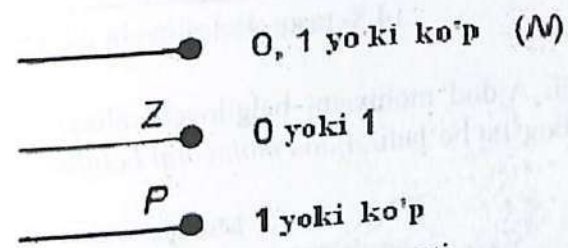
Aloqa qo'shimcha ravishda daraja yoki quvvatni (*ajdod mohiyatning har bir nusxasi uchun mavjud bo'lgan avlod mohiyat nusxalarining soni*) ko'rsatish yo'li bilan aniqlanadi. IDEFIX uslubiyatida aloqalar quvvati quyidagicha bo'lishi mumkin:

- ajdod mohiyatning har bir nusxasiga avlod mohiyatlarining 0, 1 yoki ko'p nusxalarining mosligi;
- ajdod mohiyatning har bir nusxasiga avlod mohiyatlar 0 yoki 1 nusxasining mosligi;

- ajdod mohiyatning har bir nusxasiga avlod mohiyatning 1 yoki ko'p nusxasining mosligi;
- ajdod mohiyatning har bir nusxasiga avlod mohiyat cheklangan sondagi nusxasining mosligi (masalan, 4).

Agarda avlod mohiyat nusxasi o'zining ajdod mohiyati nusxasi bilan bir qiymatli aniqlansa, *aloqa aniqlangan*, aks holda *aloqa aniqlanmagan* deb ataladi. Boshqacha aytganda, agarda mohiyat erkin bo'lsa, uning ajdod mohiyat bilan aloqasi *aniqlovchi*, aks holda *aniqlovchi bo'lmaydi*.

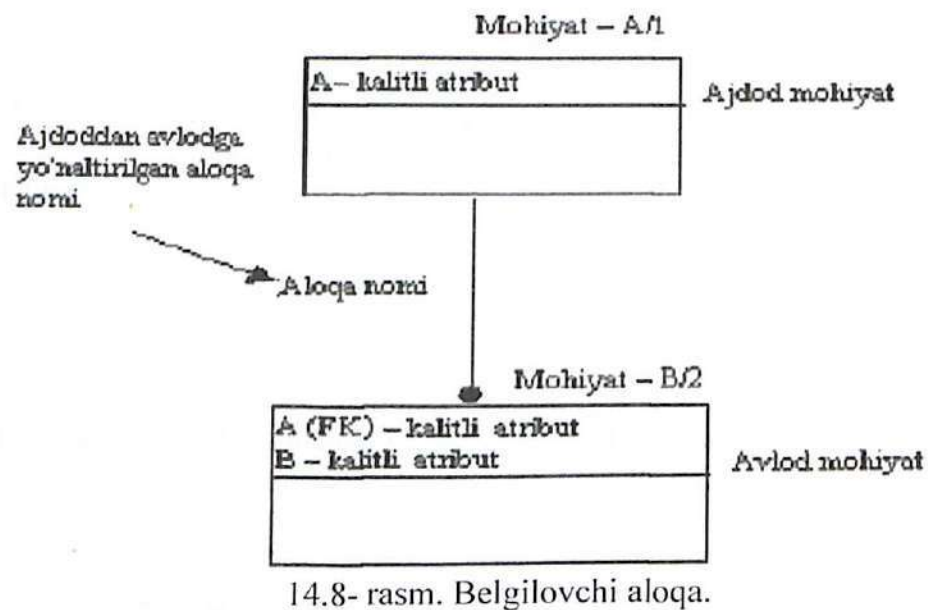
ER-diagrammada aloqa, ya'ni ajdod mohiyat va avlod mohiyat orasidagi aloqa chiziq orqali ifodalanadi. Aloqaning quvvati 14.7- rasmdagi kabi belgilanadi. *Oshkormas ravishdagi* aloqa quvvati Nga teng, deb qabul qilinadi.



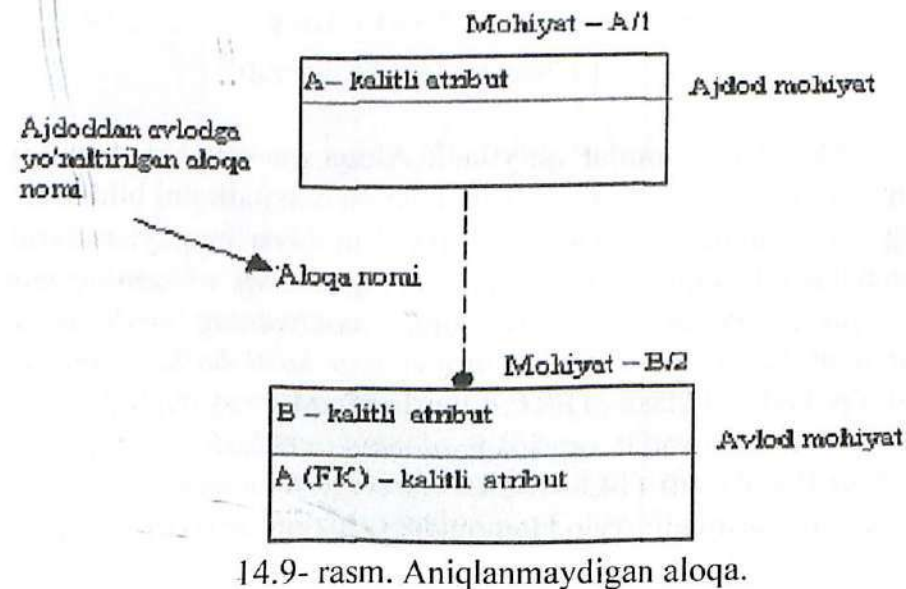
14.7- rasm. Aloqa quvvati.

Aloqalarga nomlar qo'yiladi. Aloqa quvvati ajdod mohiyat uchun avlod mohiyatning nechta nusxasi mavjudligini bildiradi.

Aloqaning nomi har doim ajdod mohiyat nuqtayi nazaridan shakllanadi. *Agarda ajdod mohiyatning nomini, aloqaning nomini, quvvatning ifodasini va yaratilgan mohiyatning nomini ma'no-siga nisbatan birlashtirsak, to'g'ri gap hosil bo'ladi* (masalan, «ko'p TALABalar – TEST – topshirmoqdalar»). Ajdod mohiyat va avlod mohiyatlar orasidagi *aloqani aniqlash uzluksiz chiziq* orqali ifodalanadi (14.8- rasm). Bu aloqa tenglashtirilgan aloqa, deb ham yuritiladi. Avlod tomonidagi chiziqning oxirida nuqta



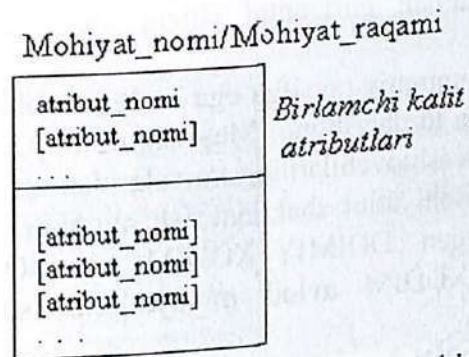
qo'yiladi. Ajdod mohiyatni belgilovchi aloqa o'zining aniqlovchisiga bog'liq bo'ladi. *Ajdod mohiyatni belgilovchi aloqa erkin*



yoki o'zining aniqlovchisiga bog'liq bo'lishi ham mumkin, bu bog'liqlik boshqa mohiyatlar bilan aloqasi orqali aniqlanadi.

*Uzluqli chiziq aniqlanmaydigan aloqani* bildiradi (14.9-rasm). Bu aloqani esa, *tenglashtirilmaydigan aloqa*, deb ham yuritiladi. Aniqlanmaydigan aloqadagi ajdod mohiyat, qaysidir anoqlovchili aloqada ajdod mohiyat sifatida kelmasa, u o'zining anoqlovchisiga bog'liq bo'lmaydi.

Atributlar mohiyat bloki ichida ro'yxat tarzida ifodalanadi. Birlamchi kalitni aniqlovchi atributlar ro'yxatning boshida keladi va qolgan atributlardan gorizontall chiziq orqali ajratiladi (14.10-rasm).



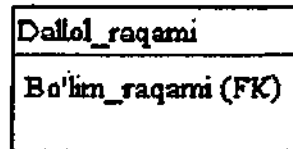
Mohiyatlar tashqi kalitlarga (Foreign Key) ham ega bo'lishi, tashqi kalitlar esa birlamchi kalitning tarkibida yoki birlamchi kalit sifatida yoki kalit bo'lmagan atribut sifatida ishlatilishi mumkin.

Mohiyatlar tashqi kalitlarga (Foreign Key) ham ega bo'lishi, tashqi kalitlar esa birlamchi kalitning tarkibida yoki birlamchi kalit sifatida yoki kalit bo'lmagan atribut sifatida ishlatilishi mumkin.

Tashqi kalit mohiyatlar blokidagi atributlar nomlarining ro'yxatida keltiriladi va qavsda FK harflari yoziladi (atributlar mohiyat bloki ichida ro'yxat tarzida ifodalanadi).

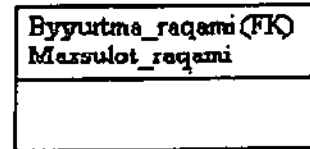
Kalit bo'lmagan atributning tashqi kalit sifatida kelishi

Dalol /12



Birlamchi kalit bo'lgan atributning kalit sifatida kelishi

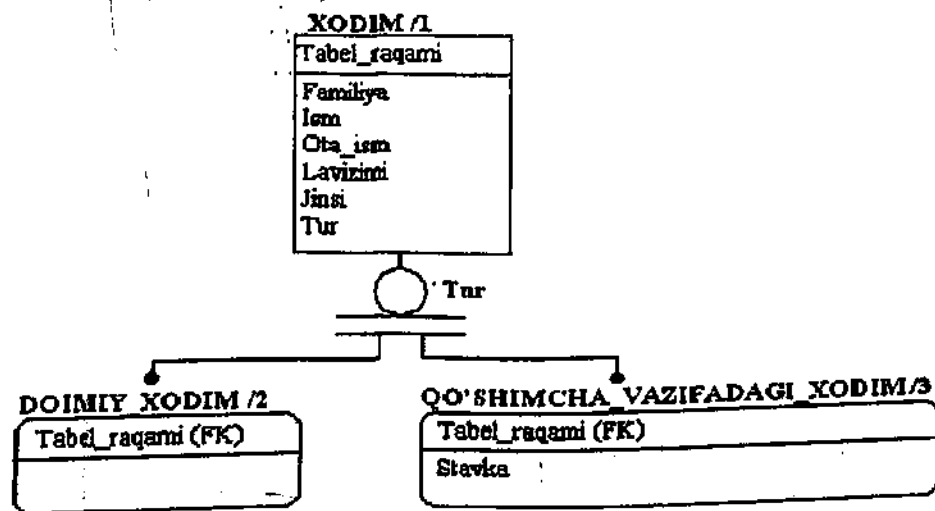
Sotib\_olishga\_byyurtma/5



14.11- rasm.

Birlamchi kalitni aniqlovchi atributlar ro'yxatni boshida keldi va qolgan atributlardan gorizontaal chiziq orqali ajratiladi (14.11- rasm).

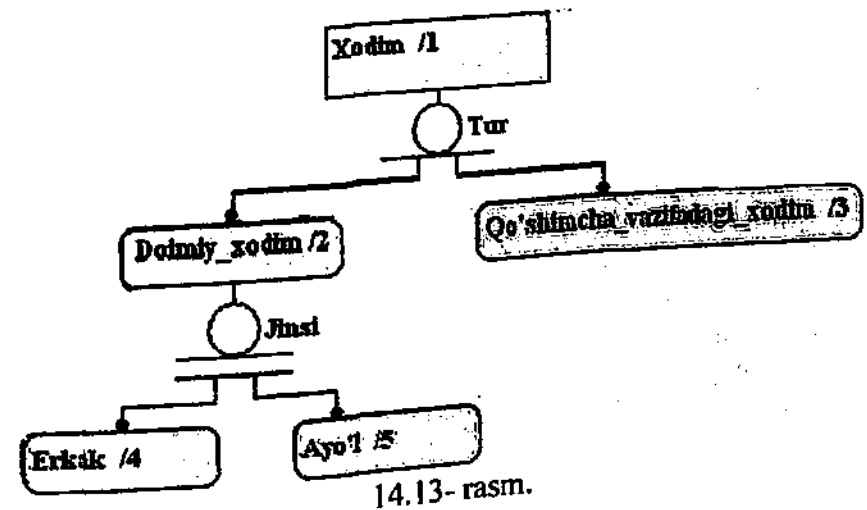
Vorilik shajarasini umumiy tavsifga ega bo'lgan mohiyatlarni birlashtirishdagi alohida turdan iborat. Masalan, doimiy xodimlar va qo'shimcha vazifada ishlovchilarning xossalariidan umumlashgan (ajdod) XODIM mohiyatini shakllantirish mumkin. Maxsus berilganlar tabaqalashgan DOIMIY\_XODIM va QO'SHIMCHA\_VAZIFADAGI\_XODIM avlod mohiyatlarda saqlanishi mumkin (14.12- rasm).



14.12- rasm.

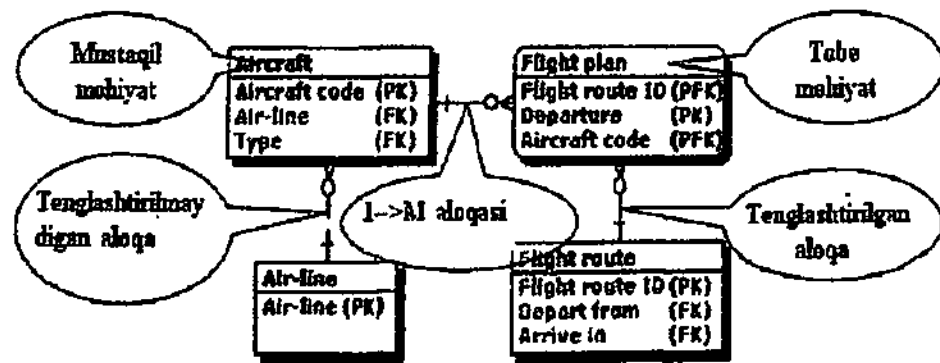
Diskriminator – bu ajdoddagi maxsus atribut bo'lib, bitta tabaqadagi mohiyatni boshqa tabaqadagi mohiyatdan ajratib ko'rsatadi. Keltirilgan misolda bu tur.

Agarda ajdodning avlodda o'ziga o'xshagan nusxalari mavjud bo'lmasa, tabaqa to'liq bo'lmaydi. Masalan, xodim maslahatchi bo'lishi mumkin, lekin u mohiyat sifatida iyerarxiyaga kiritilmagan bo'lishi mumkin. To'liq va to'liq bo'lmagan tabaqalar birgalikda kelishi ham mumkin (14.13- rasm).



14.13- rasm.

14.14- rasmda IDEF1X standarti bo'yicha mohiyat va aloqalarni tasvirlashga misol keltirilgan. Tashqi kalitning birlamchi kalit sifatida ishlatilishi PFK tarzida qavsda ko'rsatiladi.



14.14- rasm.

#### 14.8. Nazorat savollari

1. DEF0 uslubiyatining asosini nimalar tashkil qilgan?
2. Tijorat jarayonlarini baholash uchun funksional diagrammalarning imkoniyatlarini o'rganing va tahlil qiling.
3. Standart funksional blokning asosiy tarkibiy qismlarini sanab o'ting.
4. Qaysi diagrammalar kontekstli deb ataladi?
5. Millarning turlari nechta?
6. Qaysi jadvallar ajlod deb ataladi?
7. Qaysi diagrammalar avlod deb ataladi?
8. Funksional diagrammalarda ba'zi funksional bloklarning boshqalarga nisbatan ustunligi qanday ko'rsatiladi?
9. IDEFIX uslubiyatida berilganlar qaysi modelga tayanadi?
10. Avlod va ajdod mohiyatlarga misol keltiring.
11. IDEFIX uslubiyatida aloqalar quvvati necha turda bo'lishi mumkin?
12. Identifikatsiyalangan va identifikatsiya qilinmagan aloqalarga misol keltiring.
13. Birlamchi tashqi va birlamchi tashqi kalitlarni IDEFIX uslubiyatida asosida belgilang.

## AMALIY QISM

### XV- BOB. ArgoUML muhiti

#### 15.1. ArgoUML xususida

Dasturiy ta'minotni yaratish bilim nuqtayi nazaridan qiyin masaladir. Loyihachilar loyihalarni qo'lda kiritishlari kerak, ammo asosiy qiyinchilik – bu berilganlarni kiritishdan ko'ra qaror qabul qilishdir. Agar loyihachilar qaror qabul qilish imkoniyatlarini yaxshilasalar, bu yaxshi loyihaga olib keladi.

Zamonaviy CASE vositalari, shunday avtomatizatsiyalash-tirish va foydalanuvchining grafik interfeyslarini ta'minlaydiki, natijada loyihani qo'lda kiritish va uni kodga aylantirish vaqtini ancha qisqartiradi. Ular loyihachilarga asosan qaror qabul qilishda yordam beradi – loyiha diagrammalarini vizuallashtiradi va oddiy sintaktik tekshiruvlarni amalga oshiradi. Bundan tashqari, ko'plab CASE vositalari naqlni nazorat qilish va bir vaqtda loyihalash mexanizmlari taklif qilmoqda. Loyihaviy yechimlarni tahlil qilish masalasi, loyihalashda yetarlicha qo'llab-quvvatlanmadi.

Zamonaviy CASE vositalarining foydali tomoni shundan iboratki, ular birinchidan, vositalarning barcha funksiyalariga kirish imkoniyatini beradi. Ikkinchidan, loyihalash jarayonini qo'llab-quvvatlaydilar, chunki ular loyihachiga mashhur loyihalash uslubiyatlari usulidagi diagrammalarni kiritishga imkon beradi. Ammo ular, loyihachini loyihalash jarayoniga yo'naltirishda, jarayonni qo'llab-quvvatlamaydilar. Buning o'rniga loyihachilar odatda bo'sh sahifadan boshlaydilar va loyihaning har bir jihatini qamrab olishni unutmashliklari kerak bo'ladi.

ArgoUML – bu sohaga-yo‘naltirilgan loyihalash muhiti bo‘lib, u bilim yordamida ob‘yektga yo‘naltirilgan loyihalashni ta‘minlaydi. ArgoUML tijorat vositalaridagi ayrim avtomatizatsiyalashtirish funksiyalarini taqdim etadi, lekin asosiy e‘tiborni loyihachilarning bilim ehtiyojlarini qo‘llab-quvvatlovchi funksiyalarga qaratadi. Ushbu bilim ehtiyojlar quyidagi uchta bilim nazariyasi bilan tavsiflanadi:

- harakatdagi akslanish;
- yondashuvchan loyiha;
- tushunish va muammolarni hal qilish.

ArgoUML bevosita UML 1.4 standarti tasniflariga asoslangan.

### 15.2. Ob‘yektga yo‘naltirilgan tahlil va loyihalash

So‘nggi 30 yil ichida ob‘yektga yo‘naltirilgan tahlil va loyihalash dasturiy ta‘minotni ishlab chiqish timsoliga aylandi. Shu bilan birga, dasturiy ta‘minotni ishlab chiqish jarayonida barcha yaratuvchilarning fikrlash qobiliyatlarida katta o‘zgarishlar yuz bergan.

Ob‘yektlarni dasturlash tilida qo‘llab-quvvatlash Simula 67 tilidan boshlangan, ammo buni 1980- yillarda paydo bo‘lgan C++, Ada va Object Pascal kabi gibril tillardan olib tashlashga imkon berdi. Ushbu tillar ikkalasini ham qo‘llab-quvvatladi – ob‘yektga yo‘naltirilgan va protsedurali dasturlashni. Ob‘yektga yo‘naltirilgan dasturlash asosiy oqimga aylandi.

Ob‘yektga yo‘naltirilgan va protsedurali dasturlash tizimi dasturiy artefaktlar yordamida haqiqiy dunyoni simulyatsiya qilish sifatida ishlab chiqilgan va amalga oshirilgan.

Bu asos oddiy bo‘lgani bilan kuchli. Tizimni loyihalashda ob‘yektga yo‘naltirilgan va protsedurali dasturlash yondashuvidan foydalangan holda loyihalash mumkin va birinchi navbatda tizimni qurmasdan sinovdan o‘tgan yoki to‘g‘ri simulyatsiya qilingan.

1990-yillarda ob‘yektga yo‘naltirilgan tahlil va loyihalashni qo‘llab-quvvatlash vositalarining rivojlanishi, ushbu yondashuvni asosiy oqimga ko‘chirdi. Tizimlar juda ko‘p loyihalash qobiliyati bilan birlashganda ob‘yektga yo‘naltirilgan, protsedurali dasturlash va loyihalash yondashuviga asoslangan yuqori darajadagi vosita ilgari mavjud bo‘lgan imkoniyatga nisbatan yanada murakkab tizimlarni joriy etishga imkon berdi.

Ob‘yektga yo‘naltirilgan protsedurali dasturlash va loyihalash yondashuvini harakatga keltirgan yakuniy foydalanuvchi uchun undagi grafik interfeyslarini modellashtirish yaroqliligi bo‘ldi. Visual Basic va Java kabi ob‘yektga asoslangan va ob‘yektga yo‘naltirilgan grafik tillarning mashhurligi ushbu yondashuvning samaradorligini aks ettiradi.

### 15.3. ArgoUMLning rivojlanishi

1980- yillarda turli xil tadqiqot guruhlarini tomonidan bir qator ob‘yektga yo‘naltirilgan, protsedurali dasturlash va loyihalash jarayonlari uslubiyati va yozuvlari ishlab chiqilgan.

ArgoUML ob‘yektga yo‘naltirilgan tahlil va loyihalashda foydalanish uchun vosita va muhit sifatida ishlab chiqilgan dasturiy tizimdir. Shu ma‘noda u dasturiy ta‘minot tizimlarini modellashtirish uchun, asbob sifatida sotiladigan ko‘plab tijorat ishlariga o‘xshaydi. Bunday tizimlarga nisbatan, ArgoUMLning ko‘plab juda muhim farqlari bor:

1. Bu muhit tekin.
2. ArgoUML kognitiv psixologiya bo‘yicha tadqiqotlarga asoslanib, ob‘yektga yo‘naltirilgan dasturiy ta‘minot loyihachilari va me‘morlarining bilim ehtiyojlarini qo‘llab-quvvatlash orqali samaradorlikni oshiradigan yangi xususiyatlarni taqdim etadi.
3. ArgoUML – UML, XMI, SVG, OCL va boshqalar kabi ochiq standartlarni keng qo‘llab-quvvatlaydi.
4. ArgoUML 100 foiz sof Java dasturidir. Bu ArgoUMLni

barcha platformalarda ishlashiga imkon beradi, chunki Java platformasining ishonchli porti mavjud.

5. ArgoUML ochiq manbali loyihadir. Manbaning mavjudligi yangi avlodning dasturiy ta'minot loyihachilari va tadqiqotchilari endi ular ishlashlari mumkin bo'lgan, tasdiqlangan asosga ega CASE asboblari texnologiyalarining rivojlanishi va evolyutsiyasiga egaligida.

UML eng keng tarqalgan modellashtirish tili va Java eng samarali rivojlantirish platformalaridan biri hisoblanadi.

#### 15.4. ArgoUMLdan foydalanish

Biz avval mahsulotning hissi bilan tanishamiz, keyin tahlilga o'tamiz va sinov uchun rivojlanish jarayonini ko'ramiz. Mahsulotning har bir bo'lagi namoyish etilmaydi. Bu bo'laklar keyingi qismlarda ko'riladigan ma'lumotnomalarda keltiriladi.

Asosiy diagrammalar oxirida modelning holati – “.zargo” fayllari paydo bo'ladi. Bular ko'zda tutilmagan turli jihatlar bilan ishlashingiz va keyin ish joyingizdagi modelning to'g'ri holatiga qaytishingiz mumkin. Bular, “.zargo” fayllari oxirida aniqlanadi.

AKM (avtomatlashtirilgan kassa mashinasi) loyihasi turli xil narsalarni namoyish etish uchun amaliy ish sifatida tanlangan.

ArgoUML taklif qiladigan bir qancha modellashtirish jihatlari mavjud. Siz loyihangizni o'z ichiga oladigan katalog yaratishingiz mumkin. Katalogda o'zingiz his qilgan har qanday narsani nomlangan holda saqlashingiz mumkin. Tarkibini va har qanday aniq sabablarga yo'naltirilgan quyi kataloglarni ham nomlashingiz kerak.

CASE o'qitish AKM tizimi bo'ladi. Sizning shartli kompaniyangiz FlyByNight Industries. Siz ikkita rolni o'ynaysiz – bu loyiha menejeri va loyihalash tahlilchisi.

Biz misol sifatida ko'radigan loyiha AKM mahsuloti simulyatori bo'ladi. Sizning kompaniyangiz o'z ishini loyihalarda

qanday tashkil qilishi, odatda, har qanday narsa kabi siyosat bilan belgilanadi.

#### 15.5. Nima uchun ArgoUML boshqacha?

Biz yoqorida ArgoUML uchta bilim nazariyadan ilhomlanganligini aytib o'tdik. Endi shularni izohlaymiz.

##### *Harakatdagi akslanish*

Ushbu nazariya murakkab tizimlarning loyihachilari to'liq shakllangan loyihani tasavvur qilmasliklarini kuzatadi. Buning o'rniga, ular tayyor bo'lgunga qadar qisman loyihani qurishlari, baholashlari, aks ettirishlari va qayta ko'rib chiqishlari va yanada kengaytirishlari kerak.

Ishlab chiquvchilar loyiha bilan amaliy ish olib borar ekanlar, ular muammoli vaziyatning aqliy modelini yaxshilaydilar, shuning uchun ularning loyihasi yaxshilanadi.

##### *Yondashuvchan loyiha*

Bilim psixologiyasining nazariyasi shuni ko'rsatadiki, loyihachilar o'zlarining rejalarini amalga oshirishadi va buyurtma qilingan, shajaraviy tarzda ishlanmalarni tavsiflaydilar. Aslida ular mezonlarga asoslanib, ketma-ket vazifalarni tanlaydilar va kognitiv xarajatlarni qo'llaydilar.

Oddiy qilib aytganda, loyihachilar hatto o'zlarining rejalariga ham rioya qilmaydilar, balki muqobil variantlar orasidan aqlan arzon bo'lgan qadamlarni tanlaydilar.

##### *Tushunish va muammolarni hal qilish*

Bilim psixologiyasi doirasida loyihani vizualizatsiya qilish nazariyasi yotadi. Nazariya shuni ko'rsatadiki, loyihachilar kerak.

Muammo shundan iboratki, vaziyatning aqliy modeli va yechim yoki tizimning rasmiy modeli o'rasidagi bo'shliqni bartaraf etish.

Bu nazariya dasturchilarga foyda keltirishini ko'rsatgan:

1. Bunday dasturning sintaktik dekompozitsiyasi davlat ro'yxatidan o'tish, nazorat qilish kabi bir necha vakolatxonalaridagi oqim va ma'lumotlar oqimi. Bular dasturchiga elementlar va munosabatlarni yaxshiroq aniqlash imkonini beradi.

Muammo va uning yechimda, va shu bilan modellar va ishlaydigan tizim modellarining holati o'rasidagi xaritalashni osonroq yaratishda.

2. Loyihachilarning yechimlarni shakllantirish qobiliyatini yaxshilaydigan vaziyat modelining tanish jihatlarni qo'llay olishda.

### 15.6. ArgoUMLda amaliy dastur

ArgoUML ushbu nazariyalarni bir qator texnikalar yordamida amalga oshiradi:

1. Foydalanuvchiga loyihani turli xil raqamlarda ko'rishga imkon beradigan foydalanuvchi interfeysining loyihasi va foydalanuvchiga bir qator muqobil yo'nalishlar orqali maqsadlarga erishishga imkon berish.

2. Loyiha vositasi bilan parallel ravishda ishlaydigan jarayonlardan foydalanish, joriy loyihani baholash. "Eng yaxshi amaliyot" loyihasi qanday ishlashi mumkinligi modellarga bog'liqligi.

3. Loyiha tanqidchilarining takliflarini foydalanuvchiga etkazish uchun vazifalar ro'yxatidan foydalanish, shuningdek foydalanuvchi kelajak harakatlar uchun UML asosidagi ob'yektga yo'naltirilgan, protsedurali dasturlash va loyihalash yondashuvining maydonlarni yozib olishiga ruxsat berish.

4. Nazorat ro'yxatlarini foydalanish, murakkab jarayon orqali foydalanuvchi hidoyat qilish.

ArgoUML hamma interfeyslari ochiq standartlardan foydalanishiga harakat qiladi. Ochiq standartlarga rioya qilishning asosiy afzalligi shundaki, u ilovalar o'rasida o'zaro ishlashni osonlashtiradi va kerak bo'lganda bitta dasturdan boshqasiga o'tish imko -

niyatini beradi.

UML modeli – bizga ArgoUMLda yaratgan modelimizni olish va uni boshqa vositaga import qilish imkonini beradi. Bu UML bilan aloqa uchun standartga rioya qilish maqsadiga erishishga imkon beradigan afzallikdir.

### *Ob'yektni cheklash tili (Object\_Constraint\_Language - OCL)*

Ob'yektni cheklash tili UML modellariga taalluqli qoidalarni tavsiflash uchun deklarativ tildir. U IBMda ishlab chiqilgan va hozirda UML standartining bir qismiga aylangan. Dastlab OCL UML uchun faqat rasmiy tasnif tilining kengaytmasi edi.

OCL endi har qanday Meta-Object Facility (MOF)ga mos metamodeldir, shu jumladan UML bilan ishlatilishi mumkin.

Ob'yektni cheklash tili – bu har qanday MOF modelida yoki metamodelda cheklov va ob'yekt so'rovi ifodalarini ta'minlaydigan aniq matn tili, aks holda diagramma yozuvlar bilan ifodalanishi mumkin emas.

### 15.7. ArgoUML ochiq manba

ArgoUML ochiq manbali loyihadir. Bu shuni anglatadiki, har kim manba kodining bepul nusxasiga ega bo'lishi mumkin, uni o'zgartirishi, yangi maqsadlarda foydalanishi va h.k. Yagona (asosiy) majburiyat – biz kodimizni topshirishimiz kerak.

Biz qila oladigan va qila olmaydigan narsalarning aniq tabiati loyihadan loyihaga farq qiladi, ammo tamoyil bir xil.

Afzalligi shundaki, ArgoUML kabi kichik loyiha to'satdan ko'plab qo'shimcha yordam uchun ochiq dasturni qanday yaxshilash mumkinligi haqida o'z g'oyalarini aniqlay oladiganlar uchun mihim.

ArgoUMLga hisoblash zaxirasi o'rtacha miqdorda kerak. Amaliy ishlashga qodir bo'lgan har qanday kompyuter va grafik foydalanuvchi interfeysiga ega tizim yetarli bo'ladi.

### 15.8. Loyihalar bilan ishlash

Umumiy diagrammalar tahrirlash chizimasida kerakli model elementini tanlash uchun asboblarning chizimasi va kerakli holatda diagrammani bosish yordamida chiziladi.

“Diagrammani yaratish” menyusi orqali model - diagramma tanlanadi va model elementlari diagramma oynasiga qo‘shiladi. Kerakli elementni tanlab, tugmani bosib diagrammaga qo‘shish mumkin.

UML model elementlari bilan bir qatorda tahrirlash chizimasi asboblarning chizimasi umumiy chizilgan moslamalar/to‘rtburchaklarni ta‘minlaydi, doiralar, chiziqlar, ko‘pburchaklar, egri chiziqlar va matn diagrammalar haqida qo‘shimcha ma‘lumot berish uchun ishlatiladi.

ArgoUML har safar argument sifatida loyiha faylisiz ishga tushirilganda, yangi bo‘sh loyiha yaratiladi.

Model va ikkala bo‘sh diagrammani explorerda ko‘rish mumkin, bu sizning modelingiz bo‘ylab harakatlanish uchun asosiy vositadir.

Bir lahzaga faraz qilaylik, bu yerda biz yangi sotib olish tizimini modellashtirishni boshlamoqchimiz. Biz modelimizga "xarid qilish modeli" nomini bermoqchimiz va uni faylda saqlamoqchimiz. Bunun uchun, hammaga ma‘lum usuldan foydalanamiz: Fayl → “Qanday saqlash kerak” → yangi darcha ochiladi → saqlash joyi tanlanadi → saqlanadi.

### 15.10. Talablarni jamlash

Talablarni qo‘lga olish – bu buyurtmachining taklif qilingan tizimdan nimani xohlashini aniqlash jarayoni. Ushbu bosqichdagi kalit shundaki, biz muammo domenida bo‘lamiz. Ushbu bosqichda biz hamma narsani buyurtmachining nuqtayi nazaridan va uning tilida tasvirlashimiz kerak.

Talablarni qo‘lga kiritishda eng katta xavf – bu mumkin bo‘lgan yechimlar nuqtayi nazaridan o‘ylashni boshlashdir. Bu tahlil bosqichigacha kutish kerak. Tahlil talablar bosqichining natijasini olish va uni ko‘rib chiqilgan yechim tilida qayta tiklash davrining bosqichlaridan biri bo‘ladi.

Muammoni kichikroq qilib ajratganimizda, biz yana talablar jarayoniga qaytishimiz mumkin.

Biz, albatta har bir yaqinlashuv bo‘yicha talablar bosqichidan qaytamiz.

Biz hal qilayotgan muammoning yuqori darajadagi ko‘rinishi va biz funksional imkoniyatlarning asosiy sohalari bilan boshlaymiz. Har qanday yechimga murojaat qilish kerak. Bu bizning ko‘rish hujjatimiz bo‘ladi va bir necha sahifadan iborat bo‘lishi kerak.

Masalan, AKMning yuqori darajadagi ko‘rinishida, uni quyidagicha qo‘llab-quvvatlash kerak bo‘lishi mumkin:

1. Naqd depozit, naqd pul olish va mijozlar tomonidan hisob so‘rovlari.
2. Bank muhandislari tomonidan jihozlarga texnik xizmat ko‘rsatish, depozitlarni yuklash va mahalliy bank filiali tomonidan naqd pul bilan ta‘minlash.
3. Bankning markaziy kompyuter yuborgan barcha faoliyati uchun audit o‘tkazish.

Ushbu yuqori darajadagi nuqtayi nazardan biz tizimning asosiy faoliyatini va ushbu faoliyat bilan shug‘ullanadigan tashqi agentlar (odamlar, uskunalar)ni ajratib olishimiz mumkin. Ushbu tadbirlar foydalanish holatlari diagrammasi sifatida tanilgan va tashqi agentlar aktyorlar sifatida tanilgan.

Aktyorlar odamlar yoki mashinalar bo‘lishi mumkin. Amaliy nuqtayi nazardan, har qanday mashinaning orqasida turgan manfaatdor tomonni bilish kerak, chunki ular faqat talablarni olish jarayoni bilan shug‘ullanishlari mumkin.

Foydalanish holatlari tizim uchun muhim faoliyat bo'lishi kerak. Masalan, mijozning bankomat mashinasidan foydalanishi foydalanish holatidir.

Bizda naqd depozitni o'z ichiga olgan sub-use – naqd pul olish va hisob bo'yicha so'rov holatlari bo'lishi mumkin.

Qattiq va tezkor qoida yo'q. Ba'zilar oz songa nisbatan katta foydalanish holatlarini afzal ko'rishadi, boshqalar kichikroq foydalanish holatlarini afzal ko'rishadi. Foydali qoida shundaki, har qanday amaliy loyiha 30dan ortiq foydalanish holatlarini talab qilishi mumkin, agar ko'proq holatlar kerak bo'lsa, uni alohida loyihalarga bo'lish kerak.

Keyin biz foydalanish holatlari va aktyorlar o'rasidagi munosabatni bir yoki bir nechta foydalanish diagrammalarida ko'rsatamiz. Buning uchun katta loyiha, bir nechta diagramma kerak bo'ladi. Odatda, bitta diagrammada tegishli foydalanish holatlari guruhlarini ko'rsatiladi.

Keyin har bir foydalanish holatining batafsil tavsifini berishimiz kerak. Bu uning me'yoriy xatti-harakatlarini, muqobil xatti-harakatlarini va har qanday dastlamki va keyingi shartlarni qamrab oladi. Bu turli xil foydalanishlar "CASE tasnifi" yoki "CASE ssenariysi" deb nomlanuvchi hujjatda qayd etiladi.

Va nihoyat, foydalanish holatlari tabiatan funksional bo'lganligi sababli, bizga funksional bo'lmagan talablarni (imkoniyatlar, ishlash, ekologik ehtiyojlar va boshqalar) olish uchun hujjat kerak bo'ladi. Ushbu talablar qo'shimcha talablar tasnifi, deb nomlanuvchi hujjatda saqlanadi.

### 15.11. Jarayon bosqichlari

Talablarni ta'qib qilish jarayonidagi qadamlarni quyidagicha umumlashtirish mumkin.

1. Muammoning umumiy ko'rinishini va uni hal qilishning kerakli xususiyatlarini qayd etuvchi hujjat.

2. Namoyish hujjatidan foydalanish holati va aktyorlarini aniqlash va ularning munosabatlarini ko'rsatish yoki qo'shimcha foydalanish diagrammalari.

3. Oddiy va muqobil xatti-harakatlarni o'z ichiga olgan har bir (dastlabki va keyingi) foydalanish holati uchun batafsil foydalanish shartlarini berish.

4. Qo'shimcha talablar tasnifida barcha funksional bo'lmagan talablarni yozib olish.

Har qanday yaqinlashuvchi rivojlanish jarayonini biz birinchi o'ringa qo'yamiz va eng muhim foydalanish holatlarining asosiy xatti-harakati erta takrorlashdan iborat.

Aksariyat zamonaviy talablarni qo'lga kiritish jarayonlarida, buyurtmachining nufuzli vakili butun jarayon davomida to'liq ishtirok etishi zarur.

Talablarni qo'lga kiritish jarayonining deyarli barcha natijalari hujjatli film bo'lishi mumkin. Yagona diagramma – bu foydalanish holatlari va aktyorlar o'rasidagi munosabatlarni ko'rsatadigan ish diagrammasidir.

### 15.12. Namoyishli hujjat

Ushbu hujjatning odatiy bo'limlari quyidagicha bo'ladi:

- Maqsadlar. Biz nimaga erishmoqchimiz va unga qanday erishishni xohlaymiz.

- Bozor konteksti yoki shartnomaviy kelishuvlar. Bozorni rivojlantirish uchun bu quyidagilarni ko'rsatishi kerak – maqsadli bozorlar, raqobatbardosh ajraluvchilar, qiziqarli voqealar va boshqalar. Shartnomani rivojlantirish uchun bu asosiy shartnoma yuridiklarini tushuntirish kerak.

- Manfaatdor tomonlar. Tizimning foydalanuvchilari (keng ma'noda). Ularning ko'pchiligi aktyorlar uchun xarita qiladi, yoki aktyorlar ichida xaritalarni nazorat qiluvchi uskunalar o'rnatadi.

- Asosiy xususiyatlar. Eng yuqori darajada ular qaysidir muammoning asosiy funksional jihatlari ko'rsatadi va kerakli

yechimlarni taklif qiladi. Ular asosan foydalanish holatlariga to'g'ri keladi. Buhda ustuvorlik darajasini o'rnatish foydalidir.

- Cheklovlar. Tizimning funksional bo'lmagan parametrlarining yuqori darajadagi ko'rinishi. Bular qo'shimcha talablar tasnifida batafsil ishlaydi.

- Ilova. Ushbu tasavvurni qondirish uchun zarur bo'lgan aktyorlar ro'yxati va foydalanish holatlari. Bu keng qamrovni ta'minlash uchun oldingi bo'limlardan bularga havola.

## XVI- BOB. FOYDALANISH HOLATLARI DIAGRAMMASI

### 16.1. Nazariy asoslar

UML diagrammalariga nisbatan «Use Cases» iborasini mazmunan to'g'ri bo'lishi uchun "Foydalanish holatlari" deb tarjima qilsak bo'ladi.

**Foydalanish holati** – bu ba'zi bir tashqi ob'jekt (aktyor) tashabbuskorligiga javoban tizim tomonidan bajariladigan harakatlarning (tranzaksiyalar) ketma-ketligi. Foydalanish holati foydalanuvchi va tizim o'rasidagi odatiy o'zaro harakatni tavsiflaydi. Eng oddiy holatda, foydalanish holati *foydalanuvchi bilan amalga oshirmoqchi bo'lgan funksiyalarni muhokama qilish jarayonida* aniqlanadi.

**Aktyor** – bu foydalanuvchi tizimga nisbatan o'ynaydigan rol. Aktyorlar muayyan shaxslar yoki ishlarning nomlari emas, balki rollardir. Foydalanish holatlarining diagrammalarida ular inson qiyofalari sifatida tasvirlangan bo'lsa-da, aktyor – shu tizimdagi ba'zi ma'lumotlarga muhtoj bo'lgan tashqi tizim ham bo'lishi mumkin. *Foydalanish holati* diagrammasida harakatlanuvchi shaxslarni faqat ba'zi foydalanish holatlari zarur bo'lganda ko'rsatilishi kerak.

Harakatlanuvchi shaxslar uchta asosiy turga bo'linadi – tizim foydalanuvchilari, ushbu tizim bilan o'zaro aloqada bo'lgan boshqa tizimlar va vaqt. Agar tizimdagi ayrim hodisalarni boshlash vaqtga bog'liq bo'lsa, u aktyorga aylanadi.

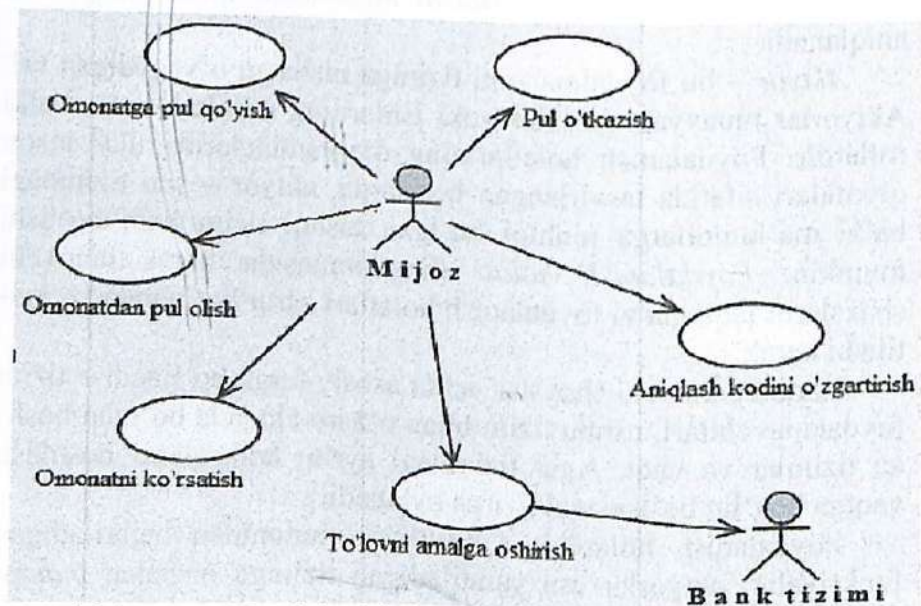
Foydalanish holatlari – bu tizim tomonidan bajariladigan funksiyalar, aktyorlar esa yaratiladigan tizimga nisbatan *manfaatdor shaxslardir (stakeholders)*. Bunday diagrammalar qaysi

harakatlanuvchi shaxslar foydalanish holatlarining tashabbuskori ekanligini ko'rsatadi. Undan, shuningdek, aktyor qachon foydalanish holatidan ma'lumot ola boshlasi ham ko'rsatiladi.

Foydalanish holatlarini vizual tasvirlash uchun dasturiy ta'minotni ishlab chiqish jarayonining asosiy elementlari sifatida foydalanish holatlarining diagrammalari qo'llaniladi. 16.1- rasmda bankomat uchun bunday diagrammaga misol Avtomatlashgan Kassa Mashinasi (AKM, Automated Teller Machine) ko'rsatilgan.

Ushbu diagrammada inson figuralari harakatlanuvchi shaxslar (aktyor)ni, oval belgilar foydalanish holatlarini, chiziqlar va millar – aktyorlar va foydalanish holatlari o'rasidagi turli xil aloqalarni bildiradi.

Ushbu diagrammada ikkita aktyor ko'rsatilgan: mijoz va kredit tizimi. Shuningdek, modellashtirilayotgan tizim tomonidan amalga oshiriluvchi oltita asosiy faoliyat mavjud, bular – pul o'tkazish, hissa qo'shish, hisobdan pul oling, balansni ko'rsating, aniqlash kodini o'zgartirish va to'lovni amalga oshirish.



16.1- rasm. Foydalanish holatlari diagrammasiga misol

Barcha foydalanish holatlari, u yoki bu tarzda, tizimning funksionalligi uchun tashqi talablar bilan bog'lanadi. Foydalanish holatlari har doim tizimning aktyorlari bilan birgalikda tahlil qilinishi kerak, shu bilan birga, foydalanuvchilarning haqiqiy masalalarini aniqlaydi va ushbu muammolarni hal qilishning muqobil usullarini ko'rib chiqadi.

Aktyorlar foydalanish holatiga nisbatan turli xildagi rollarni o'ynashi mumkin. Ular uning natijalaridan bahramand bo'lishlari yoki o'zlari bevosita ishtirok etishlari mumkin. Aktyorning turli rollarining ahamiyati, uning aloqalari qanday ishlatilishiga bog'liq.

Foydalanish holatlari diagrammalarining o'ziga xos maqsadi – bu foydalanish holatlarini (tizim doirasiga kiradigan hamma narsa), aktyorlarni (bu sohadan tashqaridagi hamma narsa) va ular o'rasidagi aloqalarni hujjatlashtirishdir. Foydalanish holatlarining diagrammalarini ishlab chiqishda quyidagi qoidalarga rioya qilishga harakat qilish kerak:

- ✓ aktyorlar o'rasidagi aloqalarni modellashtirmang. Ta'rifga ko'ra, aktyorlar tizim doirasidan tashqarida. Bu shuni anglatadiki, ular orasidagi aloqalar ham uning vakolatiga kirmaydi;
  - ✓ bevosita ikkita foydalanish holatini to'g'ridan-to'g'ri uzluksiz mil bilan bog'lamang. Ushbu turdagi diagrammalar ularni bajarish tartibini emas, balki faqat tizim uchun qanday foydalanish holatlari mavjudligini tavsiflaydi. Foydalanish holatlarini bajarish tartibini akslantirish uchun faoliyat diagrammalaridan foydalaniladi;
  - ✓ foydalanish holati aktyor tashabbusi bilan amalga oshirilishi kerak. Bu shuni anglatadiki, aktyordan boshlanib, foydalanish holatida tugaydigan uzluksiz mil bo'lishi kerak.
- Tashqi hodisalar foydalanish holatlarini aniqlash uchun yaxshi manba hisoblanadi. Tizim qandaydir tarzda javob berishi kerak

bo'lgan, tashqi dunyoda sodir bo'layotgan barcha voqealarni sanab o'tishdan boshlash kerak. Qandaydir aniq hodisa foydalanuvchi aralashuvini talab qilmaydigan tizim etiborini keltirib chiqarishi yoki aksincha, foydalanuvchi etiborini keltirib chiqarishi mumkin. Javob berish kerak bo'lgan hodisalarni aniqlash, foydalanish holatlarini aniqlashga yordam beradi.

Foydalanish holatlari, tizim nima qilishi kerakligini tasvirlay boshlaydi. Tizimni ishlab chiqish uchun aniq *tafsilotlar* kerak bo'ladi. Ushbu tafsilotlar "voqealar oqimi" (flow of events), deb nomlangan hujjatda tasvirlanadi. Voqealar oqimining maqsadi, foydalanish holatida amalga oshiriladigan berilganlarga islov jarayonini hujjatlashtirishdir. Ushbu hujjat, tizim foydalanuvchilari va tizimning o'zi nima qilishlari haqida batafsil ma'lumot beradi.

#### **Tavsif**

Har bir foydalanish holatida u nima qilishi haqida qisqacha tavsif bo'lishi kerak. Masalan, AKM tizimining "pul o'tkazish" dan foydalanish holatida quyidagi tavsif bo'lishi mumkin:

"Pul o'tkazish" dan foydalanish holati mijozga yoki bank xodimiga pulni bitta talab qilingan hisob-varaqadan yoki omonat hisob varag'idan boshqasiga o'tkazishga imkon beradi.

#### **Old shartlar**

*Foydalanish holatining old shartlari* – bu foydalanish holati o'z-o'zidan amalga oshirilishidan oldin bajarilishi kerak bo'lgan shartlardir. Masalan, bunday holat boshqa foydalanish holatini bajarish yoki foydalanuvchi uni ishga tushirish uchun zarur bo'lgan kirish huquqlari talab qilinadigan shart. Hamma foydalanish holatlari ham old shartlarga ega emas.

#### **Asosiy va muqobil voqealar oqimlari**

Foydalanish holatlarining o'ziga xos tafsilotlari asosan va muqobil voqealar oqimida tasvirlanadi. Voqealar oqimi foydala-

nish holatlariga kiritilgan funkcionallikni bajarish paytida nima bo'lishi kerakligini bosqichma-bosqich tavsiflaydi. Voqealar oqimi, tizim buni qanday qilishiga emas, balki nima qilishiga e'tibor beradi va bularning barchasini foydalanuvchi nuqtayi nazaridan tavsiflaydi. Asosiy va muqobil voqealar oqimlari quyidagi tavsiflarni o'z ichiga oladi:

- foydalanish holatini ishga tushirish usulini;
- foydalanish holatini bajarishning turli usullarini;
- me'yoriy yoki asosiy foydalanish holatidagi hodisalar oqimini;
- hodisalarning asosiy oqimidan chetga chiqishini (muqobil oqimlar, deb ataladi);
- xato oqimlarni;
- foydalanish holatini tugatish usulini.

Masalan, AKMdan "Pul olish"da, foydalanish holatidagi voqealar oqimi quyidagicha ko'rinishi mumkin:

#### *Asosiy oqim*

1. Foydalanish holati mijoz o'z kartasini AKMga kiritgandan so'ng boshlanadi.
2. AKM salomni chiqaradi va mijozga shaxsiy identifikatsiya raqamini kiritishni taklif qiladi.
3. Mijoz raqamni kiritadi.
4. AKM kiritilgan raqamni tasdiqlaydi. Agar raqam tasdiqlanmagan bo'lsa, A1 hodisalarining muqobil oqimi bajariladi.

5. AKM mavjud harakatlar ro'yxatini ko'rsatadi:

- hisob raqamiga pul qo'yish;
- hisobdan pul olish;
- pul o'tkazish.

6. Mijoz "pul olish" bandini tanlaydi.

7. AKM qancha pul olish kerakligini so'raydi.

8. Mijoz kerakli miqdorni kiritadi.

9. AKM hisobda yetarli pul bor-yo'qligini aniqlaydi. Agar pul yetarli bo'lmasa, muqobil A2 oqimi amalga oshiriladi. Agar

mablagi tasdiqlash paytida xatolar yuzaga kelsa, E1 xatolar oqimi amalga oshiriladi.

10. AKM mijozning hisobvarag'idan kerakli miqdorni ushlab qoladi.
11. AKM mijozga kerakli miqdordagi naqd pulni beradi.
12. AKM mijozga uning kartasini qaytaradi.
13. AKM mijoz uchun chekni chop etadi.
14. Foydalanish holati tugaydi.

*A1 muqobil oqim. Noto'g'ri identifikatsiya raqamini kiritish.*

1. AKM mijozga identifikatsiya raqami noto'g'ri kiritilganligi to'g'risida xabar beradi.
2. AKM mijozga uning kartasini qaytaradi.
3. Foydalanish holati tugaydi.

*A2 muqobil foydalanish. Hisobda yetarli pul yo'q.*

1. AKM mijozga uning hisobidagi pul yetarli emasligi to'g'risida xabar beradi.
2. AKM mijozga uning kartasini qaytaradi.
3. Foydalanish holati tugaydi.

*E1 Xato oqimi. So'ralgan miqdorni tasdiqlashda xato.*

1. AKM foydalanuvchiga so'ralgan mablagi tasdiqlashda xatolik yuz berganligini aytadi va unga bank mijozlarga xizmat ko'rsatish telefon raqamini beradi.
2. AKM xato ma'lumotlarini xatolar jurnaliga kiritadi. Har bir yozuvda xato sanasi va vaqti, mijozning ismi, hisob raqami va xato kodi mavjud.
3. AKM mijozga uning kartasini qaytaradi.
4. Foydalanish holati tugaydi.

### **Oxirgi shartlar**

*Oxirgi shartlar* - bu foydalanish holati tugagandan so'ng har doim bajarilishi kerak bo'lgan shartlar. Masalan, foydalanish

holatining oxirida siz biron bir kalitni belgilash katakchasi bilan belgilashingiz mumkin. Ushbu turdagi ma'lumotlar oxirgi shartlarning bir qismidir. Old shartlarda bo'lgani kabi, oxirgi shartlardan foydalanib, tizimdan foydalanish holatlarining bajarilish tartibi to'g'risida ma'lumot kiritishimiz mumkin. Agar, masalan, bitta foydalanish holatidan keyin har doim boshqasi bajarilishi kerak bo'lsa, buni oxirgi shart sifatida ta'riflash mumkin. Bunday shartlar har bir foydalanish holatida mavjud emas.

### **16.2. Foydalanish holatlari va aktyorlar o'rasidagi aloqalar**

UML tilida foydalanish holatlarining diagrammalarida diagramma elementlari orasidagi bir necha turdagi aloqalar qo'llab-quvvatlanadi:

- kommunikatsiya (communication);
- kiritish (include);
- kengaytma (extend);
- umumlashtirish (generalization).

*Kommunikatsiya aloqasi* – bu foydalanish holatlari va aktyor o'rasidagi bog'liqlik. UML tilida kommunikatsiya aloqalari bir yo'nalishli yushma (uzluksiz milli chiziq) yordamida ko'rsatiladi. Milning yo'nalishi aloqa tashabbuskorini aniqlashga imkon beradi.

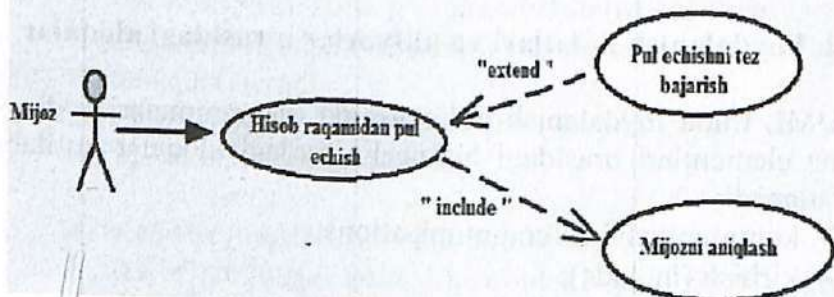
*Kiritish aloqasi* bir nechta foydalanish holatlarida takrorlanadigan tizim xatti-harakatlarining biron bir qismi mavjud bo'lgan holatlarda qo'llaniladi. Bunday bog'lanishlar yordamida odatda ko'p ishlatiladigan funktsionallik modellashtiriladi

AKM misolida, "Pul olish" va "Hisob raqamiga pul qo'yish"-dan foydalanish tranzaksiyalarini amalga oshirishga ruxsat berishdan oldin mijoz va uni aniqlash raqamini tasdiqlash kerak. Ularning har biri uchun autentifikatsiya jarayonini batafsil bayon qilish o'rniga, ushbu funktsiyani "mijozni autentifikatsiya

qilish"deb nomlangan o'z foydalanish holatimizga qo'yish mumkin.

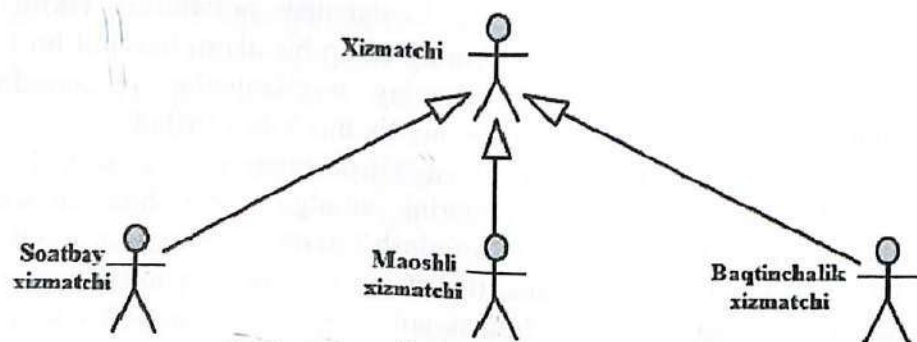
*Kengayish aloqasi* tizimning me'yoriy xatti-harakatlaridagi o'zgarishlarni tavsiflashda qo'llaniladi. Bu, foydalanish holatiga, faqat zarur bo'lganda boshqaning funktsional imkoniyatlaridan foydalanishga imkon beradi.

UML tilida kiritish va kengaytma aloqalari 16.2- rasmda ko'rsatilganidek, tegishli qat'iy turlar bilan bog'liqlik sifatida ko'rsatiladi.



16.2- rasm. Foydalanish va kengaytirish aloqalari

*Umumlashtirish aloqalari* yordamida, bir nechta aktyorlar umumiy xususiyatlarga ega ekanligi ko'rsatadi. Masalan, mijozlar ikki xil bo'lishi mumkin – korporativ va shaxsiy. Ushbu aloqani 16.3- rasmdagi belgilash yordamida modellashtirish mumkin.

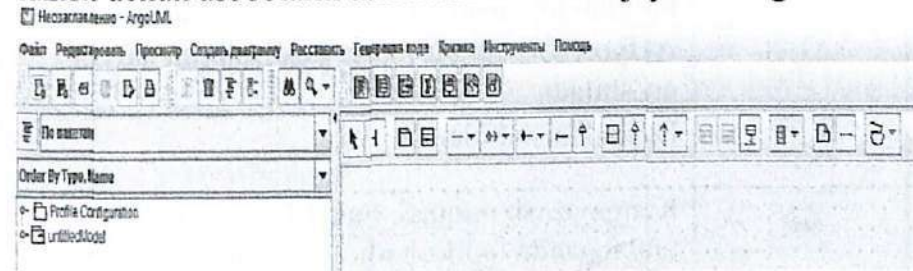


16.3- rasm. Aktyorlarni umumlashtirish

Har doim ushbu turdagi aloqalarni yaratishga hojat yo'q. Umuman olganda, agar bir turdagi aktyorning xatti-harakati boshqa turdagi xatti-harakatlardan farq qilsa, ular kerak bo'ladi, chunki bu tizimga ta'sir qiladi. Agar ikkita qism turlar bir xil foydalanish holatlaridan foydalansa, aktyorning umumlashtirishini ko'rsatish shart emas.

Foydalanish holatlari dasturiy ta'minot talablarini shakllantirish bosqichida zarur vositadir. Har bir foydalanish holati, tizim uchun yashirin talabdir va u aniqlanmaguncha, uni amalga oshirishni rejalashtirish mumkin emas.

ArgoUML muhitini yuklab olgandan so'ng, 16.4- rasmda ko'rsatilgan shaklga (faqat bo'limlar nomi bo'lgan asosiy oynaning yuqori chap burchagi) chiqadi, bu yerda sinf diagrammasini tuzish uchun asboblarni tahrirlash chizimasi joylashtirilgan.



16.4- rasm. ArgoUML muhitining qisni

### 16.3. Foydalanish holatlari diagrammasidagi asboblari








Foydalanish holatlari diagrammasini yaratish vositalariga o'tish uchun "Создать диаграмму" menyusi buyruqlari orasida "Диаграмма вариантов использования" (foydalanish holatlari) buyrug'i tanlanadi. Bunday holda, 16.5- rasmda ko'rsatilgan foydalanish holatlari diagrammasi vositalari tahrirlash panelida paydo bo'ladi.



16.5- rasm. Foydalanish holatlari diagrammasi asboblari

Asbob tasvirlangan tugma oshkormas rabishda bir marta bosilsa, undan bir marta foydalanish mumkin. Asbobdan ko'p marta foydalanish ucun unu ikki marta bosish yoki CTRL tugmasini bosgan holda, uni bir marta bosish kerak. Foydalanish holatlari diagrammasi asboblarning tavsifi 16.1- jadvalda keltirilgan.

16.1- jadval. Foydalanish holatlari diagrammasi asboblari

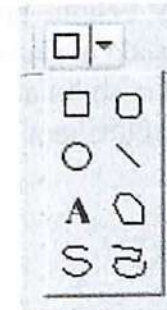
Asbob	Tavsifi
 Select	Asbobdan ko'p marta foydalanishni bekor qilish
 Broom	Supurgi. Diagrammada bir vaqtning o'zida bir nechta elementlarni harakatga keltiradi. Diagramma elementlarini tanlangan chiziq chiq bo'ylab joylashtirish uchun ishlatiladi.
	Aktyor. Tugma bir marta bosilsa bitta aktyor qo'shiladi.
	Foydalanish holati.
	Kengaytirish nuqtasi. Faqat foydalanish holati tanlanganda faollashadi. Ikki marta bosish va klaviaturadan foydalanish orqali tahrirlash mumkin.
	Diagramma elementiga izoh qo'shish uchun ishlatiladigan asbob.
	Diagramma elementini izoh bilan bog'lash uchun ishlatiladigan asbob.

Shakl chizish asbobi 16.6- rasmda keltirilgan.











16.6- rasm. Shakl chizish asbobi

Shu tugma bosilsa, 16.7- rasmdagi suzuvchi menyu chiqadi. Unda 16.2- jadvalda tavsivlangan asboblarning suzuvchi piktoqrammasi chiqadi.




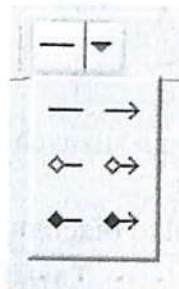
16.7- rasm. Chish uchun suzuvchi menyu

16.2- jadval. Foydalanish holati diagrammasini chizish asboblari

Asbob	Tavsifi
	To'rtburchak
	Aylanma shaklidagi to'rtburchak
	Овал
	Chiziq
	Matn
	Ko'pburchak
	Ochiq splayn. So'nggi nuqtaga ikki marta bosiladi
	Tanlangan nuqtalardan o'tkazilgan chiziqlar to'plami. Chiziqlar kesishishi mumkin.





ArgoUML qo'llab-quvvatlovchi yushmalar 16.8- rasmda kel-

tirilgan. Yo'nalishli yoki yo'nalishsiz (oddiy yushma, agregatsiya yoki kompozitsiya) yushma turini tanlash uchun  shu tugmalarning o'ng tomondagisini bosish kerak. Keyin 16.8- rasmdagi "—" belgidan boshqa ixtiyotiy belgini bossak, tanlangan yushma foydalanish holati diagrammasida akslanadi. Chiziq tobe elementdan aktyorga tortiladi. Foydalanish holati diagrammasidagi asboblarning tavsifi 16.3- jadvalda keltirilgan.



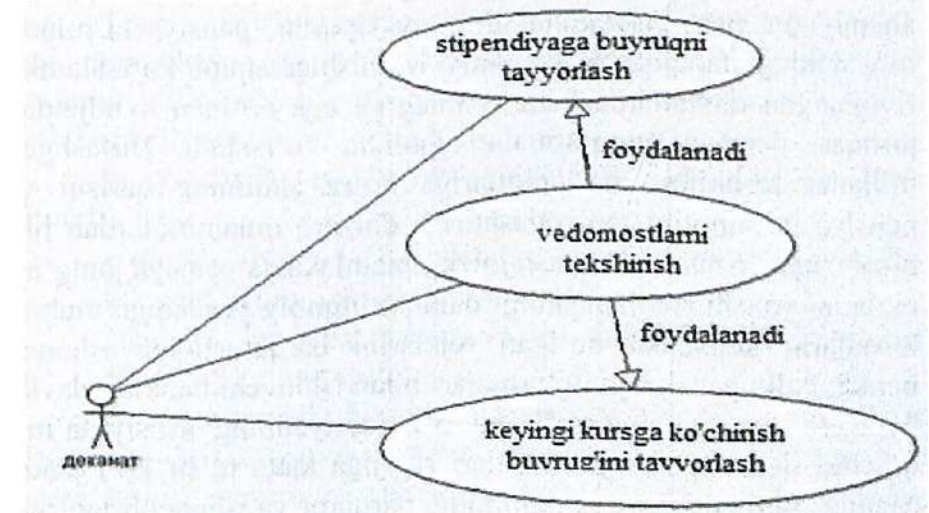
16.8- rasm. ArgoUMLda uyushma turini tanlash asbobi

16.3- jadval. Foydalanish holati diagrammasidagi uyushmalarni tanlash asboblari

Asbob	Tavsif
	Tobe
	Umumlashtirish. Avloddan ajdodga tortiladi
	Kengaytma. Qo'shimcha elementdan asosiyga tortiladi
	Yoqish

Foydalanish holati diagrammasini tuzish uchun dekanat tomonidan bajariladigan ba'zi ishlarni ko'rib chiqamiz. Biz aktyorning diagrammasida asboblarning chizimasidagi aktyorning rasmini, so'ngra tahrirlash chizimini bir marta bosish orqali tasvirlaymiz. Biz to'rtburchaklar ichidagi aktyorning rasmini olamiz. Sichqoncha ko'rsatgichini to'rtburchakdan tashqariga ko'chirmasdan, darhol aktyorga nom beramiz – "dekanat", uni klaviaturada yozamiz. Tahrirlash chizimasida yaratilgan element (bu holda aktyor) yo'naltirish chizimasining daraxtsimon tuzilishida ham paydo bo'ladi.

Keyin biz foydalanish holatlari va yushmalarni qo'shamiz. Ushbu elementlar asboblarning paneli yordamida aktyorlar singari qo'shiladi. Olingan foydalanish holatlari diagrammasi 16.9- rasmda ko'rsatilgan. Biz diagrammalar, diagramma elementlari nomlarini belgilaymiz, shuningdek qismlar chizimasidagi "Имя" oynasida elementlarning nomlarini tahrirlashimiz mumkin. 16.4- rasmda ko'rsatilgan foydalanish holatlari diagrammasi asboblarning chizimasidan foydalanmasdan tahrirlash chizimasiga elementlarni qo'shishimiz mumkin.



16.9- rasm. Foydalanish holatlari diagrammasi

Agar tegishli elementlar (aktyorlar, foydalanish holatlari va uyushmalar) ilgari ma'lum bir diagramma yoki joriy loyihaning boshqa diagrammalari uchun yaratilgan bo'lsa, ular yo'naltirish chizimasida keltirilgan. Ularni yo'naltirish chizimasidan tahrirlash chizimasiga sudrab, joriy diagrammaga nusxalash mumkin. Tahrirlash chizimasidagi diagramma elementini tanlash yo'naltirish chizimasida va qismlar chizimasida ta'kidlanishi bilan birga keladi.

#### 16.4. "Pensiya ishi" axborot tizimining foydalanish holatlari diagrammasi

Dunyodagi mamlakatlarda fuqarolarning pensiya ta'minoti tizimi, pensiya jamg'armalari mamlakatni ijtimoiy-iqtisodiy rivojlantirishdagi majmua va strategik vazifalarni ham o'z zimmasiga oladi. Har bir davlatning ijtimoiy siyosatida aholining ijtimoiy yordamga muhtoj qatlamini ijtimoiy qo'llab-quvvatlash muhim ahamiyatga ega. Davlatning ijtimoiy siyosati, pensiya ta'minoti to'g'risidagi tadqiqotlar va ilmiy izlanishlar shuni ko'rsatadiki, rivojlangan davlatlarda katta mablag'ga ega bo'lgan byudjetdan tashqari pensiya jamg'armalari faoliyat ko'rsatadi. Birlashgan millatlar tashkiloti ma'lumotlariga ko'ra aholining qarishi va pensiya ta'minotini moliyalashtirish dolzarb muammolardan biri hisoblanadi. Shuni ta'kidlash joizki, mamlakatda pensiya jamg'armalarini yaxshi rivojlanganligi nafaqat ijtimoiy yordamga muhtoj insonlarni kelajakka bo'lgan rejalarini barqarorligiga ishonch beradi, balki pensiya jamg'armalari bilan ishlovchi banklar, davlat byudjeti, pensiya mablag'larini o'z faoliyatining asosiy ta'minotchisi deb bilgan fond birjalari rivojiga katta ta'sir ko'rsatadi. Mamlakatimizda pensiya tizimining barqaror va ishonchli faoliyat yuritishi, izchil yuqori o'sish sur'atlari, iqtisodiyotda amalga oshi-

rilayotgan ijtimoiy o'zgarishlar, mamlakatimizni modernizatsiya qilish yo'lidagi dadil qadamlarda ijtimoiy barqarorlik, ijtimoiy ziddiyatlarga yo'l qo'ymaslik masalalari dunyo jamoatchiligi, shuningdek, nufuzli xalqaro moliya tashkilotlari tomonidan e'tirof etildi.

Ob'yektga yo'naltirilgan texnologiyaning asosi modellashtirishdir. Ob'yektga yo'naltirilgan texnologiya asosan artefakt dasturiy ta'minotidan foydalangan holda haqiqiy dunyoga taqlid qilish sifatida ishlab chiqilgan va amalga oshirilgan. Ushbu asos soddaligi tufayli juda kuchli natijalar beradi. Ob'yektga yo'naltirilgan tizimlarni tahlil qilish, ishlab chiqish va amalga oshirishda bir xil tilda ifodalangan fikrlardan foydalanish mumkin.

Modellashtirish birinchi navbatda tizimni yaratishdan oldingi jarayon sifatida qaraladi va loyihalash uchun asos bo'ladi. Bu xususiyat amaliyotchilarning tajribasidan foydalanish orqali tizim juda yuqori darajada loyihalash qobiliyati bilan birgalikda ilgari mumkin bo'lgandan ko'ra murakkabroq tizimlarni ishlab chiqish va muvaffaqiyatli amalga oshirish imkonini beradi.

Yuqorida keltirilgan fikrlar asosida biz amaliyotda ko'radigan diagrammalarimizda ayrim misollardan tashqari "*Pensiya ishi*" axborot tizimining har bir jihatini 7ta diagrammada tahlil qilib, lohilashtirib boramiz.

Biz chizmada (16.10- rasm) uchta aktyorni tanlab olamiz. Bunda:

- birinchi aktyor pensiya olish huquqiga ega bo'lgan fuqaro;
- ikkinchi aktyor pensiya jamg'armasi mas'ul xodimi;
- uchinchi aktyor esa moliya vazirligining byudjetdan tashqari pensiya jamg'armasining mas'ul xodimi.

Bu diagrammada birinchi aktyor o'zi to'g'ridan-to'g'ri o'zi yoki mehnat faoliyatini olib borayotgan tashkilot orqali pensiya jamg'armasiga murojaat qilishi mumkin. Pensiya jamg'armasidagi ikkinchi aktyor fuqarolardan kelgan murojaatni qabul qilib fuqaroga pensiya tayinlash uchun kerak bo'lgan hujjatlarni kerakli



Dispenser (kassa asbobi). Diagrammadagi har bir sinf uch qismga bo'lingan to'rtburchaklar shaklida ko'rinadi. Birinchisida sinf nomi, ikkinchisida uning atributlari mavjud. Oxirgi qismda uning xatti-harakatlarini aks ettiruvchi sinf amallari (sinf tomonidan bajariladigan harakatlar) mavjud. Chiziqlar bog'lovchi sinflardagi o'zaro aloqalarni aks ettiradi. Masalan, Account sinfi AKM ekrani sinfi bilan bog'liq, chunki ular bir-biri bilan bevosita muloqot qiladi va o'zaro ta'sir qiladi. Card Reader sinfi Cash Dispenser sinfiga ulanmagan, chunki ular bir-biri bilan bevosita aloqa qilmaydi.

### Sinf qat'iy turlari

*Qat'iy turlar* – bu omollar bo'yicha sinflarni ajratish mexanizmidir. UMLda sinf bo'yicha uchta asosiy qat'iy turlar belgilangan:

- ❖ *chegara (boundary)*;
- ❖ *entity (mohiyat)*;
- ❖ *control (boshqaruv)*.

*Chegara sinflari (boundary classes)* deganda, tizim va butun atrof-muhitning chegarasida joylashgan sinflar tushuniladi. Bular ekran shakllari, hisobotlar, aabob interfeyslari (masalan, printerlar yoki skanerlar) va boshqa tizimlar bilan interfeyslar. Chegara sinflarini topish uchun foydalanish holatlari diagrammalarini tadqiq qilish kerak. Aktyor va foydalanish holati o'rasidagi har bir o'zaro ta'sir kamida bitta chegara sinfiga mos kelishi kerak. Aynan shu sinf aktyorga tizim bilan o'zaro aloqada bo'lishga imkon beradi.

*Mohiyat sinflari (entity classes)* saqlanadigan hamma berilganlarni o'z ichiga oladi. Ular foydalanuvchi uchun eng katta ahamiyatga ega va shuning uchun ularning nomlari ko'pincha mavzu sohasidagi atamalardan iborat bo'ladi. Odatda, har bir mohiyat sinflari uchun berilganlar bazasining jadvallari yaratiladi.

*Boshqaruv sinflari (control classes)* boshqa sinflarning harakatlarini muvofiqlashtirish uchun javobgardir. Odatda, har bir foydalanish holatida undagi voqealar ketma-ketligini boshqaradigan bitta boshqaruv sinfi mavjud. Boshqaruv sinfi muvofiqlashtirish

uchun javob beradi, lekin o'zi hech qanday funksiyaga ega emas, chunki qolgan sinflar unga ko'p sonli xabarlarini yubormaydi. Buning o'rniga u o'zi ko'plab xabarlarini yuboradi.

### Paketlar mexanizmi

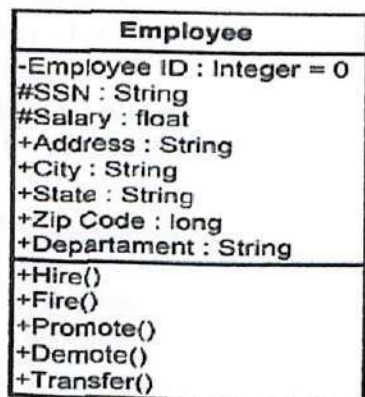
Paketlar ma'lum bir umumiylikka ega bo'lgan sinflarni guruhlash uchun ishlatiladi. Guruhlashning keng tarqalgan yondashuvlari mavjud. *Birinchidan*, biz ularni qat'iy turlar bo'yicha guruhlashimiz mumkin. Bunday holda, bitta paketda chegara sinflari, ikkinchisida mohiyat sinflari va uchichsida boshqaruv sinflari olinadi va h.k. Ushbu yondashuv tayyor tizimni joylashtirish nuqtayi nazaridan foydali bo'lishi mumkin, chunki barcha mijozlar komputerlarida chegara sinflari bitta paketda jamlanib qoladi.

Yana bir yondashuv-sinflarni birlashtirish ularning funksionalligi bo'yicha. Masalan, Security (xavfsizlik) paketi ilovalar xavfsizligi uchun mas'ul bo'lgan barcha sinflarni o'z ichiga oladi. Bunday holda, boshqa paketlarni Employee Maintenance (xodimlar bilan ishlash), Reporting (hisobotlarni tayyorlash) va Error Handling (xatolar ishlovi), deb atash mumkin. Ushbu yondashuvning afzalligi qayta foydalanish imkoniyatining mavjudligida.

### Atributlar

*Atribut* – bu sinf bilan bog'liq bo'lgan berilganlar elementi. Masalan, Company (kompaniya) sinfiga Name (Ism), Address (manzil) va NumberOfEmployees (xodimlar soni) atributlari bo'lishi mumkin. Atributlar sinf ichida joylashganligi sababli, ular boshqa sinflardan yashiringan. Shu munosabat bilan, qaysi sinflar atributlarni o'qish va o'zgartirish huquqiga ega ekanligini ko'rsatish kerak bo'lishi mumkin. Ushbu xususiyatni, biz atributning ko'rish chegarasi (attribute visibility) deb nomlagan va +Public (ochiq), -Private (yopiq), #Protected (himoyalangan) va Package or Implementation (paketli) nomli hamda maxsus belgili ko'rsatkichlarga ega ekanligini aytgan edik.

Ularning har birini 17.2- rasm misolida tahlil qilamiz. *Address* atributiga ega bo'gan *Employee* sinfi va *Company* sinfida *+Public*, *-Private*, *#Protected* va *Package* korsatkichlari berilgan bo'lsin. *+Public* har qanday sinf atribut qiymatini ko'rishi yoki o'zgartirishini taxmin qiladi. Bunday holda, *Company* sinfi *Employee* sinfining *Address* atributining qiymatini o'zgartirishi mumkin.



17.2- rasm. Atributlarning ko'rinish chegarasi

*-Private* orqali tegishli atribut boshqa biron bir sinf tomonidan ko'rinmaydi. *Employee* sinfi *Address* atributining qiymatini bilib oladi va uni o'zgartira oladi, lekin *Company* sinfi uni ko'ra olmaydi yoki tahrirlay olmaydi. Agar zarur bo'lsa, u *Employee* sinfidan ushbu atributning qiymatini ko'rishni yoki o'zgartirishni so'rashi kerak, bu odatda umumiy amallar orqali amalga oshiriladi. *#Protected* atributi faqat sinfning o'zi va uning avlodlari uchun mavjud. Aytaylik, bizda ikki xil turdagi xodimlar bor – soatbay ish haqi va maoshli. Shunday qilib, biz *Employee* sinfining avlodlari bo'gan yana ikkita *Hourlyemp* va *SalariedEmp* sinflariga ega bo'lamiz. Himoyalangan *Address* atributini *Employee*, *HourlyEmp* va *SalariedEmp* sinflaridan ko'rish yoki o'zgartirish mumkin, lekin *Company* sinfidan emas. *Package* ko'satkichi, atributi umumiy deb taxmin qiladi, lekin faqat o'zining paketi doirasida.

Aytaylik, *Address* atributi *package* ko'rinishga ega bo'sin. Bu holda, uni *Company* sinfidan o'zgartirish mumkin, agarda u ham shu sinfda bo'lsa.

Umuman olganda, atributlarni yopiq yoki himoyalangan qilish tavsiya etiladi. Bu atributning o'zi va kodini yaxshiroq nazorat qilish imkonini beradi.

Yopiqlik yoki xavfsizlik yordamida atributning qiymati, tizimning barcha sinflari tomonidan o'zgartiriladigan vaziyatdan qochish mumkin. Buning o'rniga, atributni o'zgartirish mantig'i ushbu atributning o'zi bilan bir xil sinfda bo'lgani yaxshi. Belgilangan ko'rinish chegaralari yaratiladigan kodga ta'sir qiladi.

### Amallar

*Amallar* sinf bilan bog'liq xatti-harakatlarni amalga oshiradi. Amal uchta qismni o'z ichiga oladi – nom, parametrlar va qaytish qiymatining turi.

*Parametrlar* – bu "kirish"amali tomonidan olingan argumentlardir. Qaytish qiymatining turi amal natijasidagiharakatga tegishli Sinf diagrammasida amallar nomlari va amallar nomlari bilan ularning parametrlari va qaytish qiymatining turi bilan birga ko'rsatilishi mumkin. Diagrammaning yuklamasini kamaytirish uchun ularning ba'zilarida faqat amal nomlarini ko'rsatish foydali bo'ladi, boshqalarida esa to'liq holda.

UMLda amallar 4 xil ega va quyidagi formatga ega:

**Amal nomi (argument1: turi, argument2: turi,...): qaytish qiymatining turi**

### Amalga oshirish amallari

*Amalga oshirish amallari (implementor operations)* ba'zi tijorat funksiyalarini amalga oshiradi. Bunday amallarni o'zaro harakat diagrammalarini tadqiq qilish orqali topish mumkin. Ushbu turdagi diagrammalar tijorat funksiyalariga qaratilgan va diagrammaning har bir xabari, ehtimol, amalga oshirish amali bilan bog'liq bo'lishi mumkin.

### Kirish amallari

Atributlar odatda yopiq yoki himoyalangan bo'ladi. Biroq, boshqa sinflar ba'zan ularning qiymatlarini ko'rishi yoki o'zgartirishi kerak. Buning uchun *kirish amallari (access operations)* mavjud. Masalan, Employee (xodimlar) sinfining Salary (ish haqi) atributini tahlil qilaylik. Biz boshqa barcha sinflar ushbu atribut qiymatini o'zgartirishiga qarshi bo'laylik. Buning uchun, xodimlar sinfiga biz ikkita kirish amalini qo'shamiz – GetSalary va SetSalary. Bulardan birinchisiga umumiy bo'lgani uchun boshqa sinflar ham murojaat qilishlari mumkin. U shunchaki ish haqi atributining qiymatini oladi va uni chaqirgan sinfga qaytaradi. Setsalary amali ham umumiy bo'lib, uni chaqirgan sinfga ish haqi atributining qiymatining yangi qiymatini o'rnatishga yordam beradi. Ushbu amal ish haqini o'zgartirish uchun, bajarilishi kerak bo'lgan har qanday tekshirish qoidalari va shartlarini o'z ichiga olishi mumkin.

### Yordamchi amallar

*Yordamchi amallar (helper operations)* – bu uning javobgarligi bajarishi kerak bo'lgan, ammo boshqa sinflar bilmasligi kerak bo'lgan sinf amallari. Bu yopiq va himoyalangan sinf amallari. Bunday amallarni aniqlash uchun quyidagi harakatlarni bajaring:

1. Ketma-ketlik diagrammalarini va hamkorlik diagrammalarini o'rganing. Ushbu diagrammalardagi xabarlarining aksariyati, bu amalga oshirish amallari. Aks xabarlar yordamchi amallar bo'ladi.

2. Boshqaruv amallarini ko'rib chiqing. Balki konstruktorlar va destruktorga qo'shish kerakdir.

3. Kirish amallarini ko'rib chiqing. Boshqa sinflar ishlashi kerak bo'lgan sinfning har bir atributi uchun Get va Set amallarini yaratishingiz kerak.

### Aloqalar

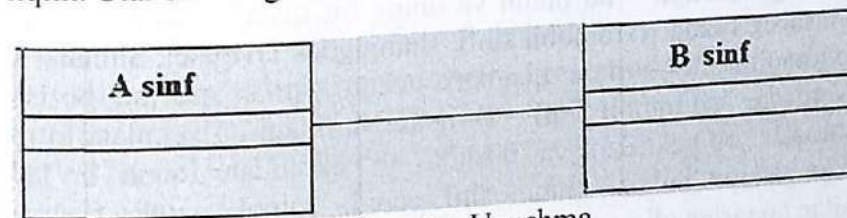
*Aloqalar* – bu sinflar o'rasidagi semantik munosabatlar. Bu

sinfga boshqa sinfning atributlari, amallari va aloqalari to'g'risida ma'lumot olish imkoniyatini beradi. Boshqacha qilib aytganda, bir sinf boshqasiga ketma-ketlik yoki hamkorlik diagrammasida xabar yuborishi uchun ular o'rasida aloqa bo'lishi kerak.

Sinflar o'rasida o'rnatilishi mumkin bo'lgan to'rt turdagi aloqalar mavjud – *uyushmalar, tobeliklar, agregatsiyalar va umumlashmatirishlar.*

### Uyushmalar

*Uyushma (association)* – bu sinflar o'rasidagi semantik bog'liqlik. Ular sinf diagrammasida oddiy chiziq shaklida chiziladi.



17.3- rasm. Uyushma

Uyushmalar 17.3- rasmda bo'lgani kabi ikki tomonlama yoki bir tomonlama bo'lishi mumkin. UMLda ikki tomonlama uyushmalar milsiz yoki uning ikkala tomonidagi millar bilan oddiy chiziq shaklida chiziladi.

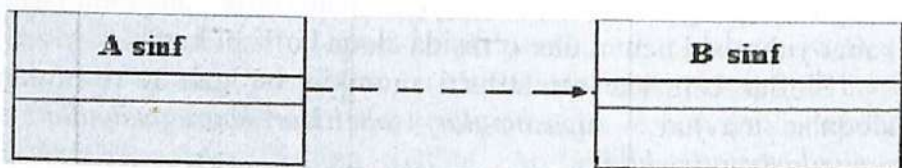
Bir yo'nalishli uyushmada uning yo'nalishini ko'rsatadigan faqat bitta mil tasvirlanadi.

### Tobelik

*Tobelik aloqalari (dependency)* ham sinflar orasidagi aloqani aks ettiradi, lekin ular har doim bir tomonlama bo'lib, bir sinf boshqasida qilingan ta'riflarga tobeligini ko'rsatadi. Tobeliklar nuqta chiziq bilan chizilgan mil shaklida tasvirlanadi (17.4- rasm).

Tobe sinflar uchun kod yaratishda ularga yangi atributlar qo'shilmaydi. Biroq, aloqani qo'llab-quvvatlash uchun zarur bo'lgan tilga xos operatorlar yaratiladi. Masalan, C++ tilida kod ke-

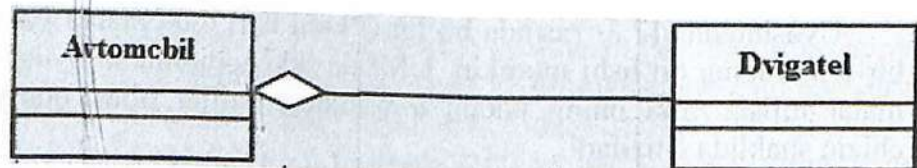
rakli # include operatorlarini o'z ichiga oladi.



17.4- rasm. Tobelik

### Agregatsiyalar

*Agregatsiyalar (aggregations)* uyushmaning zichroq shaklidir. Agregatsiya – bu butun va uning bir qismi o'rasidagi aloqa. Masalan, bizda Avtomobil sinfi, shuningdek Dvigatel, Shinalar va avtomobilning boshqa qismlari uchun sinflar mavjud bo'lsin. Natijada, Avtomobil sinfi – Dvigatel sinfidagi ob'yektdan, to'rtta shinalar ob'yektdan va boshqa ob'yektlardan iborat bo'ladi. Agregatsiya belgisi, butun sinf yonida rombli uzluksiz chiziq bilan tasvirlanadi.



17.5- rasm. Agregatsiya

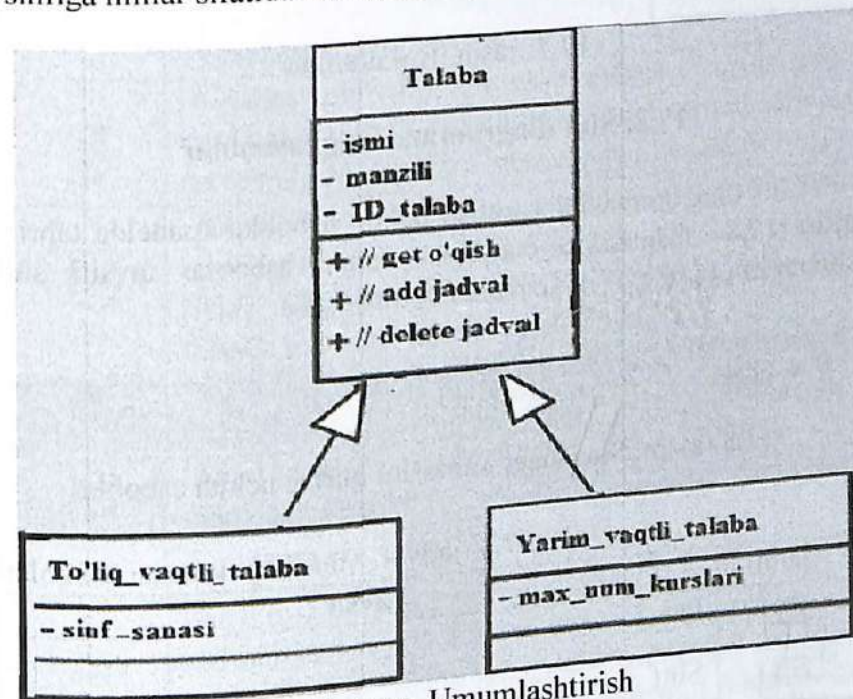
Oddiy agregatsiyadan tashqari, UML *kompozitsiya* deb ataladigan yanada kuchli agregatsiya turini taqdim etadi. Kompozitsiyaga ko'ra, ob'yekt-qism faqat bitta butunga tegishli bo'lishi mumkin va bundan tashqari, qoida tariqasida, qismlarning Yashash sikli butun siklga to'g'ri keladi, ular u bilan birga yashaydi va yo'q bo'ladi. Butunni har qanday olib tashlash uning qismlariga tarqaladi.

Bunday pog'onali olib tashlash ko'pincha agregatsiya ta'rifi-ning bir qismi sifatida ko'rib chiqiladi, ammo bu har doim rolning

karraligi 1..1 bo'lganda nazarda tutiladi. Masalan, agar mijozni olib tashlash kerak bo'lsa, unda bu o'chirish Buyurtmalarga (va o'z navbatida buyurtma qatorlariga) ham tegishli bo'lishi kerak.

### Umumlashtirishlar

*Umumlashtirishlar (generallashtirish)* yordamida ikki sinf o'rtasidagi meros aloqalar ko'rsatiladi. Ob'yektga yo'naltirilgan tillarning aksariyati meros tushunchasini bevosita qo'llab-quvvatlaydi. Bu bitta sinfga boshqasining barcha atributlari, amallari va aloqalarini meros qilib olishga imkon beradi. UML tilida meros aloqalari *umumlashtirishlar*, deb ataladi va avlod sinfidan ajdodlar sinfiga millar sifatida tasvirlanadi (17.6- rasm).

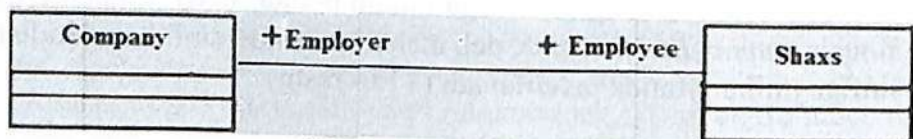


17.6- rasm. Umumlashtirish

### Rollar

Rol nomlari uyushma yoki agregatsiya aloqalarida ushbu

bog'lanishlar nima uchun kerakligini tavsiflash uchun nomlar o'rniga ishlatiladi. Shaxs va Company sinflari misoliga qaytsak, Shaxs sinfi Company sinfining xodimi rolini o'ynaydi, deyishimiz mumkin. Rolli ismlar odatda millar yoki ularga asoslangan ot turkumidagi atamalar bo'lib, ular tegishli rol o'ynaydigan sinf yonidagi diagrammada ko'rsatiladi. Qoida bo'yicha, ular rol nomi yoki aloqa nomidan foydalanadilar, lekin ikkalasidan birdaniga emas. Aloqa nomlari singari, rol nomlari majburiy emas, ular faqat aloqa maqsadi aniq bo'lmasa beriladi. Rollarga misol keltirilgan quyida:



17.7- rasm. Rol nomlari

## 17.2. Sinf diagrammasidagi asboblari

Sinf diagrammasini yaratish uchun asboblari (panelda tahrirlash) 17.8- rasmda ko'rsatilgan. uchun asboblari tavsifi sinf chizmasini qurish 17.1- jadvalda keltirilgan.



17.8- rasm. Sinf diagrammasini qurish uchun asboblari

17.1- jadval. Sinf diagrammasi asboblari

Asbob	Tavsif
	Paket
	Sinf
	Uyushma. Ikki yo'nalishli bo'lishi mumkin.

	Агрегация (Birlashtirish). Ikki yo'nalishli bo'lishi mumkin.
	Композиция. Ikki yo'nalishli bo'lishi mumkin.
	Uyushmaning tugashi. Mavjud boshqa uyushmaga boshqa uchini qo'shish uchun (chap tugma yordamida yushma markazidan sinfga yoki aksincha). Nar uyushmalarni yaratish uchun foydalaniladi.
	Umumlashtirish. Avloddan ajdodga tortiladi.
	Diagrammaga interfeysni qo'shish. Qulaylik uchun, sichqoncha ko'rsatkichi tanlangan interfeysning ustida bo'ganda u bir qalamni ko'rsatadi, keyin elementni sudrab olish mumkin.
	Amalga oshirish. Sinf va interfeys o'rasida sichqonchani chap tugmasini siljitish orqali amalga oshiriladi. Mil sinfdan interfeysga o'tkaziladi.
	Tobelik. Ikkita element orasida tobelik, sichqonchani chap tugmasini siljitish orqali amalga oshiriladi. Mil tobe elementdan o'tkaziladi. 2 turdagi tobelik mavjud – ruxsat va foydalanish.
	Xususiyat. Element faqat tanlanganda faol bo'ladi element sinfdir.
	Amal. Odatda, u yangi amal, deb nomlanadi. Uni tahrirlash mumkin.
	Uyushma sinfi. Bog'langan sinf.
	Sana turi. Qulaylik uchun, qachon sichqoncha ko'rsatkichi tanlangan element ustida joylashgan bo'lsa, tutqichlar paydo bo'ladi. Bosish mumkin bo'lgan yuqori va pastki yoki kerakli elementni hosil qilish uchun tortiladi. 2ta element turi mavjud - Enum va qat'iy tur.

Datatype

Dekanat faoliyati bilan bog'liq misolni ko'rib chiqamiz. Fan sohasini tahlil qilish bizni quyidagi sinflarni tanlashga olib keladi:

1. Fakultet.
2. Kafedra.
3. Kurs.
4. Semestr.
5. Kurator.
6. Talaba.
7. Kotiba.

### Atributlarni aniqlash

Keyingi qadamda sinf har bir atributlari (xususiyatlari)ni aniqlaymiz. Masalan, "Talaba" sinfi quyidagi xususiyatlarga ega bo'lishi mumkin – to'liq ism, tug'ilgan sanasi va pasport yoki ID raqami. "Kurs" sinfi quyidagi xususiyatlarga ega bo'lishi mumkin - Kod, dars soatlarining hajmi. Har bir atribut o'z berilganlar turiga ega bo'ladi, bu esa undagi berilganlar qandayligini belgilaydi, satrli (belgi) yoki raqamli saqlanadi.

### Funksiyalar (amallar) ta'rifi


Funksiyalar (amallar) – bu mohiyat bajarishi mumkin bo'lgan harakatlardan iborat, masalan talaba qo'shimcha kursga yoki tanlov fan bo'yicha ro'yxatdan mumkin. Har bir funksiya bir nechta kirish parametrlariga ega bo'lishi va faqat bittagina ob'yektni qaytarishi mumkin.

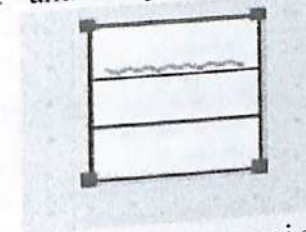
### Sinflar (uyushmalar) o'rasidagi munosabatlarni aniqlash

Har qanday soha bo'yicha mohiyatlar o'rasidagi mumkin bo'lgan munosabatlarni tahlil qilish zarur. "Dekanat" sohasidagi mohiyatlar o'rasida quyidagi munosabatlar mavjud – fakultet tarkibida tarkibiga bir nechta kafedralar kiradi va har bir kafedraga bir nechta kurs bog'lanadi. Ushbu munosabatlarni UMLda uyushmalar yordamida ifodalash mumkin.

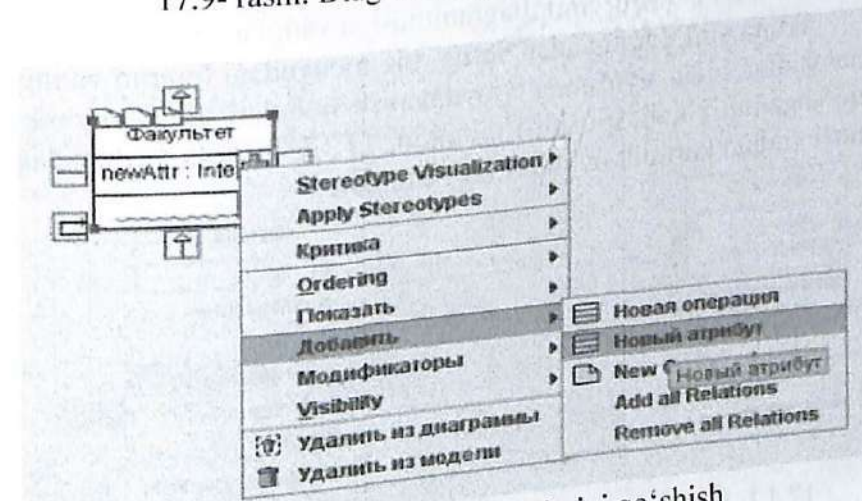
Uyushma ikkita sinf o'rasidagi aloqani aniqlaydi va har bir uyushmaning oxiri karralik bilan belgilanadi. Masalan, *fakultet* – *kafedra* munosabatining turi 1: M bo'lsa, diagrammada bu 1..\* kabi ko'rsatiladi.

## 17.3. Sinf diagrammasini qurish

Yangi sinf qo'shish uchun biz  belgini ustiga 17.8- rasmda ko'rsatilgan asboblardan bosishimiz kerak. Bu ko'rsatkichdan keyin sichqoncha ko'rsatkichi bo'lishi mumkin bo'lgan xoch shaklni ko'rsatadi, uni tahrirlash chizimasi atrofida harakatlantiramiz. Kursorni siljitgan holda, yangi sinfni joylashtirmoqchi bo'lgan joyga, chap tugma bosiladi. Shundan so'ng, 17.9- rasmdagi sinfni ifodalovchi tahrirlash chizimasi ko'rsatiladi. Qizil to'lqinli chiziq sinf nomini kiritish joyini ko'rsatadi. Sinf uch qismga bo'linadi, yuqori qismida sinf nomi, o'rta qismida sinf atributlarini, pastki qismda - amallari yoki sinf funksiyalarini yoziladi.



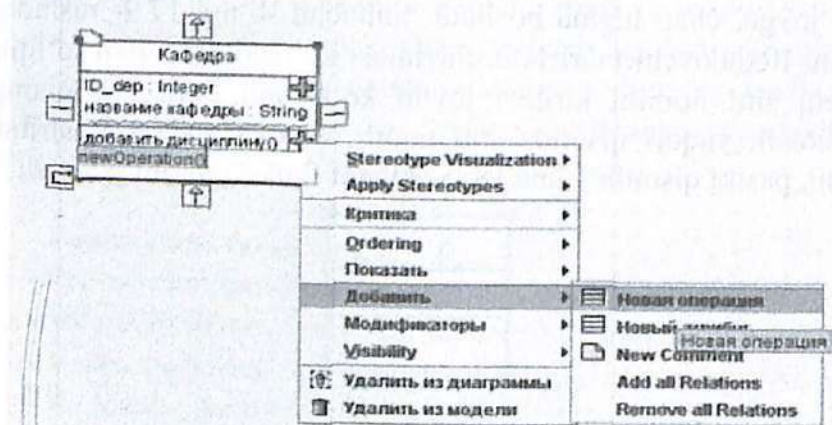
17.9- rasm. Diagrammaga yangi sinf qo'shish



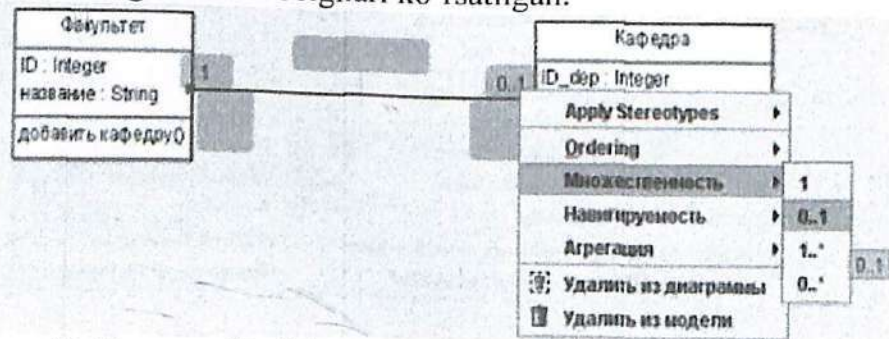
17.10- rasm. Diagrammaga sinf atributlarini qo'shish

Qizil to'liqlik chiziq ustida ko'satkich bosilsa, diagrammaga sinfnining nomini yozish mumkin. Atributlarning nomlarini 17.10-rasmda ko'rsatilgan suzuvchi menyuda qo'shish mumkin.

Atributlarning turini qismlar chizimasidagi ro'yxatdan tanlash mumkin. Yangi amalni qo'shish 17.11- rasmda ko'rsatilgan. Amalning nomi va turning ko'rsatkichlari qavslar ichida kelishi mumkin.



17.11- rasm. Sinf diagrammasiga yangi amal qo'shish  
Ikkita sinf yaratgandan so'ng, biz menyudagi buyruq va mos harakatlar bilan uyushmani foydalanish holatlari diagrammasiga qo'shganimiz kabi qo'shish mumkin. 17.12- rasmda yushmadagi turli xildagi karralilik belgilari ko'rsatilgan.



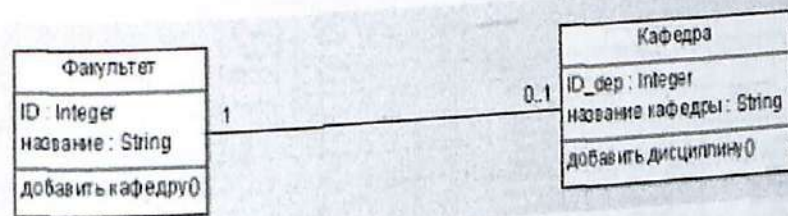
17.12- rasm. Sinf diagrammasida uyushmaning karraligini

ko'rsatish

Ko'plikning mumkin bo'lgan turlari ochiladigan menyuda «Множественность» buyrug'da ko'rsatilgan.

Uyushmaning karraligini ko'rsatishning boshqa usuli, kursorni uyushma chegarasidagi rangli to'rtburchakka olib borib, sich-qonchanning chap tugmasi bilan ikki marta bosib va klaviaturadan kiritish.

Olingan diagramma 17.13- rasmda o'xshash bo'lishi kerak.



17.13- rasm. Sinf chizmasi

#### 17.4. "Pensiya ishi" ATning sinf diagrammasi

17.14-rasmda "Pensiya ishi" axborot tizimining sinf diagrammasi ko'rsatilgan. Sinf diagrammasi orqali berilganlar bazasidagi jadvallar, ulardagi atributlar va ular orasidagi aloqalar yaratilgan.

Bunda biz murojaat qilgan fuqaro faqatgina pasport orqali murojaatining tuzilishini keltirdik. Bunda pensiya jamg'armasi-ning mas'ul xodimi fuqaroning pasport ma'lumotlari orqali kerakli tashkilotlarga (soliq, arxiv va h.k) murojaat qilib fuqaro haqidagi berilganlarni o'rganib chiqib to'playdi va moliya vazirligining byudjetdan tashqari pensiya jamg'armasiga yuboradi. Moliya vazirligining byudjetdan tashqari pensiya jamg'armasi mas'ul xodimi kelib tushgan hujjatlarga asosan murojaat qilgan



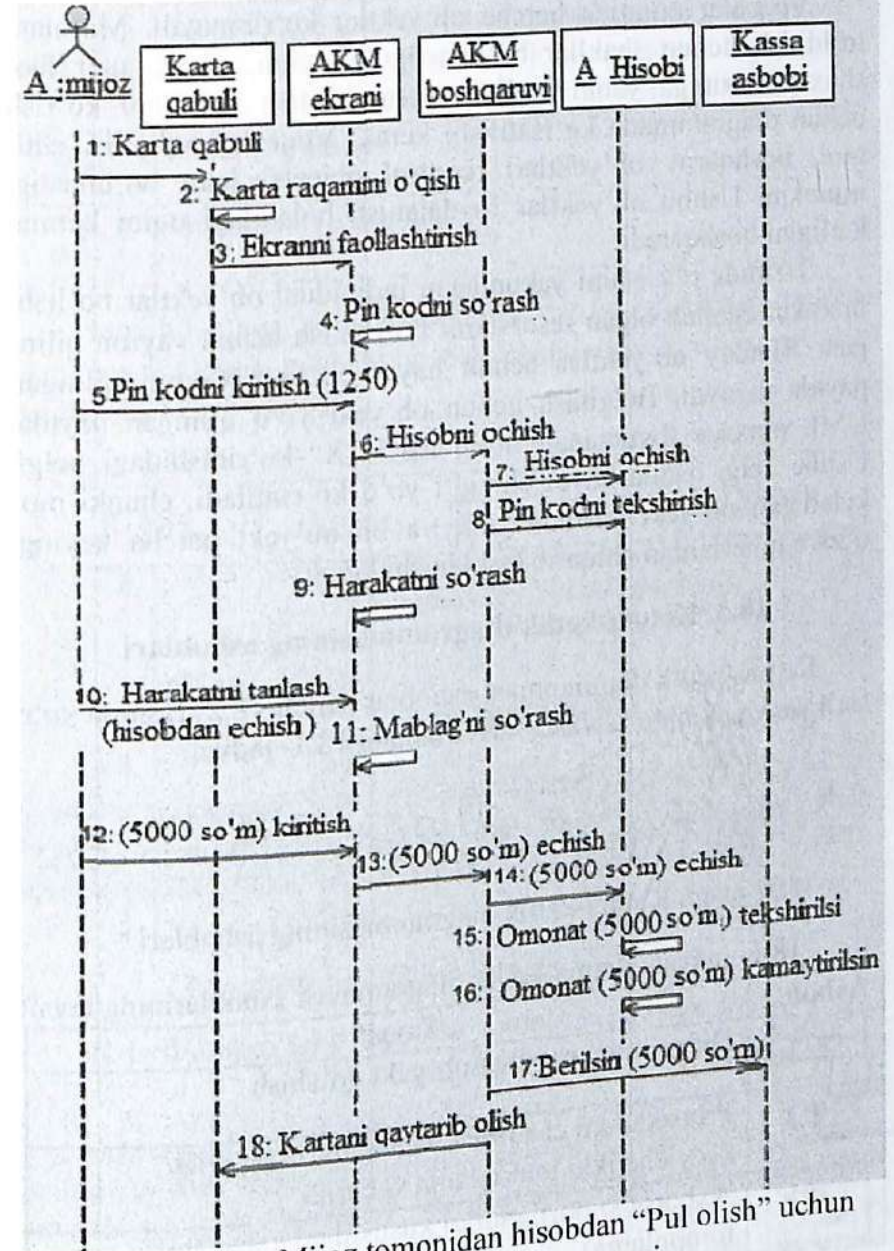
- ❖ hisob raqamida yetarli bo'lmagan holda pulni olib qo'yishga urinish;
- ❖ pulni noto'g'ri aniqlash raqami bilan olib qo'yishga urinish
- ❖ va boshqalar.

Hisobdan pul olishning me'yoriy ssenariysi (noto'g'ri aniqlash raqami yoki hisobdagi pul yetishmasligi kabi muammolar bo'lmagan holat) 18.1- rasmda keltirilgan.

Ushbu ketma-ketlik diagrammasi "pul olish" holatidagi voqealar oqimini ko'rsatadi. Barcha (aktyor)lar diagrammaning yuqori qismida ko'rsatilgan, bu misolda harakatlanuvchi shaxs sifatida mijoz akslantirilgan. "Pul olish" dan foydalanish holatini bajarish uchun tizim tomonidan talab qilinadigan ob'yektlar ham ketma-ketlik diagrammasining yuqori qismida keltirilgan. Millar talab qilinuvchi funksiyalarni bajarish uchun aktyor va ob'yekt o'rasida yoki ob'yektlar o'rasida uzatiladigan xabarlariga mos keladi.

Ketma-ketlik diagrammasida ob'yekt to'rtburchak shaklida tasvirlangan, undan pastga nuqtali vertikal chiziq chizilgan. Ushbu chiziq *ob'yektning yashash chizig'i (lifeline)* deb nomlanadi. Bu o'zaro harakat jarayonida ob'yekt hayot siklining bir qismidir.

Har bir xabar ikkita ob'yektning hayot chiziqlari orasidagi mil shaklida taqdim etiladi. Xabarlar sahifada yuqoridan pastgacha ko'rsatilgan tartibda paydo bo'ladi. Har bir xabar kamida xabar nomi bilan belgilanadi, agar xohlasak, argumentlar va ba'zi boshqaruv ma'lumotlarini ham qo'shishimiz mumkin va bundan tashqari, o'zi-o'ziga xabarni jo'natishi mumkin, bunda xabar mili o'sha hayot chizig'ini ko'rsatadi. Ba'zi ob'yektlarni dastlabki aniqlashning yaxshi usuli bu voqealar oqimidagi ot turkumidagi nomlarni o'rganishdir. Shuningdek, biz aniq bir ssenariyni tavsiflovchi hujjatlarni o'qishimiz ham mumkin. Ssenariy voqealar oqimining aniq nusxasini anglatadi. "Pul olish"dagi foydalanish holati diagrammasi uchun voqealar oqimi AKM yordamida hisobdan ma'lum miqdordagi pulni olmoqchi bo'lgan shaxs haqida gapiradi.



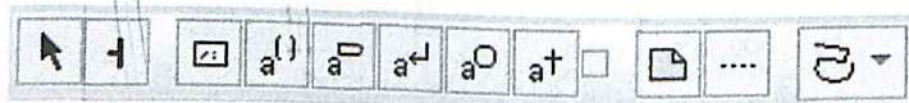
18.1- rasm. Mijoz tomonidan hisobdan "Pul olish" uchun ketma-ketlik diagrammasi

Voqealar oqimida barcha ob'yektlar ko'rinmaydi. Masalan, to'ldirish uchun shakllar bo'lmasligi mumkin, ammo ular faol shaxsga tizimga yangi ma'lumotlarni kiritish yoki uni ko'rish uchun diagrammada ko'rsatilishi kerak. Voqealar oqimida, ehtimol, boshqaruv ob'yektlari (control objects) ham bo'lmasligi mumkin. Ushbu ob'yektlar foydalanish holatidagi oqim ketma-ketligini boshqaradi.

Tizimda o'z ishini yakunlagan individual ob'yektlar bo'lishi mumkin egallab olgan resurslarni bo'shatish uchun vayron qilingan. Bunday ob'yektlar uchun hayot chizig'i vayron qilingan paytda tugaydi. Belgilash uchun ob'yekt yo'q qilingan paytda, UML maxsus foydalanadi lotin harfi "X" ko'rinishidagi belgi. Ushbu belgi ostida hayot chizig'i yo'q ko'rsatiladi, chunki mos keladigan ob'yekt tizimda yo'q va bu ob'yekt barcha keyingi o'zaro ta'sirlardan chiqarib tashlanishi kerak.

### 18.3. Ketma-ketlik diagrammasining asboblari

Ketma-ketlik diagrammasi asboblari paneli 18.2- rasmda ko'rsatilgan va alohida asboblari tasvirlangan 18.1- jadval.



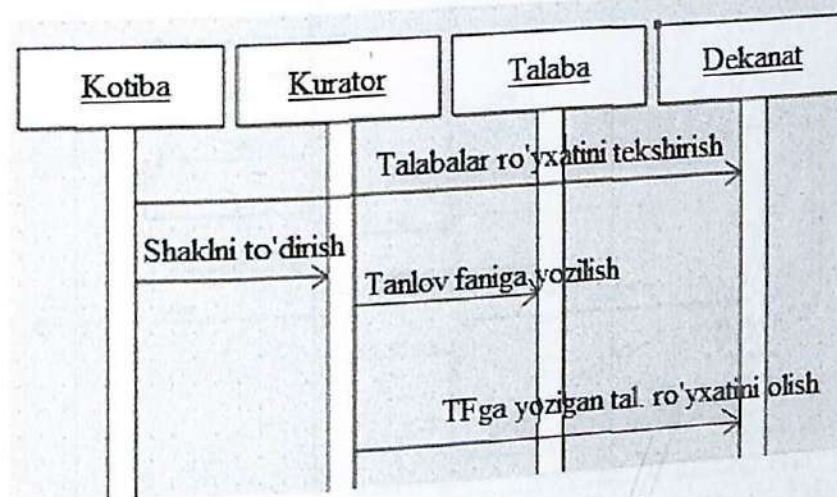
18.2- rasm. Ketma-ketlik diagrammasining asboblari

18.1- jadval. Ketma-ketlik diagrammasi asboblari tavsifi

Asbob	Tavsif
	Diagrammaga yangi ob'yekt qo'shish
	Protsedurani chaqirish va amallarni bajarish bo'yicha ikki tomonlama xabarlar
	Axborot yuborilganligi xususidagi xabar (bir tomonlama)

	Chaqirilgan protseduraning qaytganligi xususidagi xabar. Masalan, ob'yekt mijozga najalarni bermagan holda, hisoblashlar tugaganligi xususidagi xabar.
	Harakat yaratish
	Harakatni yo'q qilish

Soha bilan bog'liqlik sifatida dekanat ish tizimidagi, "tanlangan fan bo'yicha talabani yozish" ketma-ketlik diagrammasini yaratamiz (18.3- rasm).

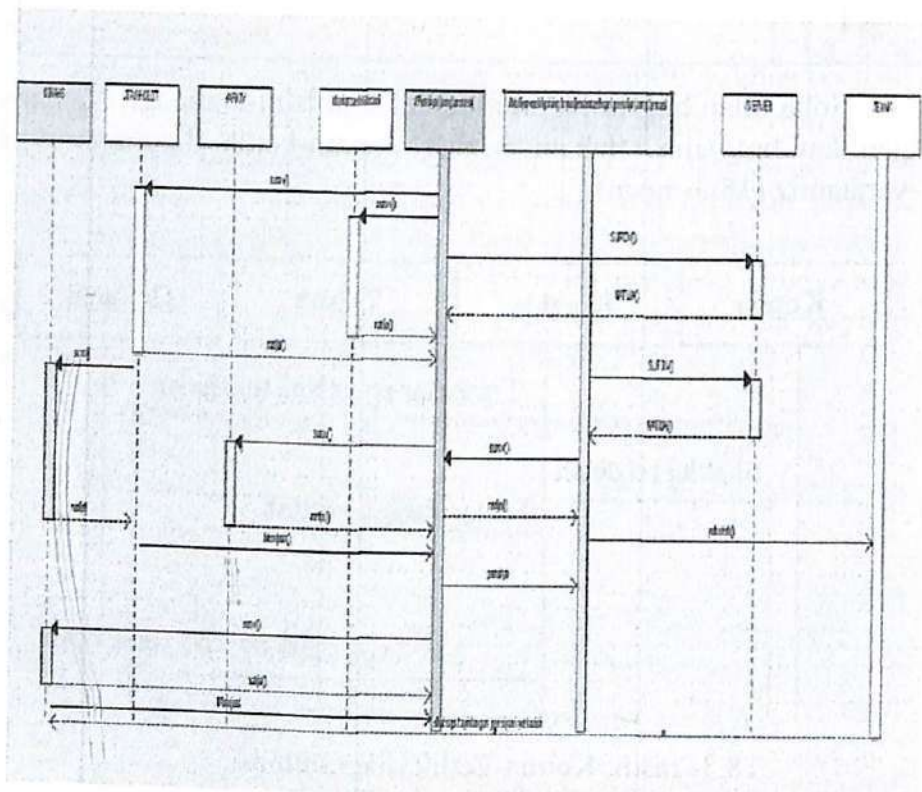


18.3- rasm. Ketma-ketlik diagrammasi

### 18.4. "Pensiya ishi" axborot tizimining ketma-ketliklar diagrammasi

Ketma-ketlik diagrammasida faqat ma'lum bir xronologik ketma-ketlikda xabar almashish (chaqiruv usullari) bilan bevosita aloqada bo'lgan ob'yektlar tasvirlangan. Ular aniq, cheklangan vaqtli vaziyatlarni modellashtirish uchun ishlatiladi. Ketma-ketlik diagrammalari ob'ektlarga yuborilgan xabarlar tartibi va

vaqtini alohida ta'kidlaydi. Ketma-ketlik diagrammalarining asosiy nuqtasi vaqt o'tishi bilan ob'yektlar o'zaro ta'sirining dinamikasidir. AgroUMLda ketma-ketliklar diagrammalarida ob'yektlar sifatida ikkita aktyor va sinf diagrammaridan foydalanish mumkin.



18.4- rasm. Ketma-ketlik diagrammasi

Yuqoridagi 18.4- rasmda ketma-ketlik diagrammasini ko'rishimiz mumkin. Bunda fuqaro tashkilotga yoki pensiya jamg'armasiga murojaatiga qandaydir (ya'ni hujjatlar to'liq kamchiliksiz bo'lsa, unga pensiya tayinlanganligi haqidagi yoki mavjud hujjatlardagi kamchiliklarga ko'ra pensiya tayinlanmaganligi, kerakli hujjatlar haqidagi) natija qaytadi. Bunda pensiya jam-

g'armasining mas'ul xodimi fuqaroning pasport ma'lumotlari orqali kerakli tashkilotlarga so'rov jo'natadi va natija oladi. Agarda hujjatlarda kamchilik bo'lmasa, pensiya jamg'armasining mas'ul xodimi moliya vazirligining byudjetdan tashqari pensiya jamg'armasiga so'rov yuboradi va undan ham natija oladi. Moliya vazirligining byudjetdan tashqari pensiya jamg'armasi mas'ul xodimi kelib tushgan so'rovni o'rganib chiqib, serverga so'rov yuboradi va undan ham natija qaytadi.

## XIX- BOB. HAMKORLIK (ALOQA) DIAGRAMMASI

### 19.1. Nazariy asoslar

Ketma-ketlik va hamkorlik diagrammalari (collaborations) modellashtirilgan mavzu sohasi elementlarining o'zaro harakatini ko'rsatish. Ketma-ketlik diagrammalari o'zaro harakatlardagi vaqtinchalik jihatlarni ifodalaydi va harakatlar xronologiyasining ketma-ketligini, hamkorlik diagrammalari o'zaro harakatining tuzilmaviy jihatlarni ifodalaydi.

Hamkorlik diagrammasida faqat ob'yektlar o'rasidagi munosabatlarning o'zaro harakatida muayyan rol o'ynaydiganlarini ko'rsatadi. Tizim tuzilmasining ob'yektlarning o'zaro harakatlar to'plami sifatida ifodalanishi, tizimning hamkorlik diagrammasidagi statik tavsifiga mos keladi.

**Sinflowchi (Classifier)** – bu tuzilmaviy va xulqiy xususiyatlarini tavsiflovchi mexanizmdir. Sinflowchilar bu – o'z nusxasiga ega bo'lgan model elementlaridir. Sinflowchilarning tuzilmaviy xususiyatlari atributlar bilan, xatti-harakatlar esa amallar shaklida ifodalanadi. Bitta sinflowchining barcha holatlari bir qator umumiy xususiyatlarga ega. Sinflowchilar turli xilda bo'sa ham, ArgoUMLda sinflowchilar bir xil belgi – to'rtburchak yordamida ko'rsatiladi.

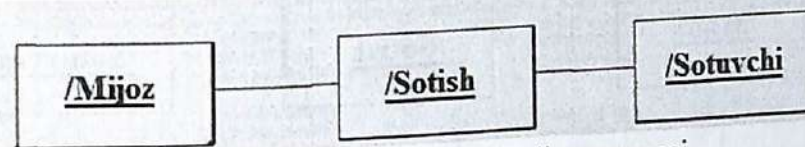
To'rtburchakdagi matn quyidagicha formatda bo'lishi kerak:

**<sinflowchi rolining nomi> : <sinflowchi nomi>**

Hamkorlik ikkita darajada ifodalanishi mumkin:

- tasnifchi darajasida (sinflowchi va uyushmalarning rolini ko'rilayotgan o'zaro harakatda ko'rsatadi);
- misollar darajasida (hamkorlikdagi alohida rollarni tashkil qiluvchi nusxalar va munosabatlarni ko'rsatadi).

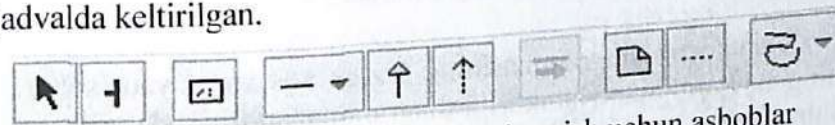
Tasnifchi darajasidagi hamkorlik diagrammasi rollar, bu ishtirok etadigan elementlarning o'zaro harakatini bildiradi. Hamkorlik elementlari bu darajadagi sinflar va uyushmalarni bildiradi, sinflowchilarning alohida rollari va ishtirokchilar o'rasidagi uyushmalarning hamkorlikdagi a'zolari ifodalaydi. Hamkorlikning namunasi 19.1- rasmda ko'rsatilgan



19.1- rasm. Hamkorlik diagrammasi

### 19.2. Hamkorlik diagrammasini qurish

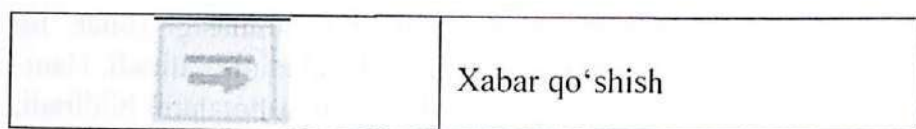
Hamkorlik diagrammasini qurish uchun asboblar chizimasi 19.2- rasmda ko'rsatilgan va alohida asboblar tavsifi 19.1- jadvalda keltirilgan.



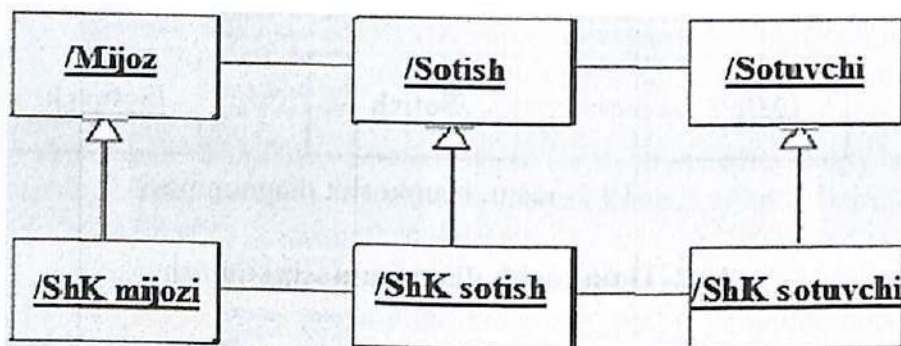
19.2- rasm. Hamkorlik chizimasini qurish uchun asboblar

6.1-jadval - hamkorlik diagrammasi vositalarining tavsifi

Asboblar	Tavsifi
	Sinflowchining roli
	Uyushmaning roli
	Umumlashtirish



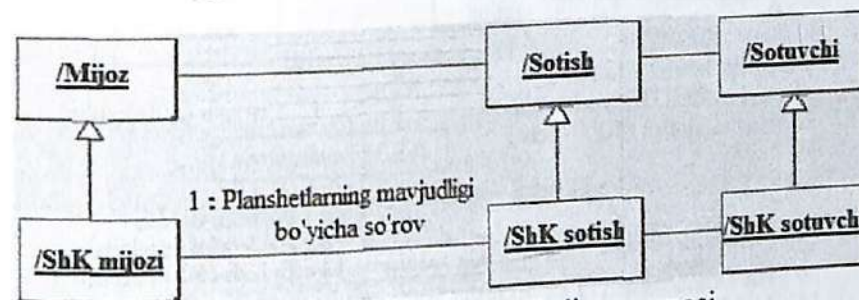
19.3- rasmda. Tasnif darajasida alohida hamkorlar o'rasidagi umumlashtirish munosabatlari ko'rsatilgan.



19.3- rasm. Umumlashtirish munosabatlari bilan hamkorlik diagrammasi

Hamkorlik diagrammasining o'ziga xos xususiyati, xabarlarining mavjudligida. Hamkorlik diagrammasini tuzishda millar qo'shimcha semantik xususiyatlarga ega. Hamkorlik diagrammasidagi xabar ikkita ob'yekt o'rasidagi aloqani aniqlaydi, ulardan biri boshqasiga ba'zi ma'lumotlarni uzatadi. Bunday holda, birinchi ob'yekt xabarni olgandan so'ng, ikkinchi ob'yektning ba'zi harakatlarni bajarishini kutadi. Shunday qilib, bu amallarni boshlash, signallarni yuborish, alohida ob'ektlarni yaratish va yo'q qilish uchun sabab yoki rag'batlantiruvchi xabar. Aloqa ob'ektlar orasidagi xabarlarni manba ob'yektidan qabul qiluvchi ob'yektga yo'naltirish uchun kanalni ta'minlaydi. UMLdagi xabarlar, shuningdek, xabar yuboruvchi va qabul qiluvchi ob'ektlarning rollarini aniqlaydi. Hamkorlik jadvalidagi xabarlar tegishli havola yoki uyushma roli bilan yonma-yon (yuqorida yoki pastda) belgilangan millar bilan tasvirlanadi. Milning yo'nalishi xabarni qabul qiluvchiga ishora qiladi.

ArgoUML muhitida hamkorlik jadvalidagi xabarlarga avtomatik ravishda raqamlar beriladi. 19.4-rasmda xabarga ega bo'lgan hamkorlik diagrammasi ko'rsatilgan.

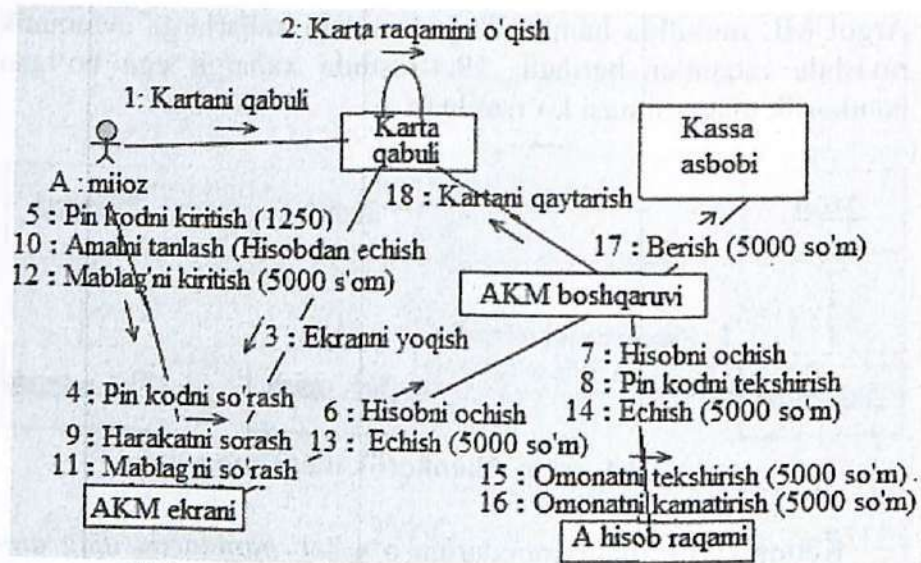


19.4- rasm. Hamkorlik diagrammasi

Ketma-ketlik diagrammalariga o'xshab hamkorlik diagrammalari (collaborations) voqealar oqimini ma'lum bir foydalanish holati ssenariysi orqali aks ettiradi. Ketma-ketlik diagrammalari vaqt bo'yicha tartiblangan bo'lsa, hamkorlik diagrammalari esa ob'ektlar orasidagi aloqalarga ko'proq e'tibor qaratadi. 19.5-rasmda mijozning hisobvaraqqasidan pulni qanday olinish tavsiflangan hamkorlik diagrammasi keltirilgan. Rasmdan ko'rinib turibdiki, bu yerda ketma-ketlik diagrammasidagi barcha ma'lumotlar keltirilgan, ammo hamkorlik diagrammasida voqealar oqimi boshqacha tasvirlanadi. Undan ob'ektlar orasidagi aloqalarni tushunish osonroq, ammo voqealar ketma-ketligini tushunish qiyinroq.

Shu sababli, har ikkala turdagi diagrammalar ko'pincha har qanday ssenariy uchun yaratiladi. Garchi ular bir xil maqsadga xizmat qilsa va bir xil ma'lumotlarni o'z ichiga olgan bo'lsa ham, ularni turli nuqtayi nazardan taqdim etadilar.

Hamkorlik diagrammasida, xuddi ketma-ketlik diagrammasida bo'lgani kabi, millar ushbu foydalanish holatida almashinadigan xabarlarni bildiradi.



19.5- rasm. Pul olish jarayonini tavsiflovchi hamkorlik diagrammasi

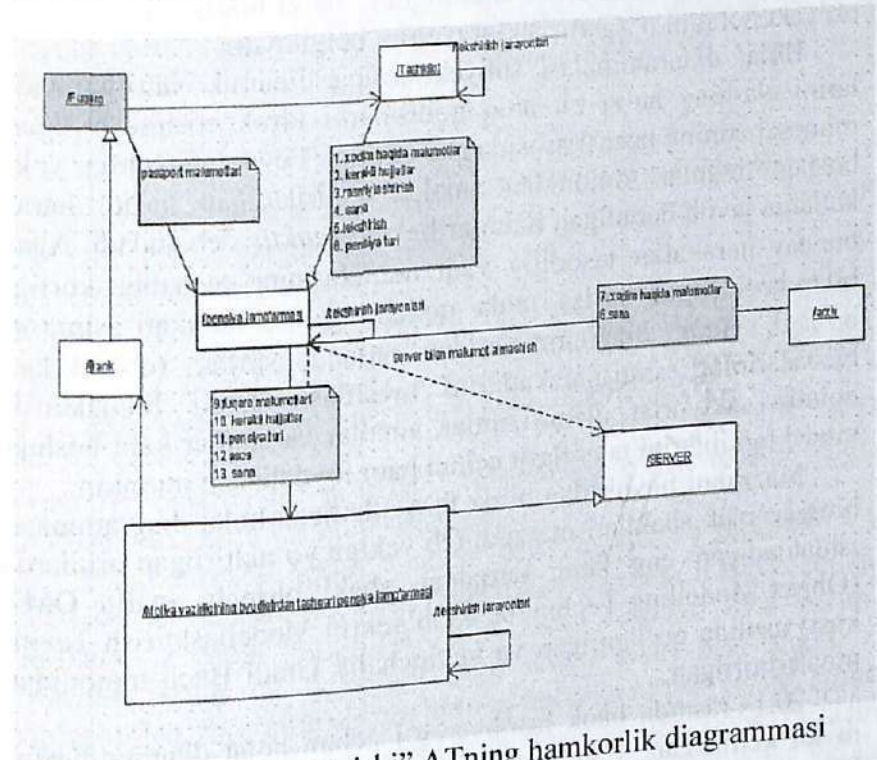
Biroq, ularning vaqt ketma-ketligi xabarlarini raqamlash orqali ko'rsatiladi.

### 19.3. "Pensiya ishi" ATning hamkorlik diagrammasi

Ketma-ketlik va hamkorlik diagrammalari modellashtirilgan mavzu sohasi elementlarining o'zaro ta'sirini ko'rsatadi. Ketma-ketlik diagrammalari o'zaro ta'sirning vaqtinchalik jihatlarini va faoliyatning xronologik ketma-ketligini, hamkorlik diagrammalari esa o'zaro ta'sirning strukturaviy tomonlarini ifodalaydi. Hamkorlik diagrammasi faqat o'zaro ta'sirda muayyan rol o'ynaydigan ob'ektlar orasidagi munosabatlarni tasvirlaydi. Tizim tuzilmasini hamkorlik diagrammasida o'zaro ta'sir qiluvchi ob'ektlar to'plami sifatida tasvirlash tizimining statik tavsifiga mos keladi.

Sinflovchi – bu tuzilmaviy va xulq-atvor xususiyatlarini tavsiflovchi mexanizm. Sinflovchilar namunalarga ega bo'lishi

mumkin bo'lgan modellashtirish elementlaridir. Sinflovchilarning tuzilmaviy xossalari atributlar bilan, xulq-atvor xususiyatlari esa amallar shaklida ifodalanadi. Xuddi shu sinflovchining barcha misollari bir qator umumiy xususiyatlarga ega. Sinflovchilar sinflar, interfeyslar, berilganlar turlari, signallar, tarkiblar, tugunlar, foydalanish holatlari va quyi tizimlarni o'z ichiga oladi. AgroUML-dagi tasniflagichlar orasidagi farqqa qaramay, ketma-ketlik va hamkorlik diagrammalarida bir xil piktogramma - to'rtburchaklar yordamida tasniflagichlar ko'rsatilgan.



19.5- rasm. "Pensiya ishi" ATning hamkorlik diagrammasi

## XX- BOB. HOLAT (AVTOMAT) DIAGRAMMASI

### 20.1. Nazariy asoslar

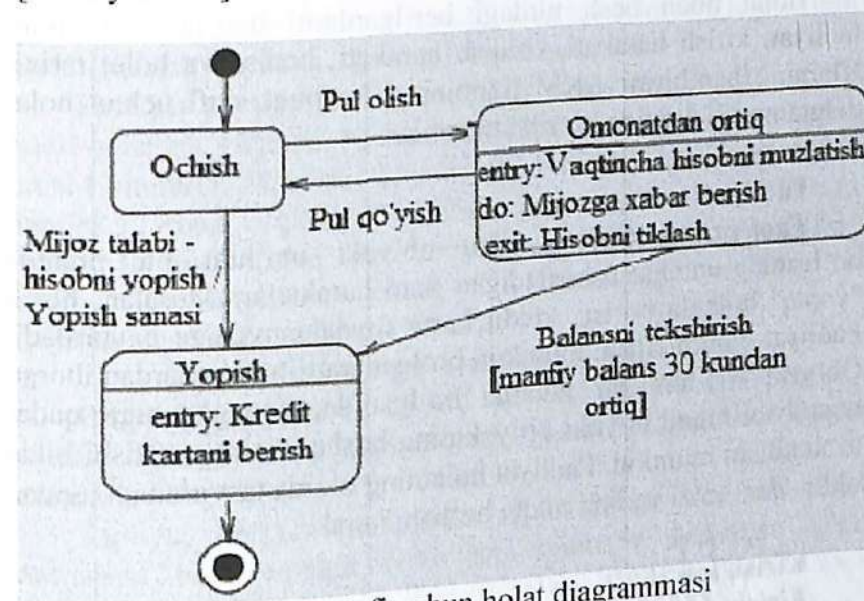
Holat diagrammasi ma'lum bir ob'yektning bo'lishi mumkin bo'lgan barcha holatlarini, shuningdek, ba'zi hodisalar natijasida ob'yekt holatini o'zgartirish jarayonini belgilaydi.

Holat diagrammalari subyektlarning dinamik xatti-harakatlarini ularning ba'zi bir aniq hodisalarni idrok etishga bo'lgan munosabatining tasnifi asosida aks ettiradi. Foydalanuvchilar yoki boshqa tizimlar tomonidan amalga oshiriladigan tashqi harakatlarga javob beradigan tizimlar ba'zan *reaktiv* deb ataladi. Agar bunday harakatlar tasodifiy vaqt nuqtalarining tashabbuskorligi bilan boshlangan bo'lsa, unda modelning xatti-harakati asinxron bo'ladi. Holat diagrammalaridan nafaqat sinflar (ob'yektlar) nusxalarining xatti-harakatlarini tavsiflash, balki foydalanish holatlari, aktyorlar, qism-tizimlar, amallar va usullar kabi boshqa model tarkiblarini tavsiflash uchun ham foydalanish mumkin.

Mazmuni bir-biridan biroz farq qiladigan holat diagrammasining ko'plab shakllari mavjud. Ob'yektga yo'naltirilgan usullarda ishlatiladigan eng keng tarqalgan shakl birinchi marta OMT (Object Modelling Technique - Ob'yektning Modellashtirish Texnikasi) usulida qo'llanilgan va keyinchalik Gradi Buch tomonidan moslashtirilgan.

20.1- rasmda bank hisobvarag'i uchun holat diagrammasiga misol keltirilgan. Ushbu diagrammadan hisobning bir holatdan boshqa holatga o'tish jarayonini ko'rishimiz mumkin. Masalan, agar mijoz ochiq hisobni yopishni talab qilsa, u "yopiq" holatga o'tadi. Mijozning talabi *hodisa (event)* deb ataladi, aynan shunday hodisalar bir holatdan ikkinchisiga o'tishni keltirib chiqaradi.

Agar mijoz ochiq hisobvarag'idan pul olsa, u "omonatdan ko'p" holatiga o'tishi mumkin. Bu faqat ushbu hisobdagi balans noldan kam bo'lsa sodir bo'ladi, bu bizning jadvalimizdagi [maxfiy balans] sharti bilan aks etadi. Kvadrat qavs ichidagi shart



20.1- rasm. Hisob sinfi uchun holat diagrammasi

(maxfiy balans) bir holatdan ikkinchisiga o'tish qachon sodir bo'lishi yoki bo'lmasligini belgilaydi.

Diagrammada ikkita maxsus holat mavjud - *boshlang'ich (ochish)* va *yakuniy (yopish)*. Dastlabki holat qora nuqta bilan ajratilgan, u yangi yaratilgan ob'yektning holatiga mos keladi. Yakuniy holat oq doiradagi qora nuqta bilan belgilanadi, u ob'yektning yo'q qilinishidan oldingi holatiga to'g'ri keladi. Holat diagrammasida bitta va faqat bitta boshlang'ich holat bo'lishi mumkin. Shu bilan birga, bizga qancha yakuniy holatlar kerak bo'lsa, shuncha bo'lishi mumkin yoki umuman bo'lmasligi ham mumkin.

Ob'yekt ma'lum bir holatda bo'lganda turli jarayonlar amalga

oshirilishi mumkin. Bizning misolimizda agar kredit oshib ketgan bo'lsa, mijozga tegishli xabar yuboriladi. Ob'yekt ma'lum bir holatda bo'lganda sodir bo'ladigan jarayonlar **harakatlar (actions)** deb ataladi.

Holat bilan besh turdagi berilganlarni bog'lash mumkin: faoliyat, kirish harakati, chiqish harakati, hodisa va holat tarixi. Ularning har birini AKM tizimining Account sinfi uchun holat diagrammasi asosida ko'rib chiqamiz.

### Faoliyat

**Faoliyat (activity)** – bu ob'yekt ma'lum bir holatda bo'lganida amalga oshiriladigan xatti-harakatlar. Masalan, hisob "yopiq" holatda bo'lsa, kredit karta foydalanuvchiga qaytariladi. Faoliyat – bu uzilishi mumkun bo'lgan xatti-harakatlardan iborat. Ob'yekt ma'lum bir holatda bo'lganida, u tugaguniga qadar bajarilishi mumkin yoki ob'yektning boshqa holatga o'tishi bilan to'xtatilishi mumkin. Faoliyat holatning o'zida tasvirlanadi, undan oldin **do:** so'zi va ikki nuqta bo'lishi kerak.

### Kirish harakati

**Kirish harakati (entry action)** – bu ob'yekt ma'lum bir holatga o'tganda bajariladigan xatti-harakatlar. Bank hisobvaragi misolida, u "Omonatdan ortiq" holatiga o'tganda, ob'yekt bu holatga qayerdan o'tganidan qat'i nazar, "hisobni vaqtincha muzlatish" harakati amalga oshiriladi. Shunday qilib, ushbu harakat ob'yekt ushbu holatga o'tgandan keyin emas, balki ushbu o'tishning bir qismi sifatida amalga oshiriladi. Faoliyatdan farqli o'laroq, kirish harakati uzluksiz deb hisoblanadi.

Kirish harakati holat ichida ham ko'rsatiladi, undan oldin **entry:** (kirish) so'zi va ikki nuqta bo'lishi kerak.

### Chiqish harakati

**Chiqish harakati (exit action)** kirishga o'xshaydi. Biroq, bu ushbu holatdan chiqish jarayonining ajralmas qismi sifatida amal-

ga oshiriladi. Bizning misolimizda hisob qaydnomasi ob'yekti "Omonatdan ortiq" holatidan chiqqanda, qayerga o'tishidan qat'iy nazar, "hisobni tiklash" harakati amalga oshiriladi. Bu bunday o'tish jarayonining bir qismidir. Kirish kabi, chiqish harakati ham uzluksizdir. Chiqish harakati holat ichida tasvirlanadi, undan oldin **exit:** (chiqish) so'zi va ikki nuqta bo'lishi kerak.

Faoliyat paytida kirish va chiqish harakatlarida ob'yektning xatti-harakati voqeani boshqa ob'yektga yuborishni o'z ichiga olishi mumkin. Masalan, account (hisob) ob'yekti voqeani Card reader (kartani o'qish) ob'yektiga yuborishi mumkin. Bunday holda, faoliyat, kirish harakati yoki chiqish harakati tavsifidan oldin "^" belgisi qo'yiladi. Diagrammadagi tegishli satr quyidagicha ko'rinishga ega bo'ladi:

### Do: ^Maqsad.Hodisa (Argumentlar)

Bu yerda **Maqsad** – hodisani qabul qiluvchi ob'yekt, **Hodisa** – yuborilgan xabar va **Argumentlar** – yuborilgan xabarning parametrlari.

Faoliyat ob'yekt tomonidan ma'lum bir hodisani olish natijasida ham amalga oshirilishi mumkin. Masalan, hisob ob'yekti **Ochiq** holatida bo'lishi mumkin. Muayyan hodisa sodir bo'lganda aniq bir faoliyat amalga oshiriladi.

**O'tish (Transition)** – bu bir holatdan ikkinchisiga o'tish. Diagrammadagi o'tishlar to'plami, ob'yekt o'z holatlari o'rasida qanday harakat qilishi mumkinligini ko'rsatadi. Diagrammadagi barcha o'tishlar dastlabki holatdan boshlanib, keyingi holat bilan tugaydigan mil shaklida tasvirlanadi.

O'tishlar refleksli bo'lishi mumkin. Ob'yekt hozirgi daqiqadagi holatiga qaytib o'tishi mumkin. Refleksli o'tishlar xuddi shu holatda boshlanadigan va tugaydigan mil shaklida tasvirlanadi. O'tishning bir nechta tasnifi mavjud. Ular **hodisalar, shartlarni o'rab turgan dalillar, harakatlar va yuboriladigan voqealarni** o'z ichiga oladi. Ularning har birini AKM misoli asosida ko'rib chiqamiz.

## Hodisalar

**Hodisa (event)** – bu bir holatdan ikkinchisiga o'tishni keltirib chiqaradigan narsa. Bizning misolimizdagi "mijozning talabi – hisobni yopish" hodisasi hisobning ochiq holatdan yopiq holatga o'tishiga olib keladi. Hodisani diagrammada o'tish chizig'i bo'ylab joylashtirishadi.

Diagrammada biz hodisani ko'rsatish uchun amal nomidan ham, oddiy iboradan ham foydalanishimiz mumkin. Bizning misolimizda voqealar oddiy iboralar bilan tasvirlangan. Agar biz amallardan foydalanmoqchi bo'lsak, "mijozni talabi - hisobni yopish" hodisasini **Yopish So'rovi ( ) (RequestClosure)** deb atash mumkin.

Voqealar argumentlarga ega bo'lishi mumkin. Shunday qilib, hisobning "omonatdan ortiq" holatidan "ochish" holatiga o'tishiga olib keladigan "omonat qo'yish" hodisasi omonat miqdorini tavsiflovchi Amount (miqdor) argumentiga ega bo'lishi mumkin. Aksariyat o'tishlarda hodisalar bo'lishi kerak, chunki ular birinchi navbatda o'tishni amalga oshirishga majbur qiladi. Shunga qaramay, hodisalar bo'lmagan avtomatik o'tishlar ham mavjud. Bunday holda ob'yektning o'zi kirish harakati, faoliyat va chiqish harakatlarini amalga oshirishga imkon beradigan tezlikda bir holatdan ikkinchisiga o'tadi.

### Yopiq shartlar

**Yopiq shartlar (guard conditions)** qachon o'tish mumkinligini va qachon amalga oshirilmasligini aniqlaydi. Bizning misolimizda "pul qo'yish" hodisasi hisobni "omonatdan ortiq" holatidan "ochiq" holatiga o'tkazadi, agarda balans noldan katta bo'lsa. Aks holda, o'tish amalga oshmaydi.

Chegara shartlari diagrammada voqea nomidan keyin o'tish chizig'i bo'ylab tasvirlanadi va ular kvadrat qavs ichiga olinadi. Yopiq shartlarni o'rnatish shart emas. Ammo, agar holatda bir nechta avtomatik o'tish mavjud bo'lsa, ular uchun o'zaro inkor qiluvchi yopiq shartlarni aniqlash kerak. Bu diagramma o'quvchi-

sig qaysi o'tish yo'li avtomatik ravishda tanlanishini tushunishga yordam beradi.

## Harakat

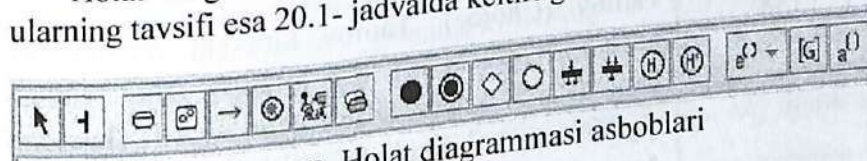
**Harakat (action)**, yuqorida aytib o'tilganidek, o'tishning bir qismi sifatida amalga oshiriladigan uzluksiz xatti-harakatlardir. Kirish va chiqish harakatlarini holatlar ichida ko'rsatiladi, chunki ular ob'yekt unga kirganda yoki undan chiqqanda nima bo'lishini aniqlaydi. Biroq, aksariyat harakatlar o'tish chizig'i bo'ylab tasvirlanadi, chunki ular holatga kirishda yoki chiqishda amalga oshirilmasligi kerak.

Masalan, hisob ochiqdan yopiq holatga o'tganda, "hisobning yopilish sanasini saqlash" harakati amalga oshiriladi. Ushbu uzluksiz xatti-harakatlar faqat "ochiq" holatdan "yopiq" holatga o'tish paytida amalga oshiriladi. Harakat voqea nomidan keyin o'tish chizig'i bo'ylab chiziladi, undan oldin qiyshiq chiziq chiziladi. Hodisa yoki harakat ob'yekt ichidagi xatti-harakatlar bo'lishi mumkin yoki boshqa ob'yektga yuborilgan xabar bo'lishi mumkin. Agar hodisa yoki harakat boshqa ob'yektga yuborilsa, uning oldida diagrammada "^" belgisi qo'yiladi.

Har bir sinf uchun holat diagrammalarini yaratish shart emas, ular faqat murakkab holatlarda qo'llaniladi. Agar sinf ob'yekti bir nechta holatlarda mavjud bo'lishi mumkin bo'lsa va ularning har biri boshqacha harakat qilsa, u uning uchun diagramma talab qilinishi mumkin.











## 20.2. Holat diagrammasini qurish asboblari








Holat diagrammasini qurish uchun asboblari 20.2- rasm, ularning tavsifi esa 20.1- jadvalda keltirilgan.

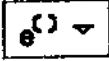


20.2- rasm. Holat diagrammasi asboblari





20.1- jadval. Holat diagrammasi asboblarning tavsifi


Asbob	Tavsifi
	Oddiy holat
	Kompozit (tarkibli) holat
	Yangi o'tish (New Transition)
	Sinxronlash holati
 Submachine State	Holatdan – joylangan avtomatga havola
 StubState	Holatni – so'ndirgich
	Boshlang'ich holat
	So'ngi (oxirgi) holat
 Junction	Tarmoq. Bitta kiruvchi o'tishni bir nechta chiquvchi o'tishga ajratish uchun ishlatiladi. O'tish paytida qo'riqchi sharti "haqiqat" qiymatini qabul qiladigan o'tish orqali amalga oshiriladi. "Agarda" deb belgilangan oldindan belgilangan shartni faqat bitta o'tish uchun aniqlash mumkin, agar boshqa o'tish uchun qo'riqchi shartlar "yolg'on" ma'nosini qabul qilsa.
	Tanlov (Choice). Tanlov kiruvchi o'tishni bir nechta chiquvchi o'tishga ajratish uchun ishlatiladi. O'tish paytida qo'riqchi sharti "haqiqat" qiymatini qabul qiladigan o'tish amalga oshiriladi. Agar bir nechta o'tish uchun shart "haqiqat"

	qiymatini qabul qilsa, unda bajarilgan o'tishni tanlash tasodifan amalga oshiriladi. Agar boshqa o'tish uchun qo'riqchi shartlari "yolg'on" ma'nosini qabul qilsa, "agarda" deb belgilangan oldindan belgilangan shartni faqat bitta o'tish uchun aniqlash mumkin bo'ladi
	Tarmoqlanish
	Birlashtirish
	Yaqindagi tarixiy holat (shallow history state)
	Eski tarixiy holat (deep history state). Joriy qism - avtomat uchun ixtiyoriy ichma-ichlikning barcha qism-holatlarini eslab qolish uchun ishlatiladi
	O'tish uchun trigger sifatidagi hodisani chaqirish. Chaqirishning 4ta turi mavjud – chaqirish hodisasi; o'zgartirish hodisasi; signal hodisasi va vaqt hodisasi (20.2- jadval)
	Qo'riqchi sharti (guard condition) trigger hodisadan keyin kvadrat qavs ichida yoziladigan mantiqiy ifodadir. Agar qo'riqchi sharti "haqiqat" qiymatini qabul qilsa, u holda tegishli o'tish ishlab chiqarilishi mumkin, natijada ob'ekt maqsadli holatga o'tadi. Agar qo'riqchi sharti "yolg'on" qiymatini qabul qilsa, u holda o'tish ishlamaydi
	Bir holatdan ikkinchisiga o'tishga olib keladigan harakatni chaqirish. Uning 7 turi mavjud – chaqirish harakati; yaratish harakati; yo'q qilish harakati; qaytarish harakati; yuborish harakati; sharhlanmagan harakat; yangi ketma-ket harakat (20.3- jadval)









Hodisalarni tanlash uchun pastga ochiluvchi menyuga . U 4 turdagi hodisalarni chaqiradi. Ularning tavsifi 20.2- jadvalda keltirilgan


20.2- jadval. Hodisalarni tanlash asboblari

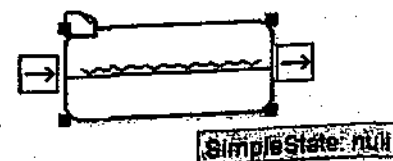
Asbob	Tavsifi
	Chaqirish hodisasi
	O'gartirish hodisasi
	Signal hodisasi
	Vaqt hodisasi

Harakatni chaqirish variantlarini tanlash uchun pastga ochiluvchi menyuga . U 7 turdagi harakatni chaqirish variantlarini tanlash imkoniyatini yaratadi. Bularning tavsifi 20.3- jadvalda keltirilgan.


20.3- jadval. Harakatni chaqirish asboblari

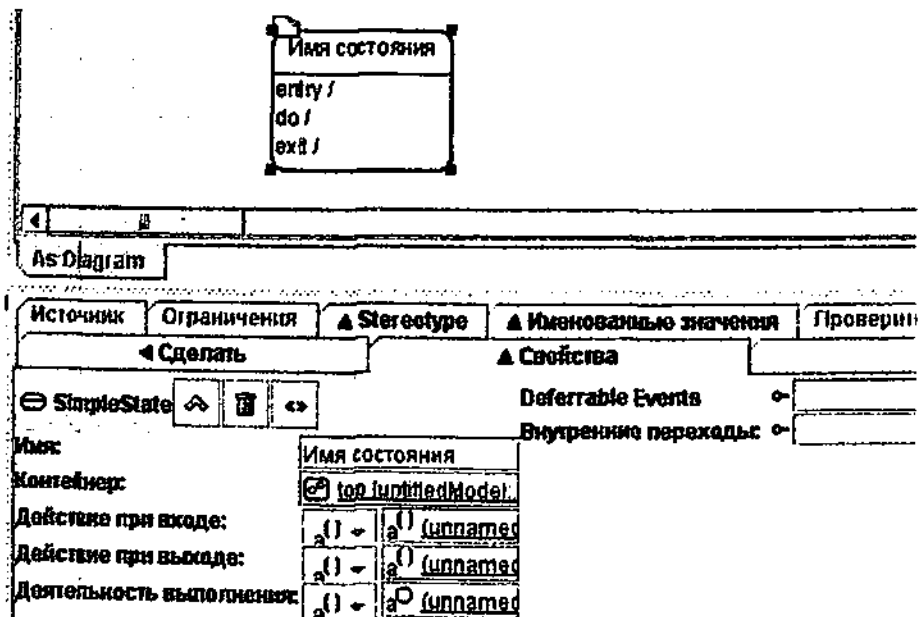
Asbob	Tavsif
	Chaqirish harakati
	Yaratish harakati
	Yo'q qilish harakati
	Qaytish harakati
	Jo'natish harakati
	Oxirgi holat
	Sharhlanmas harakat
	Yangi ketma-ket harakat

Holat (state), tizim elementlarining xatti-harakatlarining dinamik yoki funksional jihatlarini akslantiruvchi xususiyatlarining qiymati bilan tavsiflanadi. Yangi oddiy holatni qo'shish uchun  biz 20.2- rasmda ko'rsatilgan asboblar panelidagi belgini bosishimiz kerak. Tahrirlash chizimasidagi kursorni yangi holat joylashtiriladigan joyga ko'chirgandan so'ng, chap tugmani bosishimiz kerak. Shundan so'ng, tahrirlash panelida oddiy holatni tasvirlaydigan qiyofa paydo bo'ladi (20.3- rasm). Qizil to'liqlik chiziq holat nomini kiritish uchun mo'ljallangan joyni ta'kidlaydi. Holat gorizontall chiziq bilan ikki qismga bo'linadi, yuqori qismida holat nomi, pastki qismida uning ichki harakatlari ro'yxati bo'ladi.



20.3-rasm. Oddiy holatni diagrammaga qo'shish



Qizil to'liqlik chiziqdagi ko'rsatgichni ikki marta bosgandan so'ng, biz diagrammaga holat nomini qo'shishimiz mumkin. Ichki harakatlar nomlari – entry (kirish), do (bajariladigan), exit (chiqish) qismlar chizimasida joylashgan «Действие при входе» (kirish harakati), «Действие при выходе» (chiqish harakati) va «Деятельность выполнения» (bajarish faoliyati) vositalari yordamida qo'shilishi mumkin (20.4- rasm). Harakatlar turini  belgini bosgandan so'ng tafsilotlar chozimidagi ochiladigan ro'yxatdan tanlash mumkin.




20.4- rasm. Oddiy holatdagi ichki harakatlar ro'yxatini qo'shish

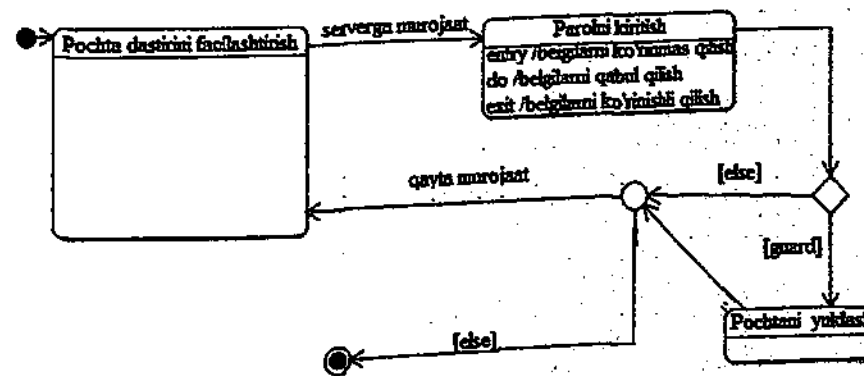
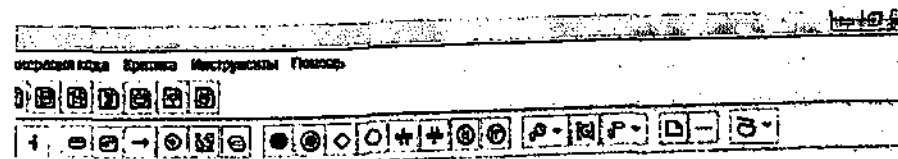
Ichki harakatlar nomlari tahrirlash panelidagi oddiy holatni tasvirlaydigan to'rtburchakning pastki qismidagi ichki harakatlar ro'yxati maydonini ikki marta bosgandan so'ng kiritiladi.

Elektron pochta ishini tavsiflovchi misol uchun holat diagrammasi namunasi 20.5- rasmda keltirilgan. Diagrammada (20.5-


rasm) boshlang'ich holat ko'rsatilgan,  belgiga mos keladigan uchta oddiy holat "pochta dasturini faollashtirish", "parolni kiritish", "pochtani yuklash" va  belgiga mos keladigan yakuniy holat.

Diagrammadagi  piktogramma 20.1- jadvalda tasvirlangan asbobni (tarmoqlanishni) ko'rsatadi. U "parolni kiritish" holatidan kelib chiqadigan o'tishni ikkita o'tishga ajratish uchun ishlatiladi. Ulardan biri "pochtani yuklash" holatiga

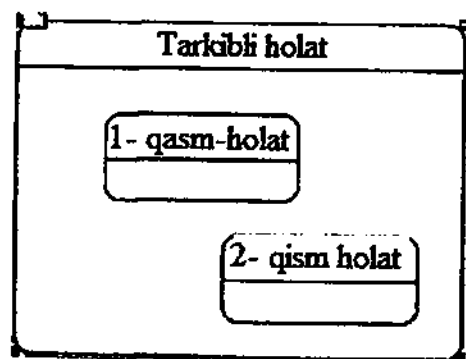
yo'naltirilgan, agar qo'riqchi sharti [parol to'g'ri kiritilgan] bo'lsa, u "haqiqat" qiymatini oladi. Diagrammadagi bu o'tish [guard] bilan ko'rsatilgan. "Else" deb belgilangan shart, agar qo'riqchi sharti [parol to'g'ri kiritilgan] bo'lsa, "yolg'on" qiymatini oladi.



20.5- rasm. Holat diagrammasi

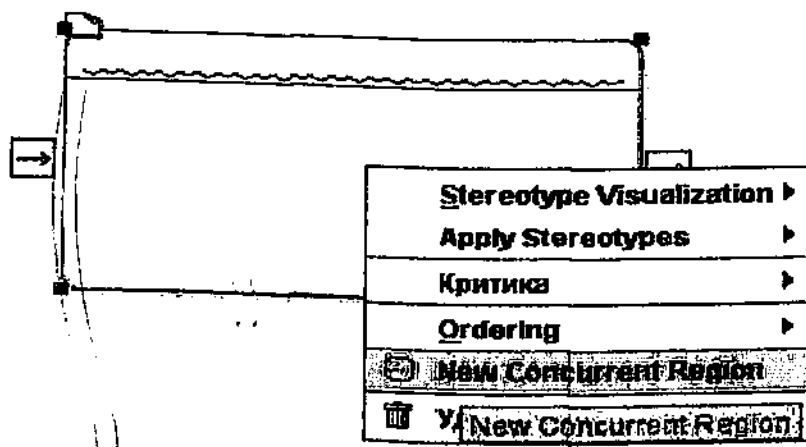
Diagrammadagi "Tanlash"  belgisi 20.1- jadvalda tasvirlangan asbobni ko'rsatadi. U elektron quti dasturiga qayta kirishni yoki chiqishni tanlash uchun ishlatiladi.

Kompozit holat (composite state) 20.1- jadvalda tasvirlangan asbob yordamida yaratiladi. Bu shunday holatki, uning ichida unga o'rnatilgan qism-holat (substate) mavjud.



20.6- rasm. Tarkibli holat

Kompozit holat va holatlar o'rasida kompozitsiyaning munosabati mavjud. Qism-holatlar ketma-ket va parallel bo'lishi

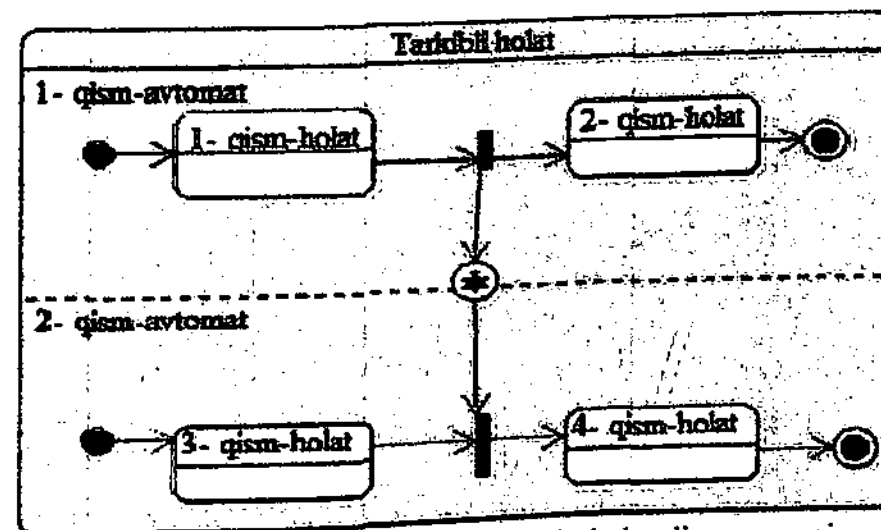


20.7- rasm. «New Concurrent Region» buyrugini tanlash

mumkin. Diagrammada parallel qism-holatlarni qurish uchun kompozit holat ichidagi maydonni gorizontali uzlukli chiziqlar bilan ajratish kerak, ya'ni yangi raqobatbardosh maydon tanlanadi. Buning uchun kursorni kompozit holat to'rtburchaklar egallagan

maydonga qo'yib, sichqonchani o'ng tugmachasini bosishimiz kerak va ochiladigan menyuda (20.7- rasm) "New Concurrent Region" buyrug'ini tanlang.

20.8- rasmda kompozitsion holat ichidagi maydon uzlukli chiziq bilan ikkita parallel jarayonga bo'linadi. Parallel qism-holatlar (concurrent substates) kompozit hodisa ichida parallel ravishda bajarilishi mumkin bo'lgan ikki yoki undan ortiq qism-holatlarni aniqlashga imkon beradi. Parallel qism-holatlar, o'z navbatida, ketma-ketlardan iborat bo'lishi mumkin. Masalan, 20.8- rasmdagi 1- qism-holat va 2- qism-holat 1- qism-avtomatning ketma-ket qism-holatlaridir. 2- qism-avtomatning ketma-ket qism-holatlari bu 3- va 4- dan iborat.



20.8- rasm. Parallel qism-avtomatli holatlar diagrammasi

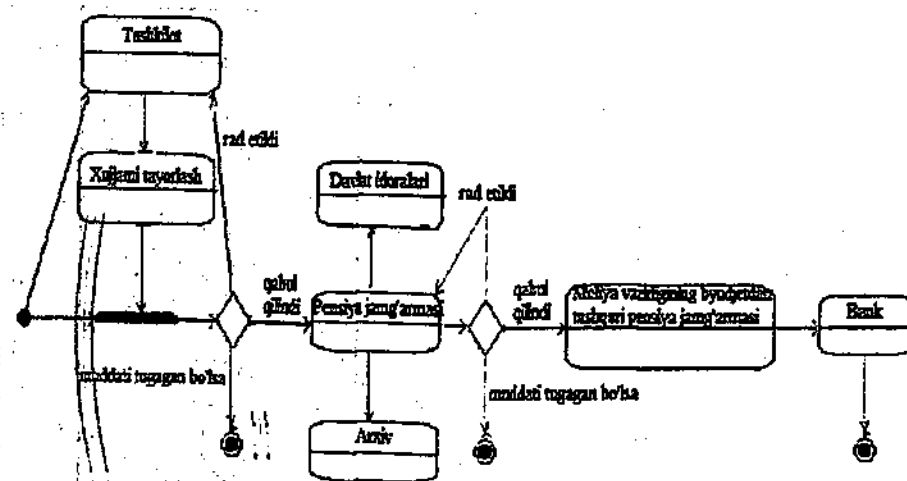
UML tilida shaxsiy hodisalarning sodir bo'lishini sinxronlashtirishni hisobga olish uchun maxsus mavhum-holat mavjud bo'lib, u *sinxronizatsiya holati* deb ataladi. Diagrammada 1- qism-avtomatning 1-qism-holat va 2- qism-holat orasidagi o'tish joylari, hamda 2- qism-avtomatning 3- va 4- qism-holatlari

moslashtirilgan (o'tish va uzlukli chiziq kesishmasidagi belgisi).



### 20.3. "Pensiya ishi" ATning holat diagrammasi

"Pensiya ishi" ATning holat diagrammasi – bu ob'ektlarining xatti-harakatlarini modellashtirish uchun ishlatiladigan matematik abstraksiya.



20.9- rasm. "Pensiya ishi" ATning holat diagrammasi

"Pensiya ishi" ATning holat diagrammasi (20.9- rasm) quyidagi algoritm asosida ishlaydi:

1. Fuqaro tashkilot orqali yoki to'g'ridan-to'g'ri o'zi hududidagi pensiya jamg'armasiga murojaati romb orqali ifodalanadi.
2. Romb – bu tekshirish jarayoni va kutish vaqtini anglatadi, agarda murojaat qilgan fuqaroning hujjatlarini o'rganish jarayonida kamchiliklar topilsa, kam bo'lgan hujjatlarni pensiya jamg'armasi tashkilotiga yoki fuqaroning o'ziga murojaat qilib belgilangan vaqt mobaynida taqdim etilishini talab qiladi.

3. Agarda belgilangan vaqt mobaynida kerakli hujjatlar tashkilot yoki fuqaro tomonidan taqdim etilmasa, ish jarayoni to'xtatiladi.

4. 1-3 qadamlar asosida pensiya jamg'armasi to'plangan hujjatlar asosini moliya vazirligining byudjetdan tashqari pensiya jamg'armasiga yuboradi.

5. Agarda hujjatlarda kamchilik topilsa, moliya vazirligining byudjetdan tashqari pensiya jamg'armasi ham hudud pensiya jamg'armasiga belgilangan vaqt mobaynida hujjatlarni taqdim etilishini so'raydi.

6. Belgilangan vaqt mobaynida kerakli hujjatlar hudud pensiya jamg'armasi tomonidan taqdim etilmasa, ish jarayoni to'xtatiladi.

7. Hamma hujjatlar to'liq va kamchiliksiz bo'lsa, moliya vazirligining byudjetdan tashqari pensiya jamg'armasi tayinlangan pensiyani bank hisobiga o'tkazadi va bank orqali fuqarolarga yetkaziladi.

## XXI- BOB. FAOLIYAT DIAGRAMMASI

### 21.1. Nazariy asos

Boshqa UML vositalaridan farqli o'laroq, faoliyat diagrammalari Buch, Rambo va Jakobsonning oldingi manbalarida aniq manbaga ega emas va bir nechta turli usullardan, xususan, holatlarini modellashtirish usuli va Petri tarmoqlaridan g'oyalarni olgan.

Ko'pgina boshqa xatti-harakatlarni modellashtirish vositalari singari, faoliyat diagrammalari ham ma'lum afzallik va kamchiliklarga ega, shuning uchun ularni boshqa vositalar bilan birgalikda qo'llagan ma'qul.

**Faoliyat (activity, o'zaro harakat)** diagrammalarining eng katta afzalligi parallellikni qo'llab-quvvatlashidir. Shu sababli, ular ish oqimlarini modellashtirish va asosan parallel dasturlashning kuchli vositasidir. Ularning eng katta kamchiligi shundaki, harakatlilar va ob'yektlar o'rasidagi aloqalar aniq ko'rinmaydi.

Ushbu aloqalarni aniqlashga harakat qilish mumkin, bunda faoliyat uchun ob'yektlarning nomi bilan tamg'alash usulidan foydalanish mumkin, ammo bu usul o'zaro harakat diagrammasidagi kabi yaxsi natija bermaydi.

Faoliyat diagrammalaridan quyidagi holatlarda foydalanish afzalroq:

- ❖ foydalanish holatlarini tahlil qilishda. Ushbu bosqichda biz harakatlar va ob'yektlar o'rasidagi bog'liqlik bilan qiziqmaymiz, faqat qanday harakatlar sodir bo'lishi kerakligini va tizimning xatti-harakatlaridagi tobelikni tushunishimiz kerak. Usullar va ob'yektlarni bog'lash keyinchalik o'zaro harakat diagrammalari yordamida amalga oshiriladi;

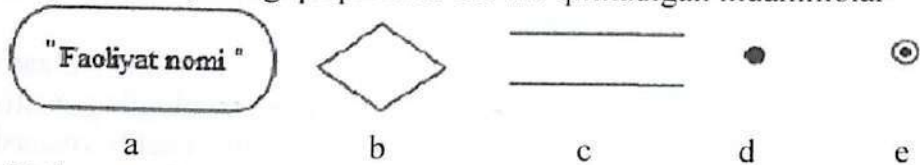
- ❖ turli xil foydalanish holatlaridagi ish oqimlarini (workflow) tahlil qilishda. Foydalanish holatlari bir-biri bilan o'zaro harakatlanganda, faoliyat diagrammalari ularning xatti-harakatlarini ifodalash va tahlil qilishning kuchli vositasi hisoblanadi.

Faoliyat diagrammasi tizim tomonidan amalga oshiriladigan amallarning algoritmik va mantiqiy bajarilishining xususiyatlarini aks ettiradi. Algoritmik yechimlarni aks ettirish uchun axborot tizimlarini ishlab chiqishda tizimli yondashuvda an'anaviy ravishda blokli chizmalar ishlatilgan. Bunda faoliyat deganda, qo'l-da yoki avtomatlashtirish vositalari yordamida amalga oshirilishi kerak bo'lgan amal tushuniladi. Har bir foydalanish holatiga o'z vazifalar ketma-ketligi mos keladi. Nazariy jihatdan faoliyat diagrammasi, tahlil qilinadigan foydalanish holatini amalga oshiruvchi umumiy algoritmdir. Diagrammada faoliyat yumaloq burchakli to'rtburchak bilan ifodalanadi (21.1- rasmdagi a).

Faoliyat diagrammasi muqobil va parallel jarayonlarni tavsiflashga imkon beradi. Shu bilan bir qatorda, muqobil jarayonlarni belgilash uchun romb (21.1- rasmdagi b) ishlatiladi, shart chap yoki o'ng tomonda ko'rsatiladi va "ha", "yo'q" muqobillari tegishli chiqishlarning yonida ko'rsatiladi. Xuddi shu blok bilan biz siklik jarayonni qurishimiz mumkin. Faoliyat faollashuvining karraligi faoliyatni faollashtirish milining yonida joylashgan "\*" belgisi bilan belgilanadi va agar kerak bo'lsa, "har bir satr uchun" yozuvi bilan belgilanadi. Parallel jarayonlarni belgilash uchun moslashtirish chizig'i (21.1- rasmdagi c) ishlatiladi va moslashtirish holati diagrammada ko'rsatiladi. Faoliyat diagrammasining boshlanishi (21.1- rasmdagi d) va tugashi (21.1- rasmdagi e) belgilar bilan ko'rsatiladi.

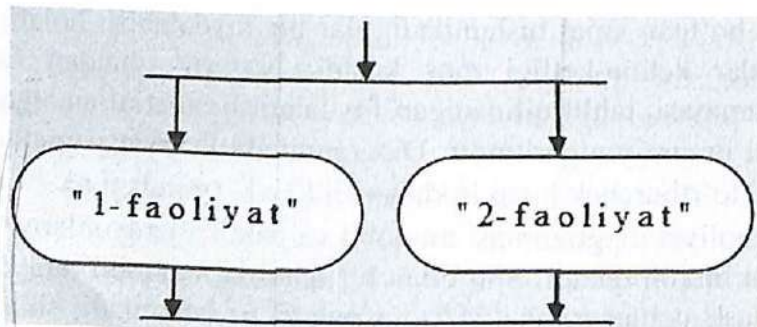
21.2- rasmda "1- faoliyat" va "2- faoliyat" parallel ravishda amalga oshirilishi mumkinligini ko'rsatadi. Tasniflarni aniqlash bosqichida faqat foydalanish holatlarini aniqlab olish mantiqan

to'g'ri keladi, uning qisqacha ta'rifi hal qilinadigan muammolar-



21.1- rasm. Faoliyat diagrammasidagi shartli belgilar: a - faoliyat; b - tanlov; c - chiziqli moslashtirish; d - boshlang'ich; e - yakuniy

ning mohiyatini tushunish uchun yetarli emas. Shunday qilib, faoliyat diagrammalari foydalanish holatlarini ta'riflash o'rniga yoki ularga qo'shimcha sifatida ishlatilishi mumkin.



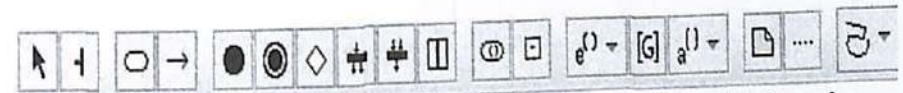
21.2- rasm. Jarayonlarning parallelligini ko'rsatuvchi faoliyat diagrammasi

Faoliyat diagrammasi amalga oshiriladigan amallar ketma-ketligini tavsiflaydi va protsedurali va moslashgan boshqaruv xususiyatlarini aks ettiradi. Ularni holat diagrammalarining xususiy naqli deb hisoblash mumkin.

### 21.2. Faoliyat diagrammasi asboblari

ArgoUMLda faoliyat diagrammasini yaratish uchun asboblar paneli 21.3- rasmda keltirilgan. Faoliyat va holat diagrammalarida

holatlar, o'tishlar, kuzatuvlar, bo'limlar, boshlang'ich va yakuniy holatlarning belgilari mavjud. Farqi harakatlarni ifodalash uchun ishlatiladigan holatlarning semantikasida yotadi. Faoliyatning har bir holati ba'zi bir boshlang'ich amalning bajarilishiga mos keladi va faoliyat jadvalidagi keyingi holatga o'tish avvalgi holatdagi amal tugagandan so'ng amalga oshiriladi.



21.3- rasm. Faoliyat diagrammasining asboblar chizimasi

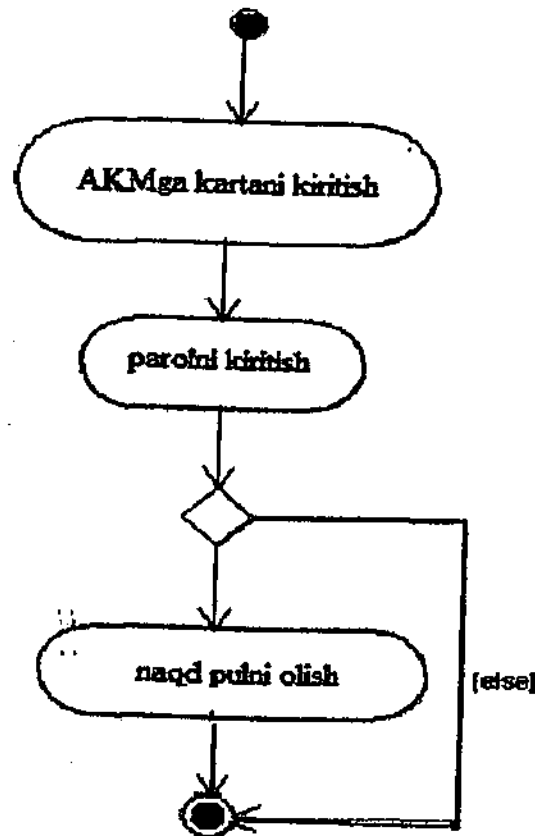
21.1- jadvalda faqat holat diagrammasini yaratish vositalaridan farq qiluvchi faoliyat diagrammasini yaratish asbob keltirilgan.

21.1- jadval. Faoliyat diagrammasini yaratish asbob


Asboblar	Tavsifi
	Faoliyatning yangi holati
	Yo'llar (swimlanes). Faoliyatning holatini faoliyat diagrammasida, vertikal chiziqlar bilan bir-biridan ajratilgan guruhlarini belgilash uchun ishlatiladi.
	Holatni chaqirish
	Holatlar orasidagi o'tish ob'yekti



Faoliyat diagrammasining to'liqlik darajasiga qarab sinf diagrammasi kabi rivojlanishning turli bosqichlarida qo'llaniladi. Talablarni tahlil qilish va faoliyat jadvalining xususiyatlarini aniqlashtirish bosqichida, biz ishlab chiqilgan dasturiy ta'minotning asosiy funksiyalarini aniqlashtirishga imkon beradi.

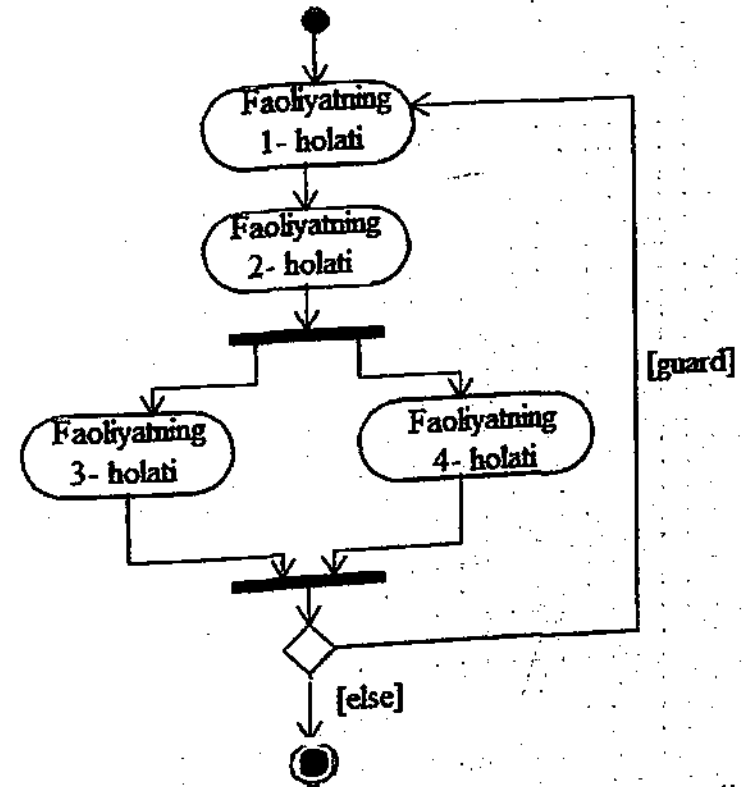
Grafik jihatdan faoliyat diagrammasi faoliyatning cho'qqilari bo'lgan faoliyat grafi shaklida taqdim etiladi va yo'lar harakatning bir holatidan ikkinchisiga o'tishi ko'satiladi. Faoliyatning eng oddiy diagrammasi 21.4- rasmda ko'rsatilgan.




21.4- rasm. Faoliyat diagrammasi

Faoliyat holati parallel gorizontalar va yon yo'lar bilan chegaralangan raqam bilan ifodalanadi. Yangi faoliyat holatini qo'shish uchun  piktogramma asbobi ishlatiladi. Parallel jarayonlarni taqdim etish uchun faoliyat jadvallarida

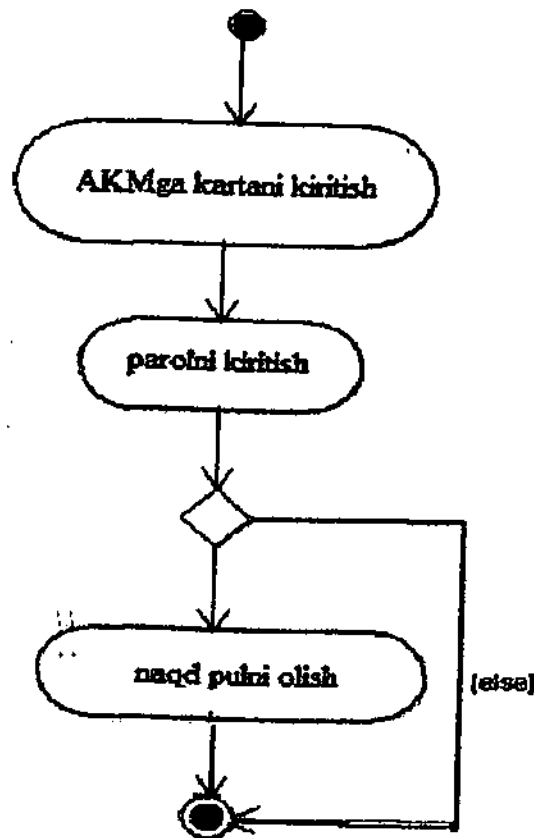
maxsus vositalar mavjud -  "tarmoqlanish" va  "birlashtirish". 21.5- rasmda uchinchi va to'rtinchi holatlarga mos keladigan jarayonlar parallel ravishda amalga oshiriladigan faoliyat diagrammasi keltirilgan.




21.5- rasm. Ketma-ket va parallel amallar bajaruvchi faoliyat diagrammasi



ArgoUMLning parallel jarayonlarni taqdim etishining yana bir vositasi - "Track" (swimlanes)  vositasi. Bu korxonalar

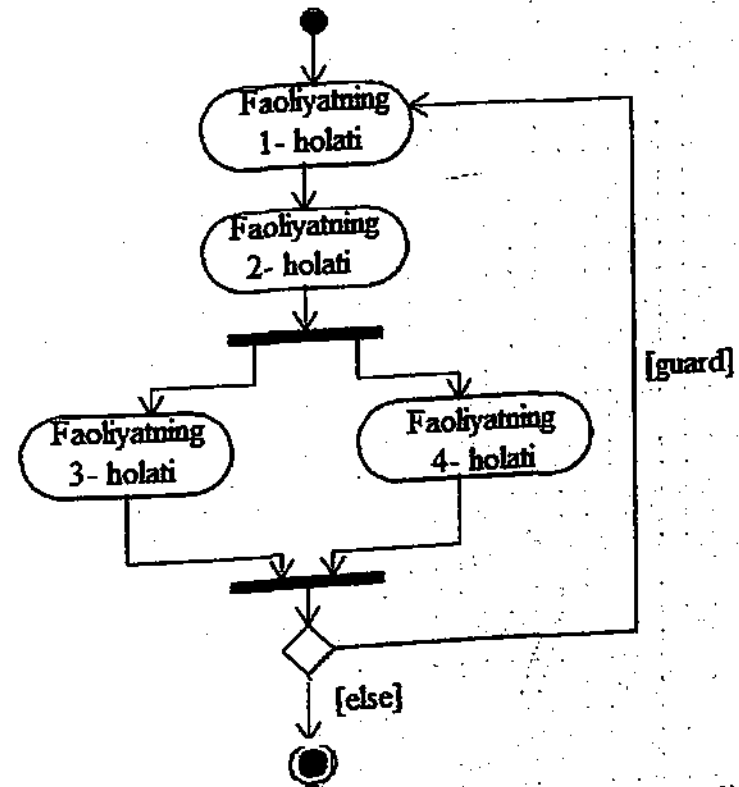
Grafik jihatdan faoliyat diagrammasi faoliyatning cho'qqilari bo'lgan faoliyat grafigi shaklida taqdim etiladi va yo'lar harakatining bir holatidan ikkinchisiga o'tishi ko'satiladi. Faoliyatning eng oddiy diagrammasi 21.4- rasmda ko'rsatilgan.




21.4- rasm. Faoliyat diagrammasi

Faoliyat holati parallel gorizontaal chiziqlar va yon yo'lar bilan chegaralangan raqam bilan ifodalanadi. Yangi faoliyat holatini qo'shish uchun  piktogramma asbobi ishlatiladi. Parallel jarayonlarni taqdim etish uchun faoliyat jadvalarida

maxsus vositalar mavjud -  "tarmoqlanish" va  "birlashtirish". 21.5- rasmda uchinchi va to'rtinchi holatlarga mos keladigan jarayonlar parallel ravishda amalga oshiriladigan faoliyat diagrammasi keltirilgan.




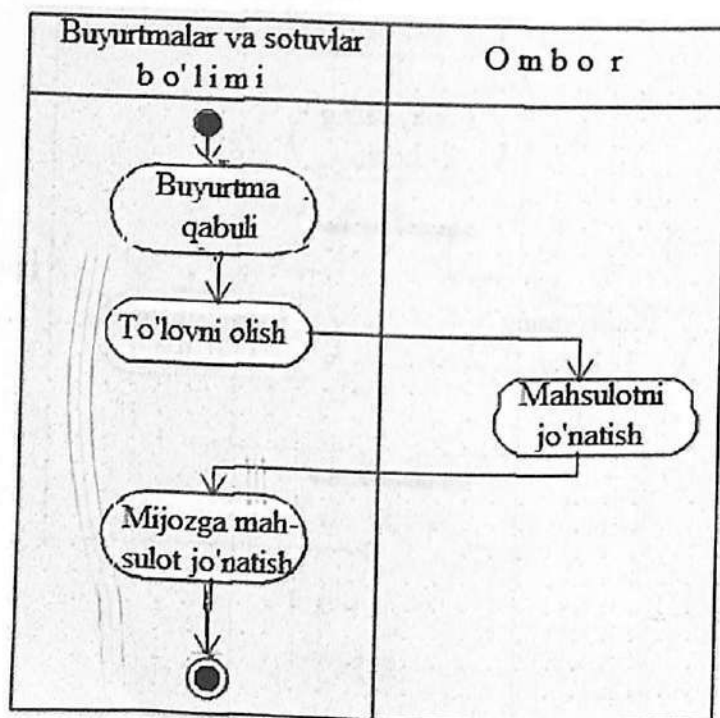
21.5- rasm. Ketma-ket va parallel amallar bajaruvchi faoliyat diagrammasi

ArgoUMLning parallel jarayonlarni taqdim etishining yana bir vositasi - "Track" (swimlanes)  vositasi. Bu korxonalar

(tijorat-jarayonlar) ishini taqlid qilish imkonini beradi, hamda korxonada bo'limlari o'rasida o'zaro aloqani o'z ichiga oladi.

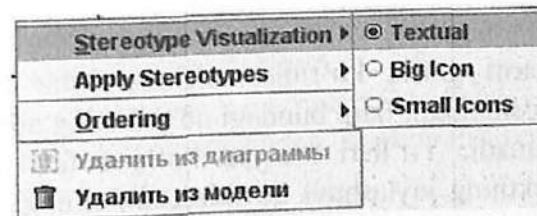
21.6- rasmda korxonaning ikki bo'limini modellashtirish uchun "trek" vositasidan foydalanadigan faoliyat diagrammasini

ko'rsatilgan. Bu  asbobdan kerakli sonli treklarni olish uchun ko'p marta ishlatilish mumkin. Tahrirlash panelidagi treklar yonma -yon joylashtirilgan va bir xil uzunlikka ega.



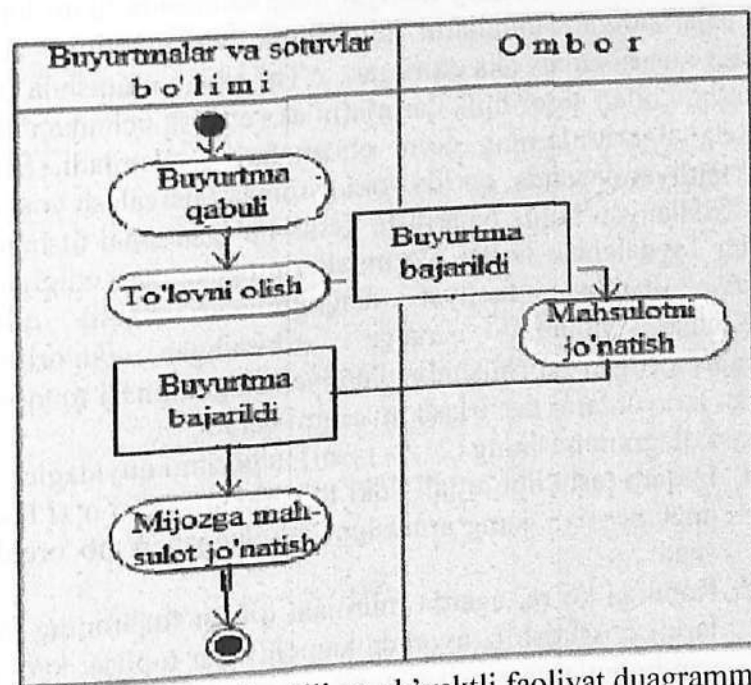
21.6- rasm. Parallel yo'llik faoliyat diagrammasi

Faoliyat jadvalidagi yozuvlarga yangi yozuv qo'shish uchun o'ng tugmani bosib va ochiladigan menyudan foydalaning (21.6- rasm).



21.7- rasm. Yozuvlarni qo'shish uchun ochiluvchi menyular

Umuman olganda, faoliyat diagrammasidagi harakatlar muayyan ob'ektlar ustida amalga oshiriladi. Ushbu ob'ektlar yoki harakatlarni amalga oshirishda tashabbus ko'rsatadi yoki ushbu harakatlarning ayrim natijalarini aniqlashda. Shu bilan birga, harakatlar faoliyat grafigining bir ob'ektidan boshqasiga o'tkaziladigan chaqiruvlarni tasniflaydi. Ob'ektlarni faoliyat diagrammasida (21.8- rasm) oshkora ko'rsatish mumkin.



21.8- rasm. Parallel yo'lli va ob'yektli faoliyat diagrammasi

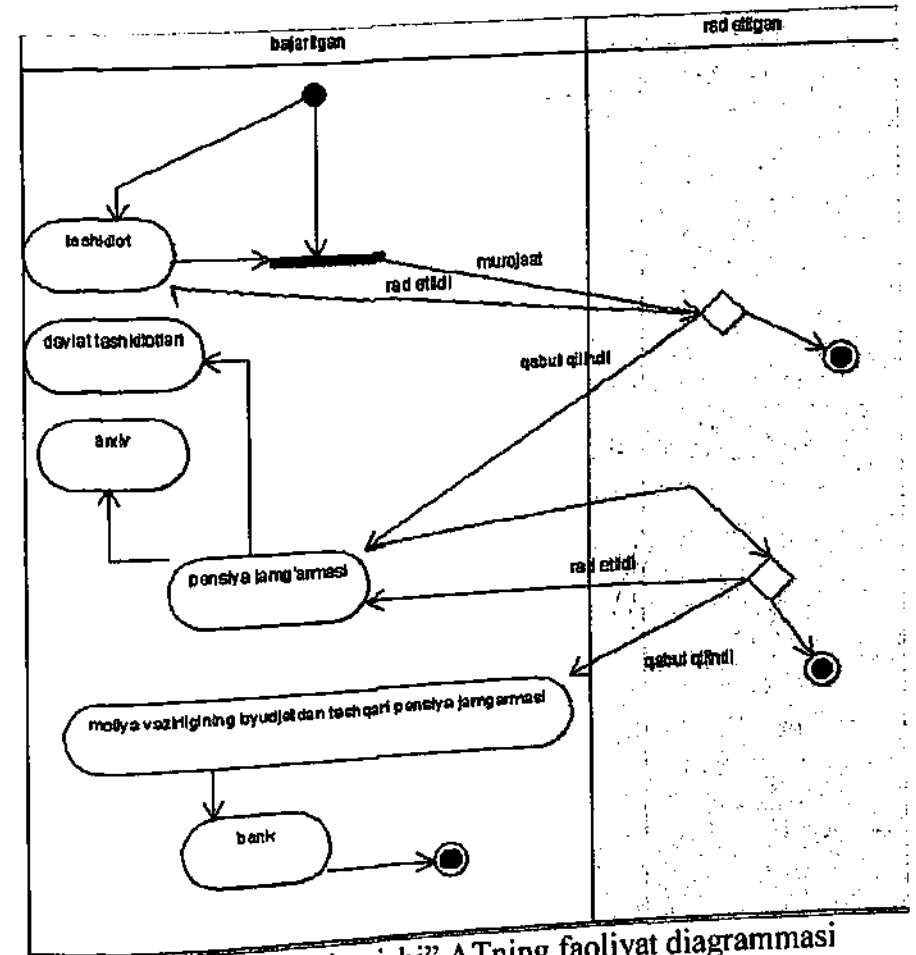
Ob'yektlarni grafik ko'rinishi uchun sinfni belgilashdagi to'rtburchak ishlatiladi, farqi bundagi ob'yektning nomi tag chiziq bilan ta'kidlanadi. Yo'llari mavjud bo'lgan faoliyat diagrammasida ob'yektning joylashuvi qo'shimcha ma'noga ega bo'lishi mumkin. Masalan, agar ob'yekt ikki yo'lning chegarasida joylashgan bo'lsa, demak, keyingi bosqichga o'tish keyingi yo'lda ba'zi tayyorgarlik bilan bog'liq hujjat (ob'yekt muayyan holatda). Agar ob'yekt butunlay yo'l ichida joylashgan bo'lsa, unda bu ob'yektning holati butunlay ushbu yo'lning harakati bilan belgilanadi.

### 21.3. "Pensiya ishi" ATning faoliyat diadnammasi

"Pensiya ishi" ATning faoliyat diagrammasida tizim tomonidan bajariladigan amallarni algoritmik va mantiqiy amalga oshirish xususiyatlari aks ettirilgan. ATni ishlab chiqishda tizimli yondashuv bilan algoritmik qarorlarni aks ettirish uchun an'anaviy ravishda algoritmning oqim chizmalari qo'llaniladi. Bunday holda faoliyat deganda, qo'lda yoki avtomatizatsiyalash vositalaridan foydalangan holda bajarilishi kerak bo'lgan amal tushuniladi. Har bir foydalanish holati o'z amallarining ketma-ketligiga ega. Nazariy jihatdan, faoliyat diagrammalari tahlil qilingan foydalanish holatini amalga oshiradigan algoritmning umumlashtirilgan ko'rinishidir. Faoliyat diagrammasi muqobil va parallel jarayonlarni tasvirlash imkonini beradi.

Faoliyat diagrammasining (21.9- rasm) algoritmi quyidagicha:

1. Fuqaro tashkilot orqali yoki to'g'ridan-to'g'ri o'zi hududidagi pensiya jamg'armasiga belgilangan romb orqali tushadi.
2. Rombga ko'ra, agarda murojaat qilgan fuqaroning hujjatlarini o'rganish jarayonida kamchiliklar topilsa, kam bo'lgan hujjatlarni pensiya jamg'armasi tashkilotga yoki fuqaroning o'ziga murojaat qilib belgilangan vaqt mobaynida taqdim etilishini talab qiladi.
3. Agarda belgilangan vaqt mobaynida kerakli hujjatlar tashkilot yoki fuqaro tomonidan taqdim etilmasa, ish jarayoni to'xtatiladi.
4. 1-3 qadamlar asosida pensiya jamg'armasi to'plangan hujjatlarni moliya vazirligining byudjetdan tashqari pensiya jamg'armasiga yuboradi.



21.9- rasm. "Pensiya ishi" ATning faoliyat diagrammasi

5. Agarda hujjatlarda kamchilk topilsa, moliya vazirligining byudjetdan tashqari pensiya jamg'armasi ham hudud pensiya jamg'armasiga belgilangan vaqt mobaynida hujjatlarni taqdim etilishini so'raydi.
6. Belgilangan vaqt mobaynida kerakli hujjatlar hudud pensiya jamg'armasi tomonidan taqdim etilmasa ish jarayoni to'xtatiladi.
7. Hamma hujjatlar to'liq va kamchiliksiz bo'lsa, moliya vazirligining byudjetdan tashqari pensiya jamg'armasi tayinlangan pensiyani bank hisobiga o'tkazadi va bank orqali fuqarolarga yetkaziladi.

## XXII- BOB. JOYLASHTIRISH DIAGDAMMASI

### 22.1. Nazariy asoslar

UML-dagi joylashtirish (yoyish) diagrammasi (Deployment diagram) tugunlarda artefaktlarning jismoniy joylashishini modelashtiradi. *Joylashtirish diagrammasi* – bu tizimni amalga oshirishda ishtirok etadigan asbob qurilmalari o'rasida ma'lumotlarni uzatish yo'nalishlaridagi jismoniy aloqalar mavjudligini ko'rsatadi. Joylashtirish diagrammasi faqat uni bajarish bosqichida mavjud bo'lgan dastur elementlari va tarkibiy qismlarini namoyish qilish uchun mo'ljallangan (runtime).

Masalan, web-saytni tavsiflash uchun joylashtirish diagrammasi qaysi asbobning tarkiblari ("tugunlar") mavjudligini – masalan, veb-server, BBning serveri, dasturlar, printerlar, modemlar, raqamli kameralar, skanerlar va manipulyatorlar; balki har bir tugundagi dasturiy tarkiblar ("artefaktlar") ishlashini (web-ilova, BB) va ushbu majmuaning turli qismlarining bir-biri bilan qanday bog'langanligini ko'rsatishi kerak.

Tugun – bu ba'zi bir hisoblash zaxiraloriga ega bo'lgan tizimning jismoniy mavjud elementi. Tugunlar artefaktli to'g'ri burchakli parallapiped sifatida ifodalanadi. Tugunlar qism-tugunlarga ega bo'lishi va ular ichma-ich joylangan to'g'ri burchakli parallapiped sifatida bo'lishi mumkin. Bitta joylashtirish diagrammasining tuguni g'oyaviy ravishda BB serverlari klasteri kabi, ko'plab jismoniy tugunlarni aks ettirishi mumkin.

Tugunlarning ikki turi mavjud:

- ❖ qurilma tuguni;

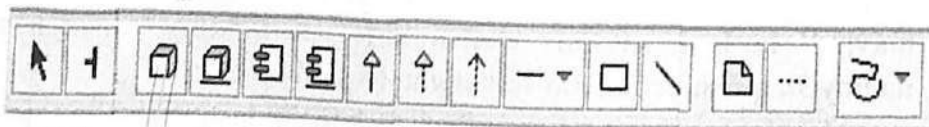
❖ ish vaqti muhiti tuguni.

*Qurilma tugunlari* – bu oddiy kompyuterlar, mobil telefonlar kabi dasturiy ta'minotni bajarish uchun xotira va xizmatlarga ega bo'lgan jismoniy hisoblash zaxiralaridan iborat.

*Ish vaqti muhiti tuguni* – bu tashqi tugun ichida ishlaydigan va boshqa bajariluvchi dastur elementlariga xizmat ko'rsatadigan dasturiy hisoblash zaxirasidan iborat.

## 22.2. Joylashtirish diagrammasining asboblari

Joylashtirish diagrammasini qurish uchun asboblari paneli 22.1-rasmda keltirilgan. Asboblarning tavsifi 22.1- jadvalda keltirilgan.



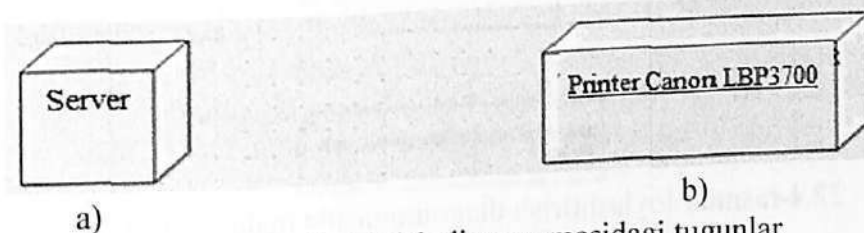
22.1-rasm. Joylashtirish diagrammasini qurish uchun asboblari

22.1- jadval. Joylashtirish diagrammasi asboblarning tavsifi

Asboblari	Tavsifi
	Tugun (tugun-tur)
	Tugun nusxasi
	Tarkib
	Tarkib-nusxasi
	Umumlashtirish (avloddan ajdodga)
	Amalga osirish (sinf dan interfeysga)

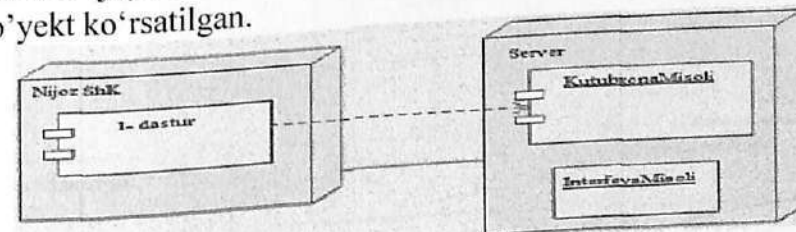
	Tobelik (tobe elementdan)
	Uyushma (uning 6ta turi 17- bobda keltirilgan)
	Ob'yekt
	Aloqa (bog'lanish)

Quyida joylashtirish diagrammasidagi tugun-tur (22.2- rasm a) va tugun nusxasiga (22.2- rasm b) misollar keltirilgan.





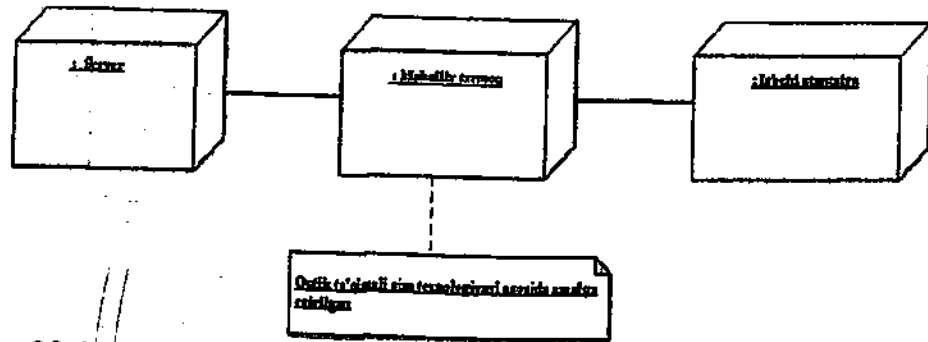
22.2- rasm. Joylashtirish diagrammasidagi tugunlar

Joylashtirish diagrammasidagi tugunlar ichida bajariladigan tarkiblar joylashishi mumkin. 22.3- rasmda joylashtirish diagrammasida tugunlar ichida bajariladigan tarkiblar joylashgani ko'rsatilgan. ArgoUMLda interfeysni ko'rsatish uchun asbob yo'qligi sababli, joylashtirish diagrammasida interfeysning nusxasi – ob'yekt ko'rsatilgan.



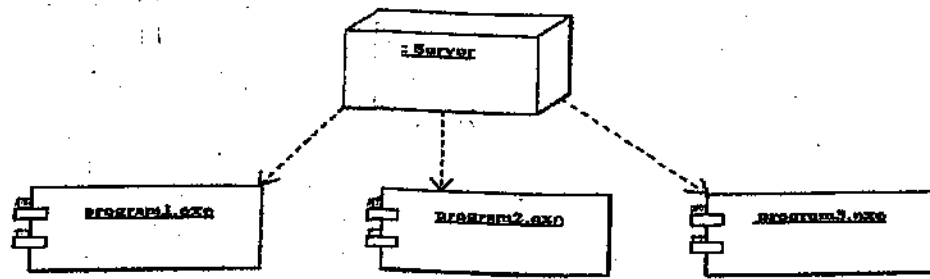
22.3- rasm. Joylashtirish diagrammasidagi tugunlar ichida bajariladigan tarkiblar

Joylashtirish diagrammalarida tugunlar orasidagi munosabatlar sifatida tugunlar va tarkiblar orasidagi jismoniy aloqalar va tobeliklarni ifodalash mumkin. 22.4- rasmda kompyuter tarmog'ini uchun joylashtirish diagrammasi ko'rsatilgan.  asbobi bilan sharh qo'shilgan, sharh bilan tugun orasidagi aloqa esa  asbobi yordamida qo'shilgan.



22.4-rasmda Joylashtirish diagrammasida mahalliy tarmoqning tasviri

Agar tugunda ko'p sonli tarkiblar joylashtirilgan bo'lsa, ularni tugun ichida emas, balki tobelik munosabatlari sifatida ko'rsatish mumkin (22.5- rasm).

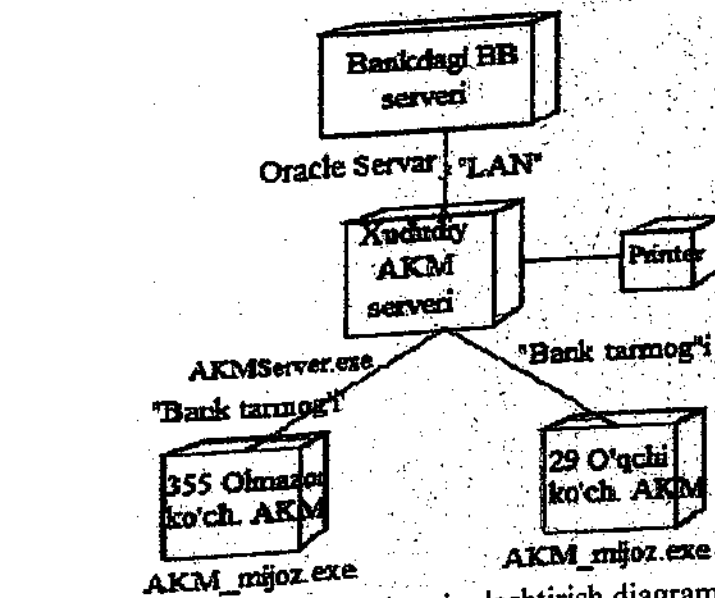


22.5- rasm. Joylashtirish diagrammasida tobelik munosabatlardan foydalanish

### 22.3. AKM tizimining joylashtirish diagrammasi

Ushbu diagrammadan tizimning jismoniy joylashuvi haqida bilib olishimiz mumkin. AKM mijoz dasturlari turli saytlarda bir nechta joylarda ishlaydi. Yopiq tarmoqlar orqali ular AKM hududiy serveri bilan aloqa o'rnatadilar. Unda AKM server dasturi ishlaydi. O'z navbatida, mahalliy tarmoq orqali hududiy server Oracle tomonidan boshqariladigan bank berilganlar bazasi serveri bilan aloqa o'rnatadi. Printer esa mintaqaviy AKM serveriga ulangan. Joylashtirish diagrammasining loyiha menejeri, foydalanuvchilar, tizim arxitektori va amaliyot xodimlari tomonidan tizimning va uning alohida qism-tizimlarining jismoniy joylashishini tushunish uchun ishlatiladi.

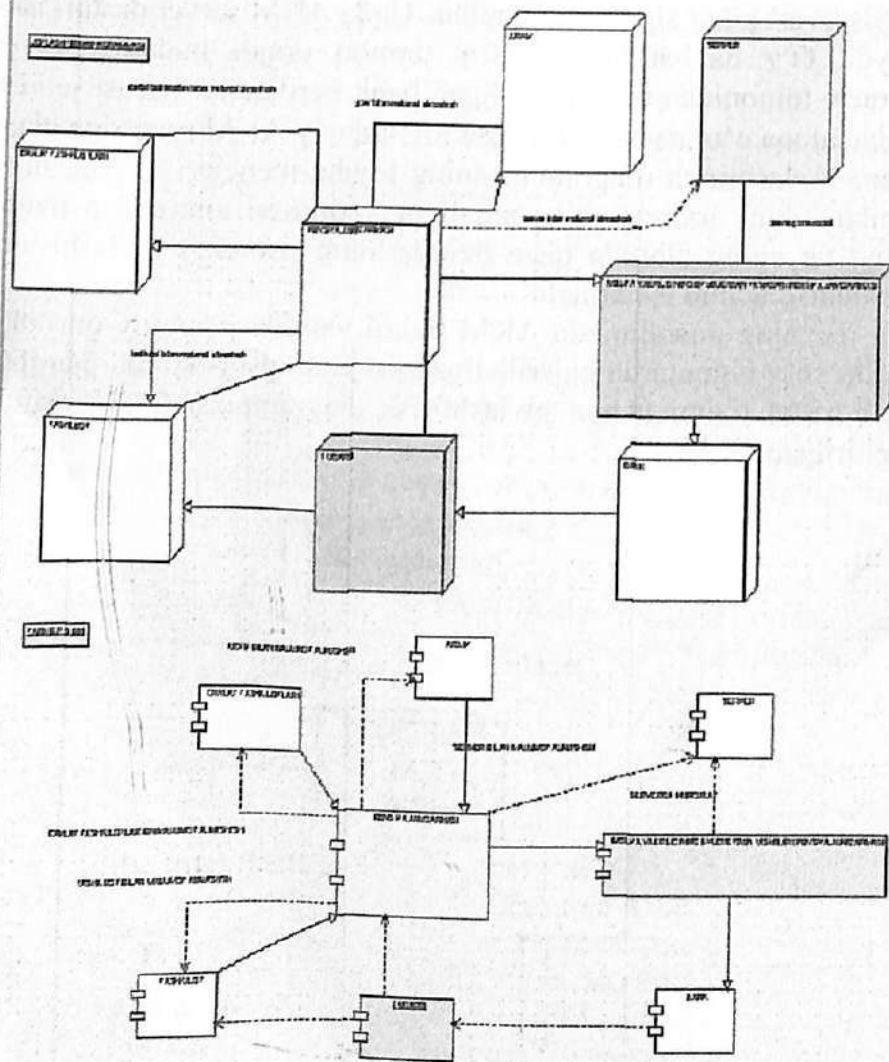
Bizning misolimizda AKM tizimi alohida jismoniy qurilmalarda yoki tugunlarda bajariladigan ko'plab qism-tizimlardan iborat. AKM tizimi uchun joylashtirish diagrammasi 23.6- rasmda keltirilgan.



22.6- rasm. AKM tizimi uchun joylashtirish diagrammasi

## 22.4. "Pensiya ishi" ATning joylashtirish diagrammasi

Joylashtirish diagrammasining (22.7- rasm) algoritmda fuqaro tashkilot yoki to'g'ridan-to'g'ri pensiya jamg'armasiga murojaat qilishi ko'rsatilgan:



22.7- rasm. "Pensiya ishi" ATning joylashtirish diagrammasi

1. Fuqarolar hudud pensiya jamg'armasiga pensiya tayinlash yoki pensiyani qayta hisoblash uchun murojaat qilishlari mumkin.
2. Agar fuqaro pensiyani tayinlash uchun murojaat qilgan bo'lsa, hujjatlarda kamchiliklar aniqlanmasa, moliya vazirligining byudjetdan tashqari pensiya jamg'armasi tomonidan tasdiqlanadi va pensiya tayinlanadi. Bunda fuqarolarga tayinlangan pensiyalar haqidagi BB serverga saqlab boriladi.
3. Agarda fuqaro tayinlangan pensiyasini mavjud asoslariga ko'ra qayta hisoblash uchun hudud pensiya jamg'armasiga murojaat qilishi mumkin bo'ladi.
4. Hudud pensiya jamg'armasi mavjud asoslar asosli deb topsa, u holda fuqaroning tayinlangan pensiyasi haqidagi ma'lumotlar serverga murojaat qilish orqali olishlari mumkin bo'ladi va fuqaroning pensiyasini mavjud asoslarga qayta hisob qilish uchun moliya vazirligining byudjetdan tashqari pensiya jamg'armasiga murojaat yo'llaniladi.
5. Moliya vazirligining byudjetdan tashqari pensiya jamg'armasida hujjatlar asosli deb topilsa, fuqaroning pensiyasi qayta hisob qilinib, serverdagi berilganlarni ham o'zgartirishlar kiritadi.

## XXIII- BOB. TARKIB DIAGRAMMASI

### 23.1. Nazariy asoslar

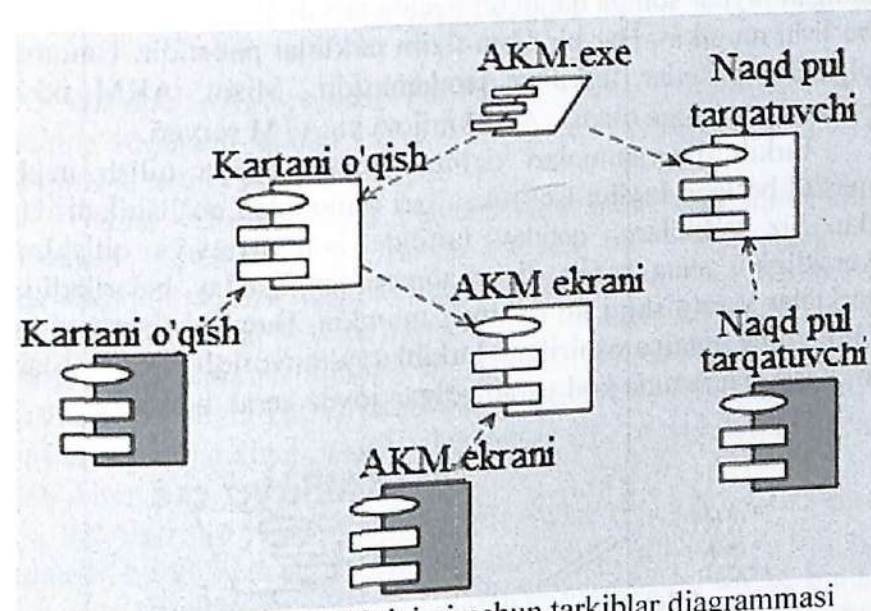
*Tarkib diagrammasida (component diagram)* modelning jismoniy darajadagi ko'rinishi ko'rsatiladi. Ularda dasturiy ta'minot tarkiblari va ular orasidagi aloqalar tasvirlanadi. Bunday holda tarkiblar diagrammasida tarkiblar ikki turga – bajariluvchi tarkiblar va kutubxonadagi kodlarga ajratiladi. Har bir model sinfi (yoki qism-tizim) boshlang'ich kodlardan iborat tarkiblarga aylantiriladi. Yaratilgandan so'ng, ular darhol tarkiblar diagrammasiga qo'shiladi. Alohida tarkiblar o'rtasida dasturni kompilyatsiya qilish yoki bajarish bosqichidagi tobeliklarga mos keladigan tobeliklar tasvirlanadi.

### 23.2. AKM tizimi uchun tarkib diagrammasi

23.1- rasmda AKM tizimi uchun tarkiblar diagrammalaridan biri ko'rsatilgan.

Ushbu diagrammada AKM tizimi mijozining tarkiblari ko'rsatilgan. Bunda tizim C++ tilida ishlab chiqilgan. Har bir sinfning o'z sarlavha fayli va .CPP kengaytmali fayli mavjud, shuning uchun har bir sinf diagrammada o'z tarkib qismlariga ega bo'ladi. Masalan, AKM ekran sinfi diagrammaning AKM ekran tarkibiga aylantiriladi. Shuningdek, u AKM Screen-ning ikkinchi tarkibiga aylantiriladi. Ushbu ikkita tarkib birgalikda AKM screen sinfining tanasi va sarlavhasini ifodalaydi. Ajratilgan qora rangli tarkib *paket tasnifi (package specification)* deb nomlanadi va C++ tilidagi AKM screen sinfining CPP kengaytmali fayliga mos

keladi. Ajratilmagan tarkib paket tasnifi deb ham ataladi, lekin C++ tili sinfidagi sarlavha fayliga mos keladi (.H kengaytmali fayl). AKM.exe tarkibi masalaning tasnifi bo'lib, axborotni qayta ishlash oqimini (thread of processing) ifodalaydi. Bunday holda ishlov berish oqimi bajariladigan dastur bo'lib hisoblanadi.



23.1- rasm. AKM tizimi uchun tarkiblar diagrammasi

Tarkiblar uzlukli chiziq bilan birlashtirilgan bo'lib, bu chiziq ular orasidagi tobelik mavjudligini bildiradi. Masalan, Card Reader sinfi AKM screen sinfiga tobe. Bu shuni anglatadiki, Card Reader sinfini kompilyatsiya qilish uchun AKM Screen sinfi avvalroq mavjud bo'lgan bo'lishi kerak.

Barcha sinflar kompilyatsiya qilingandan so'ng, AKMMijoz.exe bajariladigan fayl yaratilishi mumkin.

AKM misoli ikkita ishlov berish oqimini o'z ichiga oladi va shuning uchun ikkita bajariladigan fayl hosil bo'ladi. Ulardan biri ATM mijoz bo'lib, unda Cash Dispenser, Card Reader va AKM

## XXIII- BOB. TARKIB DIAGRAMMASI

### 23.1. Nazariy asoslar

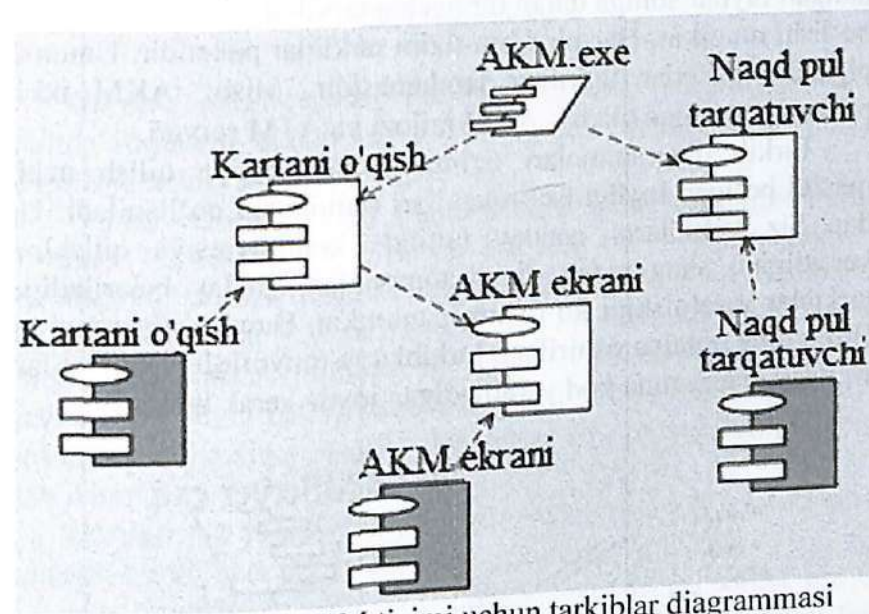
*Tarkib diagrammasida (component diagram)* modelning jismoniy darajadagi ko'rinishi ko'rsatiladi. Ularda dasturiy ta'minot tarkiblari va ular orasidagi aloqalar tasvirlanadi. Bunday holda tarkiblar diagrammasida tarkiblar ikki turga – bajariluvchi tarkiblar va kutubxonadagi kodlarga ajratiladi. Har bir model sinfi (yoki qism-tizim) boshlang'ich kodlardan iborat tarkiblarga aylantiriladi. Yaratilgandan so'ng, ular darhol tarkiblar diagrammasiga qo'shiladi. Alohida tarkiblar o'rtasida dasturni kompilyatsiya qilish yoki bajarish bosqichidagi tobeliklarga mos keladigan tobeliklar tasvirlanadi.

### 23.2. AKM tizimi uchun tarkib diagrammasi

23.1- rasmda AKM tizimi uchun tarkiblar diagrammalaridan biri ko'rsatilgan.

Ushbu diagrammada AKM tizimi mijozining tarkiblari ko'rsatilgan. Bunda tizim C++ tilida ishlab chiqilgan. Har bir sinfning o'z sarlavha fayli va .CPP kengaytmali fayli mavjud, shuning uchun har bir sinf diagrammada o'z tarkib qismlariga ega bo'ladi. Masalan, AKM ekran sinfi diagrammaning AKM ekran tarkibiga aylantiriladi. Shuningdek, u AKM Screen-ning ikkinchi tarkibiga aylantiriladi. Ushbu ikkita tarkib birgalikda AKM screen sinfining tanasi va sarlavhasini ifodalaydi. Ajratilgan qora rangli tarkib *paket tasnifi (package specification)* deb nomlanadi va C++ tilidagi AKM screen sinfining CPP kengaytmali fayliga mos

keladi. Ajratilmagan tarkib paket tasnifi deb ham ataladi, lekin C++ tili sinfidagi sarlavha fayliga mos keladi (.H kengaytmali fayl). AKM.exe tarkibi masalaning tasnifi bo'lib, axborotni qayta ishlash oqimini (thread of processing) ifodalaydi. Bunday holda ishlov berish oqimi bajariladigan dastur bo'lib hisoblanadi.



23.1- rasm. AKM tizimi uchun tarkiblar diagrammasi

Tarkiblar uzlukli chiziq bilan birlashtirilgan bo'lib, bu chiziq ular orasidagi tobelik mavjudligini bildiradi. Masalan, Card Reader sinfi AKM screen sinfiga tobe. Bu shuni anglatadiki, Card Reader sinfini kompilyatsiya qilish uchun AKM Screen sinfi avvalroq mavjud bo'lgan bo'lishi kerak.

Barcha sinflar kompilyatsiya qilingandan so'ng, AKMMijoz.exe bajariladigan fayl yaratilishi mumkin.

AKM misoli ikkita ishlov berish oqimini o'z ichiga oladi va shuning uchun ikkita bajariladigan fayl hosil bo'ladi. Ulardan biri ATM mijoz bo'lib, unda Cash Dispenser, Card Reader va AKM

## XXIII- BOB. TARKIB DIAGRAMMASI

### 23.1. Nazariy asoslar

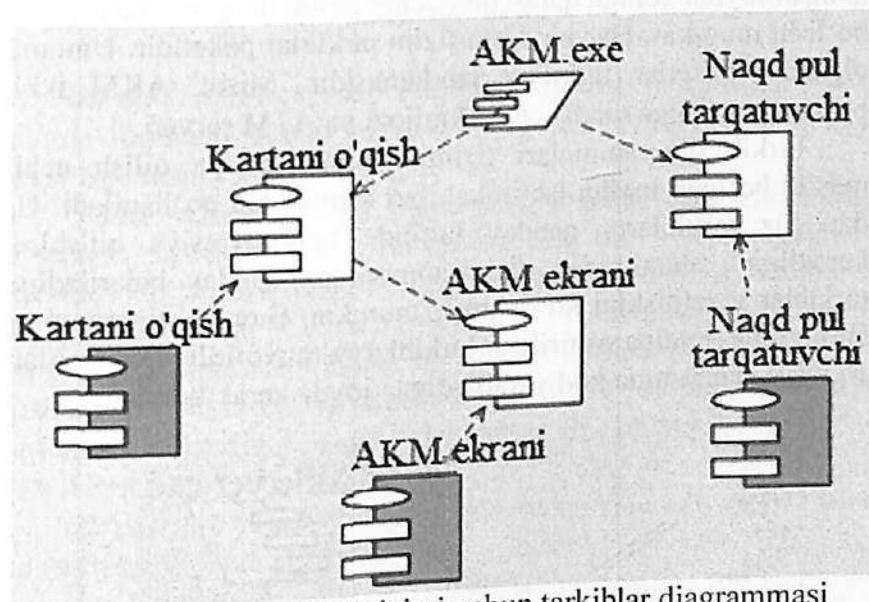
*Tarkib diagrammasida (component diagram)* modelning jismoniy darajadagi ko'rinishi ko'rsatiladi. Ularda dasturiy ta'minot tarkiblari va ular orasidagi aloqalar tasvirlanadi. Bunday holda tarkiblar diagrammasida tarkiblar ikki turga – bajariluvchi tarkiblar va kutubxonadagi kodlarga ajratiladi. Har bir model sinfi (yoki qism-tizim) boshlang'ich kodlardan iborat tarkiblarga aylantiriladi. Yaratilgandan so'ng, ular darhol tarkiblar diagrammasiga qo'shiladi. Alohida tarkiblar o'rtasida dasturni kompilyatsiya qilish yoki bajarish bosqichidagi tobeliklarga mos keladigan tobeliklar tasvirlanadi.

### 23.2. AKM tizimi uchun tarkib diagrammasi

23.1- rasmda AKM tizimi uchun tarkiblar diagrammalaridan biri ko'rsatilgan.

Ushbu diagrammada AKM tizimi mijozining tarkiblari ko'rsatilgan. Bunda tizim C++ tilida ishlab chiqilgan. Har bir sinfning o'z sarlavha fayli va .CPP kengaytmali fayli mavjud, shuning uchun har bir sinf diagrammada o'z tarkib qismlariga ega bo'ladi. Masalan, AKM ekran sinfi diagrammaning AKM ekran tarkibiga aylantiriladi. Shuningdek, u AKM Screen-ning ikkinchi tarkibiga aylantiriladi. Ushbu ikkita tarkib birgalikda AKM screen sinfining tanasi va sarlavhasini ifodalaydi. Ajratilgan qora rangli tarkib *paket tasnifi (package specification)* deb nomlanadi va C++ tilidagi AKM screen sinfining CPP kengaytmali fayliga mos

keladi. Ajratilmagan tarkib paket tasnifi deb ham ataladi, lekin C++ tili sinfidagi sarlavha fayliga mos keladi (.H kengaytmali fayl). AKM.exe tarkibi masalaning tasnifi bo'lib, axborotni qayta ishlash oqimini (thread of processing) ifodalaydi. Bunday holda ishlov berish oqimi bajariladigan dastur bo'lib hisoblanadi.



23.1- rasm. AKM tizimi uchun tarkiblar diagrammasi

Tarkiblar uzlukli chiziq bilan birlashtirilgan bo'lib, bu chiziq ular orasidagi tobelik mavjudligini bildiradi. Masalan, Card Reader sinfi AKM screen sinfiga tobe. Bu shuni anglatadiki, Card Reader sinfini kompilyatsiya qilish uchun AKM Screen sinfi avvalroq mavjud bo'lgan bo'lishi kerak.

Barcha sinflar kompilyatsiya qilingandan so'ng, AKMMijoz.exe bajariladigan fayl yaratilishi mumkin.

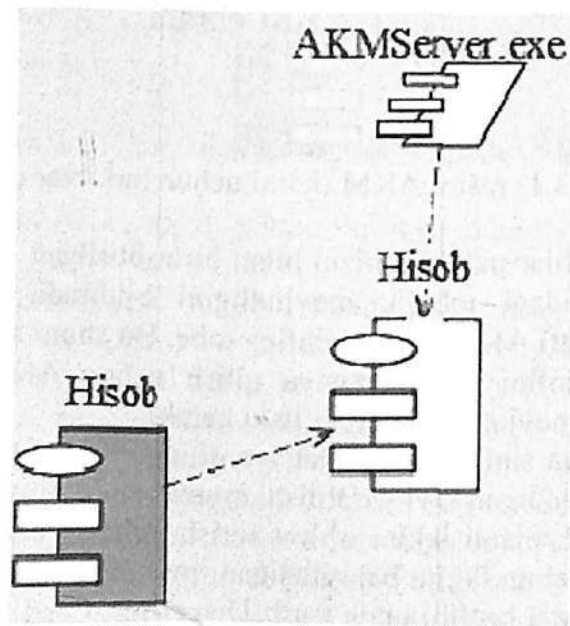
AKM misoli ikkita ishlov berish oqimini o'z ichiga oladi va shuning uchun ikkita bajariladigan fayl hosil bo'ladi. Ulardan biri ATM mijozi bo'lib, unda Cash Dispenser, Card Reader va AKM

Screen tarkiblar bo'ladi. Ikkinchi fayl – bu Account tarkibini o'z ichiga olgan AKM serveri.

ATM serveri uchun tarkiblar diagrammasi 23.2- rasmda keltirilgan.

Misoldan ko'rinib turibdiki, tizim – qism-tizimlar yoki bajariladigan fayllar soniga qarab bir nechta tarkib diagrammalariga ega bo'lishi mumkin. Har bir qism-tizim tarkiblar paketidir. Umuman olganda, paketlar tarkiblar jamlamasidir. Misol, AKM ikkita paketni o'z ichiga oladi – AKM mijozi va ATM serveri.

Tarkib diagrammalari tizimni kompilyatsiya qilish uchun mas'ul bo'lgan loyiha ishtirokchilari tomonidan qo'llaniladi. Undan biz tarkiblarni qanday tartibda kompilyatsiya qilishimiz kerakligini, shuningdek, tizim tomonidan qanday bajariladigan tarkiblar yaratilishini ko'rishimiz mumkin. Bunday diagrammada sinflarning amalga oshirilgan tarkiblarga muvofiqligi ko'rsatiladi. Bunday diagramma kod yaratiladigan joyda kerak bo'ladi.



23.2- rasm. AKM server uchun tarkiblar diagrammasi

## XXIV- BOB. OB'YEKT DIAGRAMMASI

### 24.1. Ob'yekt

*Ob'yekt diagrammasi (object diagram)* belgilangan aniq vaqtda momenti uchun ob'yektlar va ular orasidagi aloqalarni tavsiflash uchun mo'ljallangan.

Ushbu turdagi kanonik diagrammalar aloqa diagrammalariga o'xshash bo'lsa-da, ular o'rasida tub farq bor – aloqa diagrammalarida ob'yektlar diagrammalarida mavjud bo'lmagan xabarlar mavjud. Garchi ob'yekt va hayot chizig'i tushunchari orasida juda ko'p o'xshashliklar mavjud bo'lsa-da, UML 2.0 dagi ob'yekt tushunchasi hayot chizig'idan farq qiladi. Ob'yektlar diagrammalarida tasvirlanadigan asosiy elementlar shu ob'yektlarning o'zidan iborat.

*Ob'yekt (ob'yekt)* – bu modelni amalga oshirish yoki dasturni bajarish bosqichida yaratilgan sinfning alohida nusxasi.

Har bir ob'yekt aniq bir xususiylikka ega, bu uning bir xil sinfdagi boshqa mumkin bo'lgan ob'yektlarning nomlaridan farq qiladigan noyob nomga ega bo'lishini anglatadi. O'zining noyob nomidan tashqari, ob'yekt o'ziga xos atributlar va ularning qiymatlariga ega bo'lishi mumkin. Ob'yektning grafik belgisi sifatida sinflovchi belgisi ishlatiladi, uning ichida ob'yekt nomi va atributlarining qiymatlari yozilishi mumkin.

Ob'yekt nomi quyidagi Bekus-Naur (BN) shaklidagi matn qatoridir:

$$\langle ob'yekt\ nomi \rangle ::= [ \langle ob'yektning\ shaxsiy\ nomi \rangle ] | [ : \langle sinf\ nomi \rangle [ ' , ' \langle sinf\ nomi \rangle ] * ]$$

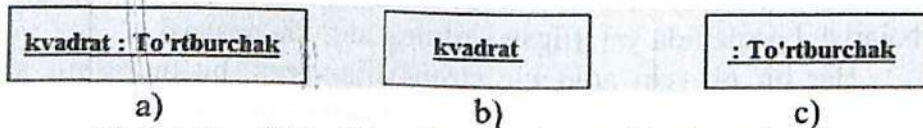
Ob'yekt nomi yozuvida  $\langle ob'yektning\ shaxsiy\ nomi \rangle$  va  $\langle sinf\ nomi \rangle$  bir vaqtning o'zida yo'zilmaligi mumkin emas va ob'yekt nomi tag chiziq bilan ta'kidlanadi.

Agar <ob'yektning shaxsiy nomi> ko'rsatilgan bo'lsa, u kichik harf bilan boshlanishi kerak. Agarda <sinf nomi> mavjud bo'lsa, undan oldin har doim ikki nuqta qo'yilishi shart.

Ob'yekt diagrammalarida ob'yekt nomlari quyidagi variantlarda yozilishi mumkin:

- **o: C** – ob'yekt uchun ob'yektning shaxsiy nomi va sinf nomi ko'rsatilgan;
- **o** – ob'yekt uchun faqat ob'yektning shaxsiy nomi ko'rsatilgan;
- **: C** – ob'yekt uchun faqat sinf nomi ko'rsatilgan.

Ob'yektlarning bir nechta aniq tasvirlarining misollarini ko'rib chiqamiz (24.1- rasm). Ob'yekt nomining to'liq tasnifi uning nomining ikkala elementini o'z ichiga oladi (24.1- rasm, *a*). Agar sinf nomi ko'satilmasa, unda bunday ob'yekt *yetim* deyiladi. Bunda faqat ob'yektning shaxsiy nomi yoziladi (24.1- rasm, *b*). Agar ob'yektning shaxsiy nomi bo'lmasa, bunday ob'yekt *anonym* deb nomlanadi va tegishli sinf nomidan oldin ikki nuqta qo'yiladi (24.1- rasm, *c*).



24.1- rasm. Ob'yektlarning grafik tasvirlariga misollar

Nusxa tasnifi sinf va ob'yekt o'rasidagi oraliq holatdir.

## 24.2. Nusxa tasnifi

**Nusxa tasnifi (instance specification)** – bu sinflovchi turidagi ixtiyoriy mohiyatning tavsifini ifodalovchi model elementidir.

Nusxa tasnifining maqsadi modellashtiriladigan tizimdagi qaysidir mohiyatning qiziqarli ekanligini ko'rsatishdir. Nusxa tasnifi modellashtiriladigan tizimdagi mohiyatning mavjudligini ko'rsatadi va uning to'liq yoki qisman tavsiflaydi. Bunday tavsif

quyidagilarni o'z ichiga olishi mumkin:

- ❖ shu mohiyatni mohiyati nusxadan iborat bo'lgan bir yoki bir nechta sinflovchilar orqali tasniflash. Agar yagona tasniflovchi sinflovchi mavhum bo'lsa, nusxa tasnifi shu mohiyatni qisman tavsiflaydi;
- ❖ sinflovchi yoki sinflovchilarga asoslangan nusxa turi. Masalan, sinflovchisi sinf bo'lgan nusxa tasnifi shu sinfning qaysidir ob'yektlarni tavsiflaydi, sinflovchisi uyushma bo'lgan nusxa tasnifi esa shu uyushmaning aloqasini tavsiflaydi;
- ❖ mohiyatning tarkibiy xususiyatlaridagi qiymatlarning tasnifi. Nusxa tasnifi sinflovchilarining barcha tarkibiy xususiyatlari uyalar bilan ifodalanishi mumkin emas, ba'zi hollarda nusxa tasnifi tavsifning qismi bo'lishi mumkin;
- ❖ nusxani qanday hisoblash, olish yoki yaratish bo'yicha, majburiy bo'magan tasnif.

UML 2.0 da nusxa tasnifi aniq bir sinfdir. Shuning uchun nusxa tasnifi va ob'yektning farqlash kerak, garchi ularning grafik tasvirlari deyarli bir xil bo'lsa ham.

Nusxa tasnifi modellashtiriladigan tizimda mumkin bo'lgan mohiyat misolini yoki namunasini ko'rsatishi mumkin. Nusxa tasnifining tegishli tafsilotlari to'liq bo'lmasligi mumkin. Nusxa tasnifi uning sinovchisi kabi bir xil belgi yordamida tasvirlanadi. Nusxa tasnifi nomi quyidagi BN shaklida yozilgan matn qatoridir:

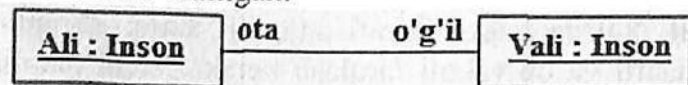
**< nusxa tasnifining nomi > ::= [**<nusxa tasnifining shaxsiy nomi >**] | [**< sinf nomi >**] | [**' <sinf nomi >'**] \***

Nusxa tasnifining nomidagi hamma ozuvning tagi chiziladi. Agarda <nusxa tasnifining shaxsiy nomi> ko'rsatilgan bo'lsa, yozma harf bilan boshlanishi kerak. Sinflovchilarning bir nechta nomlari ko'rsatilsa, ular vergul bilan ajratiladi. Nusxa tasnifining nomi yozuvda <nusxa tasnifining shaxsiy nomi > va <sinf nomi > bir vaqtning o'zida mavjud bo'lmasligi mumkin. Nusxa tasnifining nomi yo'qligi, uning asosiy modelda yo'qligini aks ettirishi

mumkin. Noma'lum sinflovchining anonim nusxasini tasniflash uchun standart belgi – bu tag chiziqli ikki nuqta (:) belgisi.

Sinflovchisi uyushma bo'lgan nusxaning tasnifi, aloqa shaklida taqdim etiladi va uyushma kabi bir xil yozuv yordamida tasvirlanadi. Bunday holda uzluksiz chiziq bilan sinflovchilar emas, balki nusxa tasniflari birlashtiriladi. Agar nusxa tasnifining ulanishi aniq bo'lsa, tagiga chizilgan aloqa nomi ko'rsatilmasligi mumkin. Bundan tashqari, shu aloqa uchlarning nomlari va yo'nalish millari qo'shimcha ravishda ko'rsatilishi mumkin. Ammo agar ular ko'rsatiladigan bo'lsa, ular mos uyushmaning yo'nalish uchlari bilan kelishishi kerak.

Misol tariqasida, nusxa tasniflari o'rasidagi Inson sinfidagi Ali va Vali ikkilik munosabatni ko'rib chiqamiz (24.2- rasm). Ular uyushma chizig'i shaklida nomsiz ikkilik aloqa bilan bog'langan. Ushbu bog'lanishning uchlari uchun tegishli nusxa tasnifining rol nomlari ko'rsatilgan.



24.2- rasm. Nusxa tasnifining aloqasi bilan bog'langan 2 ta ob'yekt

Nusxa tasniflari tegishli sinflovchilarning ob'yektlarining bajarilish vaqtining alohida xususiyatlarini tavsiflamaydi. Ular faqat bunday ob'yektlar haqida umumiy ma'lumotlarni ko'rsatadi. Shu bilan birga, sinflarni amalga oshirishdagi ish vaqti tafsilotlari to'g'risida hech qanday xulosa chiqarish mumkin emas. Bunday holda nusxa tasniflari to'liq bo'lmasligi mumkin, sinflovchilarning ayrim xususiyatlari o'tkazib yuborilishi mumkin. Haqiqiy mohiyatlar aniq sinflovchilarga ega bo'lishi kerak bo'lsa ham, nusxa tasniflarining sinflovchlari mavhum bo'lishi mumkin.

### 24.3. Uya

*Uya (slot)* nusxa tasnifi orqali modellashtirilgan ob'yekt o'zi-

ning ba'zi tarkibiy xususiyatlari uchun o'ziga xos qiymat yoki qiymatlarga ega ekanligini ko'rsatish uchun mo'ljallangan. Uya bizga nusxa tasnifi, tuzilmaviy xususiyatlar va bir yoki bir nechta qiymatlarni bog'lash imkonini beradi. Uya nusxa tasnifining mulki bo'lib, ushbu uyaga egalik qiluvchi nusxa tasnifining alohida tuzilmaviy xususiyatlari uchun bir yoki bir nechta qiymatlarni belgilaydi. Bunday holda uyadagi qiymatlar ushbu uyaning xususiyatlariga mos kelishi kerak (turi, karraligi va boshqalar bo'yicha). Nusxa tasnifi o'z sinflovchilarining alohida tuzilmaviy xususiyatlari, shu jumladan meros xususiyatlari uchun bitta uyaga ega bo'lishi mumkin. Har bir meros qilib olingan UML 2.0 tuzilmaviy xususiyati uchun uyadan foydalanishning hojati yo'q. Agar tuzilmaviy xususiyat sinfda tavsiflangan bo'lsa, unda u ushbu uyaga joylashtirilishi mumkin bo'lgan qiymatlarni aniqlaydi.

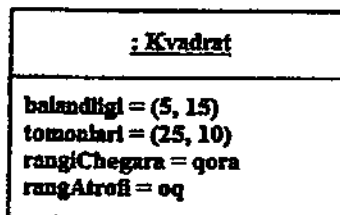
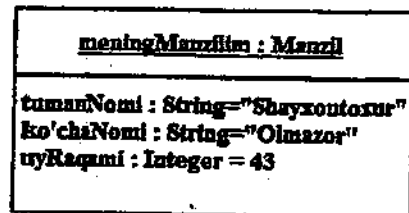
Uyalar tuzilmaviy xususiyatlar bilan mos keladigan yozuv yordamida tasvirlanadi. Agar xususiyat sinf atributlari bo'limida matn shaklida tasvirlangan bo'lsa, unda bu xususiyat uchun uya ham BN shaklida quyidagi formatda ko'rsatilishi mumkin:

`< slot> ::= < xususiyat nomi> [: < xususiyat turi>] '='`

`< tasnif-qiymatlar>`

Masalan, nusxa tasnifi meningManzilim (24.3- rasm, a) quyidagi uyalarga ega: tumanNomi (turi – string, qiymati “Shayxontoxur”), ko'chaNomi (turi – string, qiymati “Olmazor”), raqamUy (turi – integer, qiymati 43). Anonim sinf nusxasi tasnifi Kvadrat (24.3- rasm, b) ularning turini ko'rsatilmagan 4 ta uyaga ega. Bunday holda har bir tuzilmaviy xususiyatning turini tegishli sinfda tavsiflash mumkin.

Har bir uyani belgilaydigan xususiyat, ko'rib chiqilayotgan tasnif nusxasi uchun sinflovchining to'g'ridan-to'g'ri yoki meros qilib olingan tuzilmaviy xususiyati bo'lishi mumkin. Bunda bitta tarkibiy xususiyat, shu jumladan bir nechta sinflovchilardan meros bo'lib o'tgan bir xil xususiyat, tasnif nusxasidagi bitta uyadan



24.3- rasm. Qiymatli uyalarga misollar: a) tasnif nusxasi *meningManzili*; b) *Kvadrat* sinfi nusxasining anonim tasnifi

ko'p bo'lmagan xususiyatdan iborat bo'lishi mumkin.

Tasnif nusxasidagi ba'zi bir xarakteristikalar uchun uyaning yo'qligi, taqdim etilgan ob'yekt ushbu xususiyatga ega emasligini anglatmaydi, balki faqat shu xususiyat modelda qiziqish uyg'otmasligini bildiradi. Tasnif nusxasi sinf bo'lganligi sababli, u shu nusxa uchun aniq qiymatlarga ega bo'lishi mumkin.

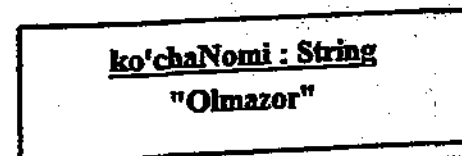
#### 24.4. Nusxa qiymati

*Nusxa qiymati (instance value)* – bu ayrim nusxani aniqlaydigan qiymat tasnidir. Qiymat tasnifi – bu ba'zi bir sinflovchining nusxasini qanday hisoblash, olish yoki yaratish bo'yicha tasnidir. Bunda, nusxa qiymati nusxa tasnifi orqali modellashtiriladigan qiymatni tasniflaydi.

Nusxa qiymati matn yoki grafik yozuv yordamida ifodalaniishi mumkin. Matnli belgilash atribut uyasi qiymatini ko'rsatish uchun ishlatiladi, bunda tegishli nusxaning nomi ko'rsatiladi. Grafik belgilash nusxaning qiymatini mos yozuvlar shaklida ifodalash va uni tegishli nusxa bilan bog'lash uchun xizmat qiladi.

Agarda nusxa tasnifi qiymat tasnididan foydalansa, u nusxa tasnidining nomi va tenglik belgisidan keyin yoki nom ostida tenglik belgisiz tasvirlanishi mumkin. Agar nusxa tasnifi to'rtburchaklar sinf shaklida tasvirlangan bo'lsa, u holda qiymat tasnifi ushbu to'rtburchakda nom qismida va uning ostida ko'rsatilishi mumkin (24.4- rasm).

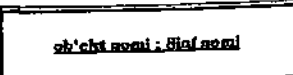
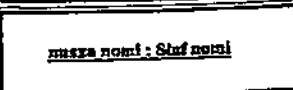

Ob'yektlar diagrammasining tavsifi oxirida ob'yektlar diagrammalarini tuzishda ishlatilishi mumkin bo'lgan barcha ko'rib chiqilgan grafik belgilarning qisqacha mazmuni keltirilgan (24.1- jadval).



24.4- rasm. *ko'chaNomi* nusxasining tasnifi

Ob'yekt diagrammalarida tasvirlangan grafik tugunlar va yo'llar boshqa tuzilish diagrammalarida ham bo'lishi mumkin.

24.1- jadval. Ob'yektlar diagrammasidagi belgilar

Asbob	Grafik element nomi
	Ob'yekt
	Nusxa tasnifi
	Aloqa

## XXV- BOB. PAKET DIAGRAMMASI

**Paket diagrammasi (package diagram)** – model elementlarini paketlarga joylashtirishni va paketlar va ularning elementlari o'rasidagi tobelik tasnifini aks ettirish uchun mo'ljallangan.

### 25.1. Paket

**Paket** – modelning boshqa elementlarini guruhlash uchun ishlatiladigan model elementi.

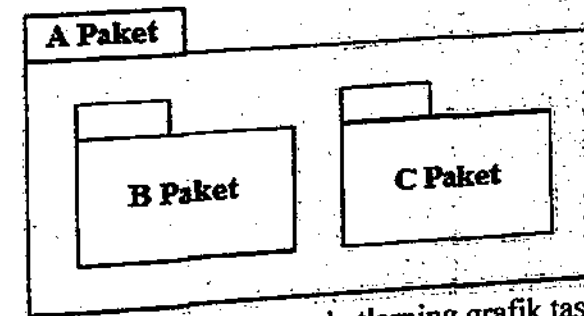
Ba'zi paketlarning bir qismi bo'lgan model elementlari ushbu **paket a'zolari**, deb ataladi. Paket barcha a'zolariga egalik qiladi. O'z navbatida, paket a'zolari haqida ular paketga tegishli yoki tobe, deb aytiladi. Agar biron bir paket modeldan olib tashlansa, u holda ushbu paketga tobelarning barchasi modeldan olib tashlanadi.

Ba'zi paketlarning nom maydoniga kiradigan model elementlari ushbu **paket elementlari**, deb ataladi. Ular orasida ushbu paketning a'zolari ham, boshqa import qilingan paketlarning a'zolari ham bo'lishi mumkin. Paket diagrammalarida ba'zi paketlar boshqa paketlarga joylashtirilishi mumkin. Bunday holda, joylangan paket **qism-paket**, deb ataladi va qism-paketning barcha elementlari, shuningdek, ko'rib chiqilayotgan qism-paket ham joylangan paket bo'lgan har qanday paketga tegishli bo'ladi. Shunday qilib, model elementlari uchun qat'iy shajara bo'lgan paketlarning joylanish nisbati belgilanishi mumkin. Yuqorida aytib o'tilganidek, paket katta to'rtburchaklar shaklida tasvirlangan, uning yuqori chap tomoniga kichik to'rtburchak biriktirilgan (5.1- jadvalga qarang).

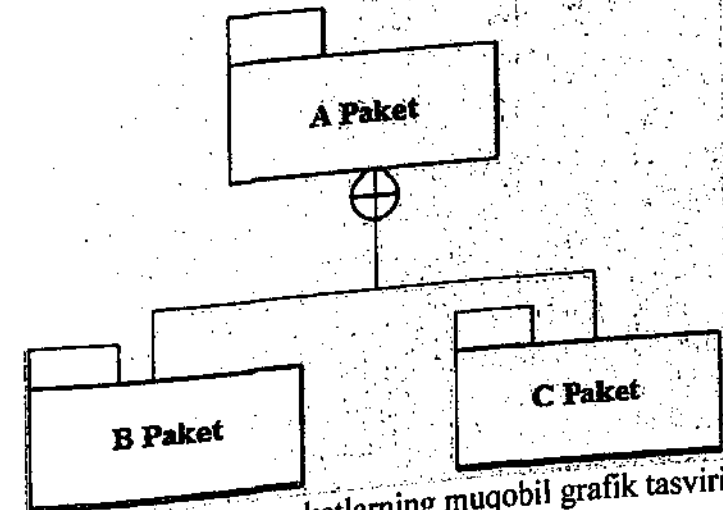
Paket nomi oldida qat'iy tur shaklida kalit so'zni o'z ichiga olgan matn qatori bo'lishi mumkin, masalan: <<facade>>, <<fra-

mework>>, <<model>>, <<stub>> va << topLevel>>.

Paketlarning tarkibi diagrammalarda ikki usulda tasvirlanishi mumkin. Birinchi usul katta to'rtburchakda paket a'zolarining tasviri bilan bog'liq. Bunday holda, paket nomi yuqori yorliqda joylashgan (25.1- rasm). Shu bilan bir qatorda, paket a'zolari paketning tashqi tomonida tasvirlanishi va paketga biriktirilgan uchidagi aylanada ortiqcha (+) belgisi bo'lgan chiziq segmenti bilan paketga ulanishi mumkin. Bunday holda, paket nomi katta to'rtburchakda joylashgan (25.2- rasm).

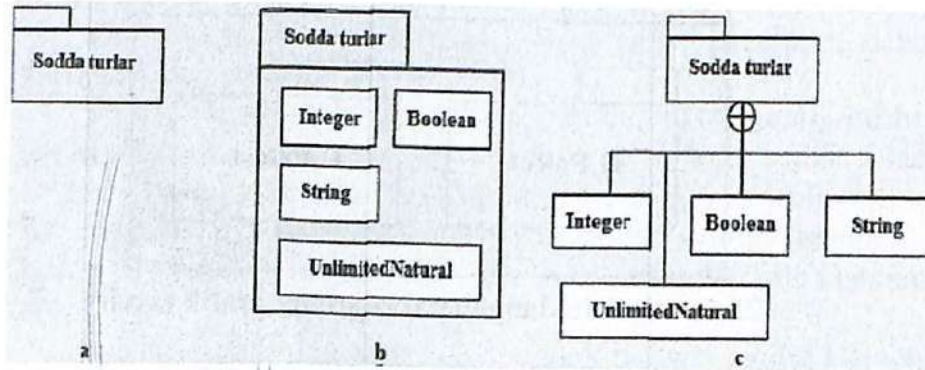


25.1- rasm. Joylangan paketlarning grafik tasviri



25.2- rasm. Joylangan paketlarning muqobil grafik tasviri

Misol tariqasida, Oddiy Turlar paketini taqdim etishning bir nechta variantlarini ko'rib chiqamiz (25.3- rasm). Paket diagrammasida barcha tarkibli paketlarni tasvirlash shart emasligi sababli, ushbu paket o'z a'zolarisiz ko'rsatilishi mumkin (25.3- rasm, *a*). UML 2.0 dagi eng oddiy turlar to'plami quyidagi turlarni o'z ichiga olganligi tasvirlangan: Integer, Boolean, String va Unlimited-Natural (25.3- rasm, *b*). Buni ikkita usuldan biri bilan tasavvur qilish mumkin – paket belgisi ichida ko'rsatilgan turlarni tasvirlash orqali (25.3- rasm, *b*) yoki paket belgisining tashqi tomonidagi turlarni tasvirlash (25.3- rasm, *c*).

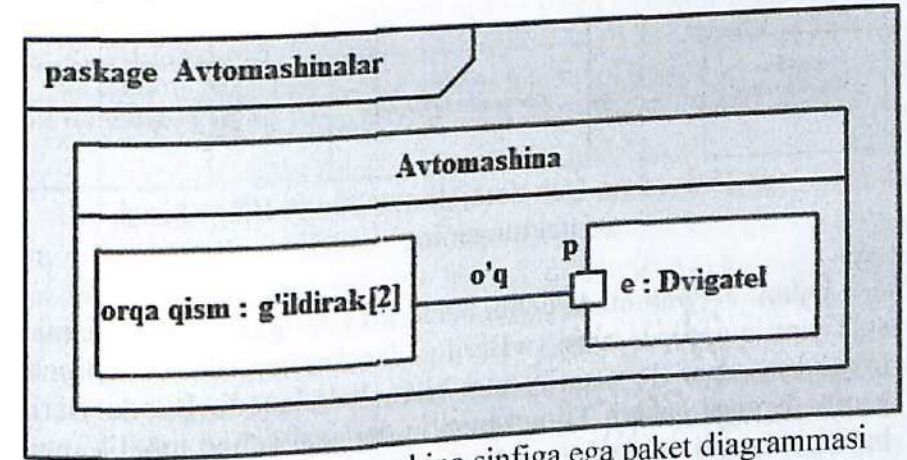


25.3- rasm. Oddiy turlar paketining variantlari

**Paket** – bu uning barcha elementlari uchun nom maydoni. Bu shuni anglatadiki, paket elementlariga malakasiz nomlar yordamida murojaat qilish mumkin, ya'ni ularni o'z ichiga olgan paket nomini ko'rsatmaslik mumkin. Paketga kiritilgan har qanday element ko'rinish chegarasiga ega bo'lishi mumkin. Umuman olganda, u umumiy (public) yoki xususiy (private) bo'lishi mumkin. Umumiy ko'rinishga ega bo'lgan paket elementlari har doim malakali nomlardan foydalanish orqali paketdan tashqarida mavjud, ya'ni ularni o'z ichiga olgan paket nomi ko'rsatilgan holda. Paket elementining ko'rinishi element nomidan oldingi ko'rinish belgisi yordamida ko'rsatilishi mumkin (eslatamiz: + umumiy uchun va – xususiy (yopiq) uchun). Paketga tegishli

a'zolaridan tashqari, biz import qilingan elementlar yoki ko'rib chiqilayotgan paket joylashtirilgan paket elementlari uchun malakasiz nomlardan foydalanishimiz mumkin. Bunda o'zi yoki import qilinadigan elementlarning har biri o'z ko'rinish chegarasiga ega bo'lishi mumkin, bu esa ushbu elementlarning o'z ichiga olgan paketdan tashqarida mavjudligini aniqlaydi.

Paket diagrammasida nafaqat paketlarni, balki alohida sinflarni ham tasvirlashga ruxsat beriladi, bu esa o'z navbatida o'z tuzilishiga ega bo'lishi mumkin. Shunday qilib, masalan, Avtomashinalar paketining diagrammasi Avtomashinalar sinfini o'z ichiga olishi mumkin, bu esa kompozit tuzilmaning qo'shimcha diagrammasiga ega (25.4- rasm).



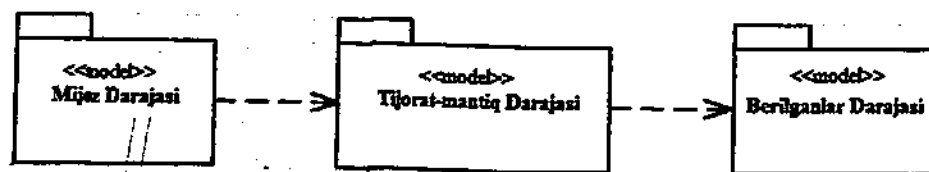
25.4- rasm. Avtomashina sinfiga ega paket diagrammasi

Paketlar orasidagi eng keng tarqalgan munosabat – bu diagrammada paketlarning turli xil ma'noli munosabatlarini aks ettirishga imkon beradigan tobelik munosabatidir.

## 25.2. Paketlardagi tobelik

Tobelik munosabati, umuman olganda, modelning ikkita elementi yoki bunday elementlarning ikkita to'plami o'rasidagi maz-

munli o'zaro aloqalarga mos kelishi mumkin. Tobelik munosabati modelning bir elementini o'zgartirish unga bog'liq bo'lgan boshqa elementni o'zgartirishni talab qilishi mumkin bo'lgan vaziyatni tavsiflash uchun ishlatiladi. Eslatib o'tamiz, tobelik munosabati grafik jihatdan mos keladigan elementlar orasidagi uzlukli chiziq bilan, uning uchlaridan biri esa ochiq mil bilan tasvirlanadi. Paket diagrammasida tobelik munosabati ikkita alohida paketni bog'lashi mumkin, mil har doim qaramlik mijoz paketidan (tobe paket) manba paketiga (mustaqil paket) yo'naltirilishi kerak. Masalan, masofadagi BBga kirishning uch darajali arxitekturasi umumiy modeli 25.5- rasmda keltirilgan.



25.5- rasm. Uch darajali masofaviy BBga kirish arxitekturasi modeli

Ushbu diagrammada uchta paket ko'rsatilgan – Mijoz darajasi, Tijorat-mantiq darajasi va Berilganlar darajasi, ular qo'shimcha ravishda <<model>> qat'iy turi bilan belgilanadi. Bunda Berilganlar darajasi paketi, Tijorat-mantiq darajasi uchun tobelik manbai hisoblanadi, u esa Mijoz darajasi paketi uchun tobelik manbai hisoblanadi.

Paket nomlar maydoni bo'lganligi sababli, u boshqa paketlardan ularning barcha a'zolarini yoki ayrimlarini import qilishi mumkin. Bundan tashqari, paketning tarkibi boshqa paketlar bilan birlashtirilishi mumkin.

### 25.3. Paketni import qilish

Paketni import qilish – paketlar orasidagi yo'naltirilgan mu-

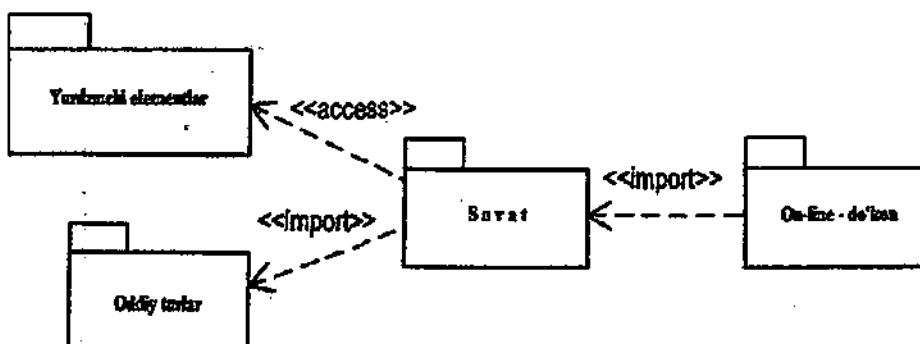
nosabat, unda bitta paketning a'zolari boshqa paketning nom maydoniga qo'shilishi mumkin. Paketni import qilish, malakasiz nomlardan foydalangan holda, faqat boshqa nom maydonida import qilingan paketning umumiy a'zolariga murojaat qilish imkonini beradi. Boshqacha qilib aytganda, import paketi import qilingan paketdan umumiy ko'rinishga ega bo'lgan barcha elementlarning nomlarini ularning nom maydoniga qo'shadi. Konseptual ravishda, paket importi import qilingan nom maydonining har bir shaxsiy elementi uchun elementni import qilishga tengdir, faqat alohida belgilangan element importi mavjud bo'lgan hollar bundan mustasno.

Paketni import qilish import qilinadigan nom maydonidan import qilinadigan paketga ochiq mil bilan uzlukli (nuqtali) chiziq yordamida tasvirlanadi. Nuqtali chiziq yonida paketni import qilish turini ko'rsatish uchun ikkita kalit so'zdan biri ko'rsatilishi kerak: <<import>> – paketni ommaviy import qilish uchun va <<access>> – paketni yopiq import qilish uchun.

Ommaviy (<<import>> qat'iy turi) paketni import qilishda import qilinadigan elementlar importi paketidan tashqarida ko'rinishi mumkin. Agar paketni import qilish yopiq bo'lsa, (<<access>> qat'iy turi), unda import qilingan elementlarni import paketidan tashqarida ko'rish mumkin emas. Oshkormas ravishda, paketni import qilish hammaga ochiq. Misol tariqasida, paket importining har xil turlarini aks ettiruvchi paket diagrammasining bir qismini ko'rib chiqamiz (25.6- rasm). Bunda paketning umumiy a'zolari "Oddiy turlar" "Savat" paketiga import qilinadi, shundan so'ng ular Savat paketning umumiy a'zolari bilan birga, "On-line - do'kon" paketiga import qilinadi. Boshqa tomondan, "Yordamchi elementlar" paketining umumiy a'zolari faqat "Savat" paketiga import qilinadi. Shunday qilib, ularga "On-line - do'kon" paketidagi malakasiz nomlar yordamida murojaat qilish mumkin emas.

UML 2.0dagi barcha paket a'zolarini import qilishdan tashqari, paketning alohida elementlarini ham import qilishga ruxsat

berilgan.



25.6- rasm. Ommaviy va yopiq paketlarning importi

#### 25.4. Element importi

Element importi butun paketni import qilmasdan ayrim elementlarni tanlab import qilish uchun ishlatiladi.

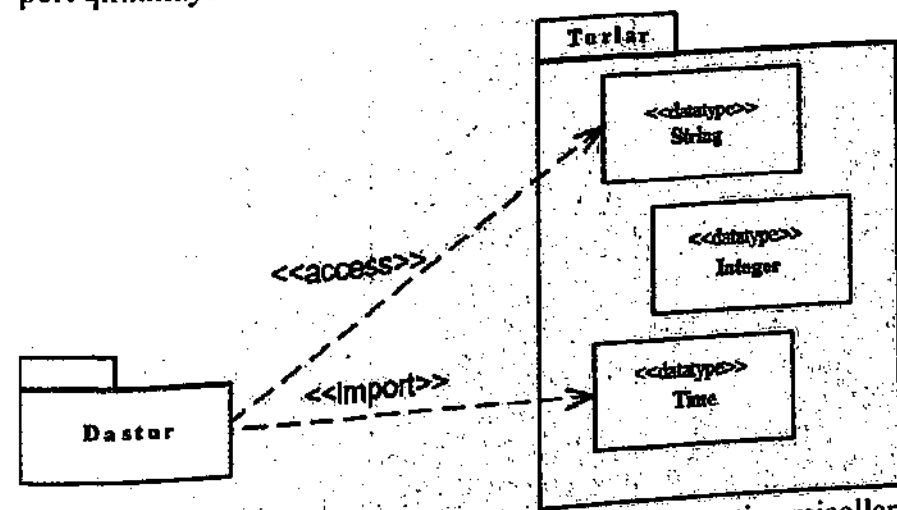
*Element importi (element import)* – import qilinadigan nomlar maydoni va paketning alohida elementi o'tasidagi yo'naltirilgan munosabatlar, bu esa ushbu elementga malakasiz nom yordamida murojaat qilish imkonini beradi.

Element importi import paketi (tobe mijoz) va import qilinadigan element (tobe manba) o'rasidagi bog'liqlik munosabatlari-ning ixtisoslashuvidir. Elementni import qilish boshqa paketdagi bitta paketning alohida elementini ushbu element nomini yoki uning taxallusini import paketining nom maydoniga qo'shish orqali aniqlashga imkon beradi. Shu bilan birga, elementni import qilish yoki paketni import qilish yordamida ushbu elementning keyingi importini boshqarish mumkin. Element importi import qilinadigan elementga havola shaklida taqdim etilishi mumkin. Bu shuni anglatadiki, import qilinadigan paketda import qilinadigan elementning xususiyatlarini o'zgartirish mumkin emas, chunki import qilingan elementni faqat unga tegishli bo'lgan paketda yoki boshqacha qilib aytganda, uning a'zosi bo'lgan paketda o'z-

gartirish mumkin.

Element importi import qilinadigan nom maydonidan import qilinadigan elementga yo'naltirilgan ochiq mil nuqtali chiziq yordamida tasvirlanadi. Agar ko'rinish chegarasi hammaga ochiq bo'lsa, unda nuqtali chiziq yonida <<import>> qat'iy turi tasvirlanadi. Yopiq ko'rinish chegarasini ko'rsatish uchun nuqtali chiziq yonida <<access>> qat'iy turi tasvirlanadi.

Misol tariqasida, har xil turdagi import elementlarini aks ettiruvchi paket diagrammasining bir qismini ko'rib chiqamiz (25.7-rasm). Berilganlar turi Time (Vaqt) uchun *Turlar* paketi ommaviy bo'lgani uchun, *Dastur* paketidagi elementlar malakasiz ravishda bo'lgani uchun, *Dastur* paketidagi elementlar malakasiz ravishda Time turiga murojaat qilishi mumkin. String (satr) turi *Dastur* paketida ishlatilishi mumkin, lekin uni ushbu paketdan boshqasiga import qilish mumkin emas, chunki ushbu elementni import qilish yopiq. Shu bilan birga, *Dastur* paketida Integer turiga murojaat qilish uchun, quyidagi shakldagi malakali nomdan foydalanish kerak – *Turlar::Integer*, chunki ushbu element *Dastur* paketiga import qilinmaydi.

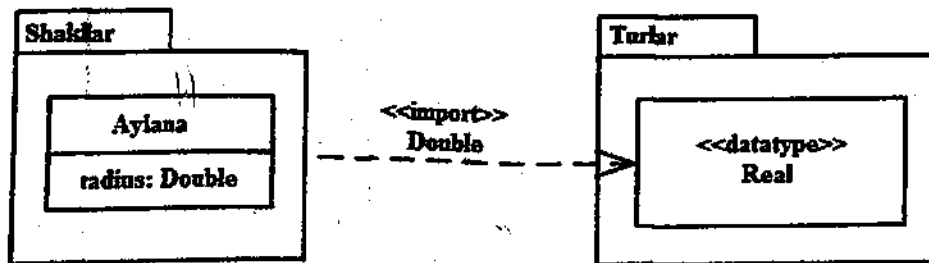


25.7- rasm. Ommaviy va yopiq elementlar importiga misollar

Agar import qilinadigan elementning malakasiz nomi import qilinadigan nom maydonida joylashgan boshqa har qanday elementning nomi bilan bir xil bo'lsa, undan foydalanish uchun ushbu import qilinadigan elementning nomi malakali bo'lishi kerak. Bunda, elementning malakali nomi import qilinadigan nomlar maydoniga qo'shilmaydi.

Agar import qilinadigan elementning nomi import qilinadigan paketning nom maydoniga qo'shilishi kerak bo'lsa, unda import qilingan elementning nomi o'rniga taxallusini (alias) belgilash kerak. Bunday taxallus import paketidagi boshqa biron bir a'zosining nomiga mos kelmasligi kerak. Taxallus nomi tobelik mili yonidagi `<<import>>` kalit so'zidan keyin yoki pastida ko'rsatilishi mumkin. Oshkormas ravishda, elementni import qilishda taxallus nomi ishlatilmaydi.

Shunday qilib, agar element importi ma'lum bir taxallusga ega bo'lsa, u import qilinadigan element nomi o'rniga ishlatilishi mumkin. Masalan, quyidagi parchada taxallus bilan elementning importi tasvirlaydi (25.8- rasm). Bu shuni anglatadiki, "Turlar" paketidagi *Real* turiga "Shakllar" paketidan *Double* deb murojaat qilish mumkin.



25.8- rasm. Taxallusli element importiga misol

Oshkormas ravishda, import qilinadigan paketdagi import qilinadigan elementning ko'rinish chegarasi, u import qilinadigan paketdagi elementning ko'rinish chegarasi bilan bir xil deb taxmin

qilinadi. Biroq, bu ko'rinish chegarasini yanada tasniflash mumkin. Agar import qilinadigan elementning ko'rinish chegarasi aniqlanmagan bo'lsa, unda biz ushbu elementning importiga ko'rinish chegarasini qo'shishimiz mumkin. Bunday holda, import qilingan element umumiy ko'rinish chegarasiga ega bo'lishi yoki umuman ko'rinmaydi.

Elementni import qilish muqobil ravishda matn shaklida taqdim etilishi mumkin, bu import qilinadigan elementni qiyofali qavslarda yagona tarzda yoki pastda yoki nom maydoni nomidan keyin aniqlaydi. Matn sintaksisi quyidagi shaklga ega:

```
{import '< malakali-nom>'} | {access '< malakali-nom>{'
```

Bundan tashqari, taxallus nomi ko'rsatilishi mumkin:

```
{import '< malakali-nom> as '<taxallus>'} |  
{access '< malakali-nom> as '<taxallus>{'
```

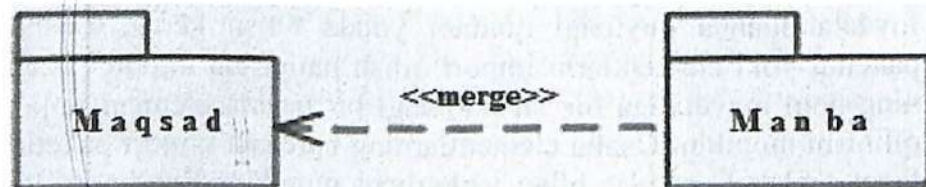
Shunday qilib, paket va elementlarni import qilish modelida foydalanilganda quyidagi qoidani yodda tutish kerak. Boshqa paketlar yoki elementlarni import qilish natijasida import paketining nom maydoniga bir xil nomdagi bir nechta element import qilinishi mumkin. Ushbu elementlarning barchasi import paketida faqat malakali nomlar bilan ishlatilishi mumkin. Bunday holda, ushbu elementlarning o'zi import qilinadigan nomlar maydoniga qo'shilmaydi.

## 25.5. Paketlarni birlashtirish

*Paketlarni birlashtirish (package merge)* – bu ikkita paket orasidagi yo'naltirilgan munosabat bo'lib, ulardan biri boshqa paket tarkibini qo'shish orqali o'z tarkibini kengaytiradi. Paketlarni birlashtirish tobelik munosabatlarining ixtisoslashuvidir. Bunda tobe paket *manba (source)*, mustaqil paket esa *maqsad (target)* deb nomlanadi. Paketlarni birlashtirish natijasida manba paketi-

ning tarkibi unga maqsad paketining tarkibini qo'shish orqali kengayadi.

Paketlarni birlashtirish umumlashma munosabatiga o'xshaydi, chunki manba paketi konseptual (g'oyaviy) ravishda maqsad paketining xususiyatlarini o'z xususiyatlariga qo'shib, natijada ikkala paketning xususiyatlarini birlashtirgan paketni hosil qiladi. Shunday qilib, konseptual ravishda paketlarni birlashtirishni ikkita paket tarkibiga asoslanib, birlashtirishda ishlatiladigan paketlar tarkibini birlashtirgan yangi paketni *yaratadigan amal* deb hisoblash mumkin. UML 2.0 ning mantiqi nuqtayi nazaridan aniq paketli birlashma modeli va barcha birlashmalar allaqachon bajarilgan model o'rasida farq yo'q. Paketlarning birlashishi manba paketidan maqsad paketiga chizilgan ochiq mil bilan uzlukli chiziq bilan tasvirlanadi. Bundan tashqari, ushbu uzlukli chiziq ustida (yonida) <<merge>> qat'iy tur tasvirlanadi (25.9- rasm).



25.9- rasm. Paketlarni birlashtirish munosabati uchun belgi

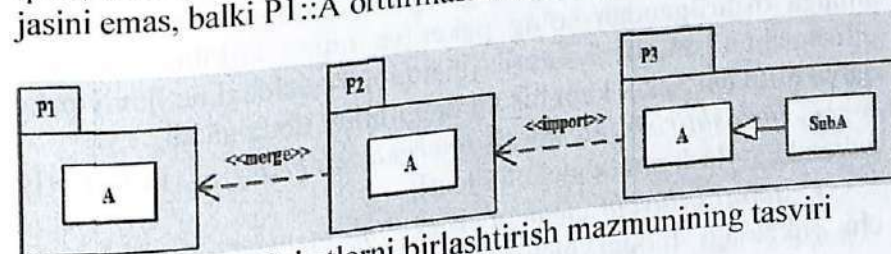
*Turli xil paketlarda aniqlangan elementlar bir xil nomga ega bo'lganda va bir xil tushunchani ifodalash uchun mo'ljallangan bo'lsa, paketlarni birlashtirish maqsadga muvofiqdir.* Ko'pincha u, ba'zi umumiy asosiy ta'riflardan boshlab, bir xil tushunchaning bir nechta ta'riflarini turli maqsadlarni uchun ifodalash uchun ishlatiladi. Ushbu asosiy tushuncha orttirma orqali kengayadi, bunda har bir orttirma boshqa paketlar bilan ketma-ket birlashadigan alohida paketda aniqlanadi. Biz maxsus maqsad uchun birlashtirish orttirmasini tanlash yordamida ba'zi tushunchalarning kerakli

ta'rifini olishimiz mumkin. UML 2.0dagi metamodelni aniqlashda, paketlarni birlashtirish zarur mexanizm hisoblanadi.

Agar birlashtiriladigan paketlardagi elementlar bir xil mohiyatni ifodalasa, unda ushbu elementlarning xususiyatlari goyaviy ravishda alohida hosil bo'lgan elementda rasmiy paketlarni birlashtirish qoidalariga (ular quyida ko'rib chiqiladi) muvofiq birlashtiriladi. Bunday holda, paketlarning modelga qo'shilishi faqat ushbu o'zgarishlarning bajarilishiga olib keladi, ammo modelga kiritilmagan o'zgarishlarning natijalari emas. Shu bilan birga, manba paketi va uning tarkibi birlashish natijasini ifodalash uchun zarur deb hisoblanadi.

Shunday qilib, paket manbasidagi model elementiga har qanday havola birlashish natijalariga havolani keltirib chiqaradi. Bunday holda, orttirmaning o'zi manba paketiga jismonan kiritilmaydi. Birlashish munosabatlarining bu xususiyatini quyidagicha tasvirlanishi mumkin. Diagrammaning bir qismini ko'rib chiqamiz (25.10- rasm), unda ikkita P1 va P2 paketi A sinfidagi bir xil turdagi orttirmalarni aniqlaydi.

P2 paketi P1 paketining tarkibi bilan birlashadi, bu esa P1::A orttirmani P2::A orttirma bilan birlashtirishga olib keladi. P3 paketi P2 paketining tarkibini import qiladi, shuning uchun unda **SubA** deb nomlanuvchi, A qism-sinfni aniqlash mumkin. Bunda, P3 paketidagi A sinfi, nafaqat P2::A ning orttirmasini, balki P3::A, P1::A ni P2::A ga birlashishi natijasini ham ifodalaydi. Shuni ta'kidlash kerakki, agar boshqa biron bir paket P1 ni import qilsa, u holda import paketidagi A ga havola P2::A qo'shilish natijasini emas, balki P1::A orttirmasini ifodalaydi.



25.10- rasm. Paketlarni birlashtirish mazmunining tasviri

Paketlarni birlashtirish qoidalarini tushunish uchun 3 xil mohiyatni aniq ajratish kerak: birlashtiriladigan orttirma, masalan, 25.10- rasmdagi P1::A, qabul qiluvchi orttirma, masalan, P2::A va birlashtiriladigan o'zgartirishlar natijasi. Asosiy qiyinchilik, manba paketi va uning tarkibi operandni ifodalash uchun ham, paketlarni birlashtirish natijalarini ifodalash uchun ham xizmat qilishidan kelib chiqadi. Bunda ma'lum bir sharh paketlarning birlashishini ko'rib chiqadigan kontekstiga bog'liq.

Masalan, paketlarni birlashtirish amali kontekstida ko'rib chiqilgan P2 paketi ushbu birlashishning operandasi bo'lgan orttirmani anglatadi (25.10- rasm). Biroq, paketni import qilish kontekstida ko'rib chiqilgan xuddi shu P2 to'plami paketlarni birlashtirish natijasini anglatadi. Modeldagi bitta elementni ikki xil talqini tushunmovchiliklarga olib kelishi mumkin. Birlashish mexanizmini tushunishga aniqlik kiritish uchun quyidagi tushunchalar qo'shimcha ravishda ko'rib chiqiladi:

- *birlashtirish paketi (merged package)* – birinchi birlashtirish operandi, ya'ni qabul qiluvchi paketga birlashtirilgan va diagrammalardagi birlashtirish milining maqsadi bo'lgan paket;

- *qabul qiluvchi paket (receiving package)* – bu ikkinchi birlashtirish operandi, ya'ni konseptual ravishda birlashish natijalarini o'z ichiga olgan va diagrammalardagi birlashtirish milining manbai bo'lgan paket. Ushbu atama birlashtirish amalga oshirilishidan oldin paket va uning tarkibiga murojaat qilish uchun ishlatiladi;

- *natijaviy paket (resulting package)* – birlashish natijalarini goyaviy ravishda o'z ichiga olgan paket. Ushbu atama birlashma amalga oshirilgandan so'ng paket va uning tarkibiga murojaat qilish uchun ishlatiladi. Grafik jihatdan, modeldagi natijaviy paket qabul qiluvchi paket kabi bir xil belgi bilan ifodalanadi;

- *birlashtirish elementi (merged element)* – birlashtirish paketida joylashgan model elementi;

- *qabul qiluvchi element (receiving element)* – qabul qiluvchi paketdagi model elementi. Agar bu element birlashtirish

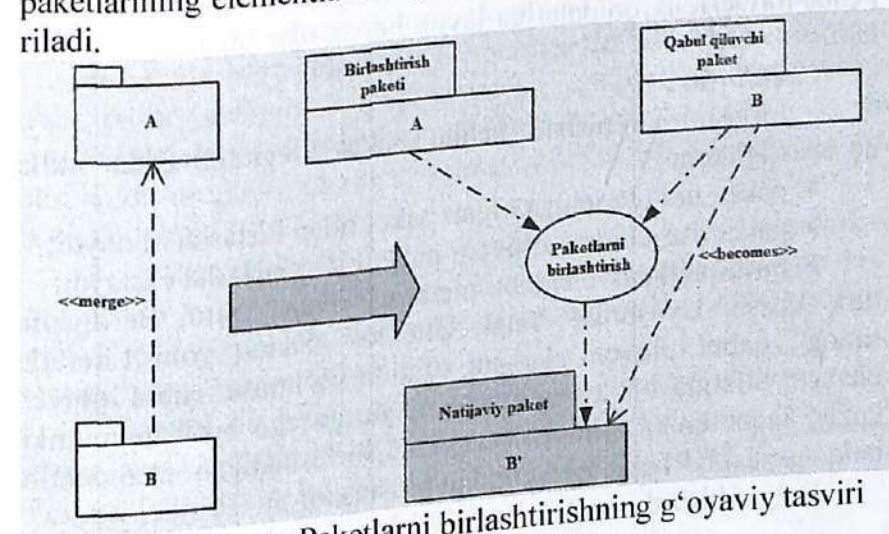
elementiga mos kelsa, ikkalasi ham natijaviy elementni olish uchun birlashadi. Bu birikma, birlashishdan oldin elementga murojaat qilish uchun, ya'ni uning natijasiga emas, balki orttir-maga murojaat qilish uchun ishlatiladi;

- *natijaviy element (resulting element)* – birlashishni amalga oshirgandan keyingi paketdagi model elementi. Birlashtiriladigan elementga mos keladigan qabul qiluvchi element uchun u qabul qiluvchi element bilan bir xil hisoblanadi, agarda birlashma amalga oshirilgan bo'lsa. Qabul qiluvchi elementga mos kelmaydigan birlashtiriladigan elementlar uchun u birlashtiriladigan elementdir. Birlashtirish elementiga mos kelmaydigan qabul qiluvchi elementlar uchun u qabul qiluvchi element bilan bir xil;

- *element turi (element type)* – UML 2.0dagi har qanday ruxsat etilgan element turi, masalan, parametr yoki atribut turi;

- *element metaturi (element metatype)* – model elementining turi masalan, Sinflovchi, Uyushma, Tavsiflar.

Ushbu atamalar paketlarni birlashtirishning g'oyaviy ko'rinishini quyidagi chizma orqali tasvirlash uchun mo'ljallangan (25.11- rasm), bu UML diagrammasi emas. Bunday holda, A va B paketlarining elementlari B' paketining nom maydoniga birlashtiriladi.



25.11- rasm. Paketlarni birlashtirishning g'oyaviy tasviri

Biroq, bu vakillik paketlarni birlashtirish mazmununi tavsiflashning qulay namunasidir va modelda ko'rinmaydi, chunki paketlarni birlashtirish mavjud bo'lganda haqiqiy model hech qanday tarzda o'zgartirilmaydi.

Birlashishni amalga oshirishning umumiy tamoyili quyidagicha. Birlashishdan keyingi har qanday elementning imkoniyatlari birlashishdan oldingisidan kam bo'lmashligi kerak. Bu, masalan, natijaviy yo'naltirish, qabul qiluvchi model elementining karraligi yoki ko'rinish chegarasi paketlarning birlashishi natijasida kamaytirilmasligi yoki chiqarib tashlanmasligi kerakligini anglatadi.

Paketlarni birlashtirish mantiqi cheklovlar va o'zgarishlar ko'rinishidagi har xil turdagi elementlar uchun ko'plab qoidalar bilan belgilanadi. Bunday holda, cheklovlar ruxsat etilgan paketlarni birlashtirish uchun old shartlarni belgilaydi, o'zgarishlar esa ruxsat etilgan mantiqiy samaralarni yoki keying shartlarni tavsiflaydi.

## 25.6. Paketlarni birlashtirishning umumiy qoidalari

Birlashtiriladigan element va qabul qiluvchi element, metatur uchun muvofiqlik qoidalariga javob bersa, ular bir-biriga mos keladi.

### **Cheklovlar:**

- birlashtirish nisbati uchun paket diagrammasida sikllar bo'lmashligi kerak;
- paket, uni o'z ichiga olgan paket bilan birlasha olmaydi;
- paket o'z ichida saqlovchi paket bilan birlasha olmaydi;
- birlashadigan element metaturi: Paket, Sinf, Berilganlar turi, Xossa, Uyushma, Amal, Cheklov, Sanoq yoki Literallar sanog'i, qabul qiluvchi element sifatida bo'lmasa, qabul qiluvchi element sifatida bir xil nom va metaturga ega bo'lishi mumkin emas, faqat qabul qiluvchi element birlashadigan elementning aniq nusxasi bo'lgan holat bundan mustasno;
- agar birlashishni amalga oshirish uchun zarur bo'lgan bar-

cha cheklovlar qondirilsa, paketlarni birlashtirish mumkin;

➤ xususiyatlar va parametrlar kabi mos keluvchi turlangan elementlar izchil turlarga ega bo'lishi kerak. Sinflar yoki berilganlar turlari kabi turlar uchun tur bir xil tur yoki umumiy superturdir. Boshqa barcha holatlarda muvofiqlik turlar bir xil bo'lishi kerakligini anglatadi;

➤ qabul qiluvchi element birlashtiriladigan elementlarning hech biriga oshkora havolalarga ega bo'lishi mumkin emas.

### **O'zgartirishlar:**

❖ agar birlashtiriladigan yoki qabul qiluvchi elementga mos keladigan element bo'lmasa, ular to'liq natijaviy paketga ko'chiriladi. Ushbu qoida oskormas ravishda qabul qilinadi;

❖ bir-birining aniq nusxalari bo'lgan mos keladigan nomlar va metaturlar bilan ikkita elementning birlashish natijasi – qabul qiluvchi element bo'ladi;

❖ mos keladigan elementlar metaturga xos bo'lgan o'gartirish qoidalariga muvofiq birlashtiriladi va natijalar natijaviy paketga kiritiladi;

❖ natijaviy paketdagi turlangan elementlarga yonaltirilgan barcha havolalar, natijaviy elementlarga yo'naltirilgan havolalarga aylantiriladi, lekin ularning mos ravishdagi orttirmalariga emas;

❖ barcha mos keladigan elementlar uchun – agar bir-biriga mos ikkita element yopiq ko'rinish chegarasiga ega bo'lsa, natijaviy element yopiq ko'rinish chegarasiga ega bo'lishi kerak, aks holda natijaviy element ommaviy ko'rinish chegarasiga ega bo'lishi kerak;

❖ sinflavchilarning barcha bir-biriga mos keluvchi elementlari uchun – agar ikkala mos keladigan element ham mavhum bo'lsa, natijaviy element mavhum bo'ladi, aks holda natijaviy element mavhum bo'lmaydi;

❖ barcha bir-biriga mos keluvchi elementlar uchun – agar ikkala mos keladigan element ham hosila bo'lmasa, natijaviy element ham hosila bo'lmaydi, aks holda natijaviy element hosila bo'ladi;

❖ barcha bir-biriga mos keluvchi karralik elementlar uchun – olingan karralik pastki chegarasi mos keladigan karralikning pastki chegaralaridan kichikroqdir;

❖ barcha mos keladigan karralik elementlar uchun – olingan karralikning yuqori chegarasi mos keladigan karralikning yuqori chegaralaridan kattaroqdir;

❖ model yoki birlashtiriladigan yoki qabul qiluvchi elementga tegishli har qanday qat'iy turlar, natijaviy elementga ham tegishli.

### 25.7. Paketlar uchun qoidalar

Paket elementlari, nomi va metaturi bo'yicha mos keladi.

#### **O'zgarishlar:**

▪ Birlashtirilayotgan paketdagi joylangan paket, natijaviy paketda o'z nomi bilan joylangan paketga aylantiriladi, agarda natijaviy paketda mos keladigan joylangan paket bo'lmasa. Bunda, birlashtirilgan joylangan paket mos keladigan qabul qiluvchi joylangan paket bilan rekursiv ravishda birlashadi;

▪ qabul qiluvchi paketga tegishli bo'lgan elementning importi, natijaviy paketdagi elementning tegishli importiga aylantiriladi. Import qilinadigan elementlar birlashtirilmaydi, agar u paketlarni import qilinadigan elementga yoki uning taxallusiga ega bo'lgan paketga birlashtirmasa.

### 25.8. Sinflar va berilganlar turlari uchun qoidalar

Sinf yoki Tur ko'rinishidagi elementlar, nomi va metaturi bo'yicha bir-biriga mos keladi.

#### **O'zgarishlar:**

• birlashtiriladigan sinflovchining barcha xossalari qabul qiluvchi sinflovchi bilan birlashtirilib, keyinchalik aniqlangan xususiyatlarni o'zgartirish qoidalariga muvofiq natijaviy sinflovchini hosil qiladi;

• joylangan sinflovchilar bir xil qoidalarga muvofiq rekursiv ravishda birlashadi.

### 25.9. Xossalar uchun qoidalar

Xossa ko'rinishidagi elementlar, nomi va metaturi bo'yicha bir-biriga mos keladi.

#### **Cheklovlar:**

• mos keladigan xossalarning statik va statik bo'lmagan xossalari bir xil bo'lishi kerak;

• mos keladigan xossalarning noyobligini tavsiflash bir xil bo'lishi kerak;

• mos keladigan xossalar bilan bog'liq har qanday cheklovlar, ziddiyatli bo'lmasligi kerak;

• mos keladigan xossalar bilan bog'liq har qanday qayta aniqlashlar ziddiyatli bo'lmasligi kerak.

#### **O'zgarishlar:**

✓ mos keladigan qabul qilish xossaga ega bo'lmagan birlashtiriladigan xossalar uchun natijaviy sinflovchida yangi yaratilgan xossaga aylanadi. Hamma natijaviy xossalar birlashtirilgan xossa bilan bir xil bo'ladi;

✓ mos keladigan qabul qiluvchi xossa ga ega bo'lgan birlashtiriladigan xossalar uchun natijaviy xossa bir xil nom va xossalarga ega bo'lgan xossadir, bu xossalar turlicha bo'lgan hollar bundan mustasno. Ushbu xossalar har xil bo'lsa, natijaviy xossalari tegishli o'zgarish qoidalarini qo'llash orqali aniqlanadi;

✓ mos keladigan xossalar uchun – agar ikkala xossa ham "faqat o'qish uchun" bo'lsa, natijaviy xossa ham "faqat o'qish uchun" bo'ladi. Aks holda, natijadaviy xossa "faqat o'qish uchun" emas;

✓ mos keladigan xossalar uchun – agar ikkala xossa ham tartibsiz bo'lsa, natijaviy xossa ham tartibsiz bo'ladi. Aks holda, natijaviy xossa buyurtma qilinadi;

✓ mos keladigan xossalar uchun – agar xossalarning hech biri ba'zi bir hosila birlashmasining qism-to'plami bo'lmasa, natijaviy xossa ba'zi bir qism-to'plam sifatida tayinlanmaydi. Aks holda, natijaviy xossa ushbu hosila birlashmasining ba'zi bir qism-to'plami sifatida tayinlanadi;

✓ mos keladigan xossalar uchun – mos keladigan turli xil bekor qilishlar xossalar birlashmalarining birlashmasidir;

✓ mos keladigan xossalar uchun – mos keladigan xossalarga nisbatan turli xil cheklovlar birlashmasining birlashmasidir;

✓ mos keladigan xossalar uchun – agarda elementlarni birlashtirish va/yoki qabul qilish noyob bo'lmasa, natijaviy element noyob bo'lmaydi. Aks holda, natijaviy element noyob sifatida belgilanadi;

✓ natijaviy xossaning turi natijaviy paketdagi tegishli turga murojaat qilish uchun o'zgartiriladi.

### 25.10. Uyushmalar uchun qoidalar

Uyushma ko'rinishidagi elementlar nomi va uyushmaning oxiri bo'yicha bir-biriga mos keladi. Uyushmalar oxirining bir-biriga mos kelishi, xossalar qoidasiga o'xshash ism va tur bo'yicha moslikni anglatadi. Uyushmalarining mosligi, tasodifiy holda, uyushmalarining umuman nomi bo'lmagan holatni o'z ichiga oladi. Ushbu qoidalar xossalar uchun ilgari tavsiflangan qoidalarni to'ldiradi.

#### Cheklovlar:

❖ ushbu qoidalar faqat ikkilik uyushmalarga nisbatan qo'llaniladi. n-ar uyushmalarni birlashtirish uchun umumiy qoidalardan foydalanish kerak;

❖ agar uyushmaning mos keladigan oxiri aralash (kompozit) bo'lsa, uyushmaning qabul qiluvchi oxiri aralash bo'lishi kerak;

❖ agar uyushmaning mos keladigan oxiri, shu uyushmaga tegishli bo'lsa, uyushmaning qabul qiluvchi oxiri uyushmaga tegishli bo'lishi kerak.

#### O'zgarishlar:

✓ mos keluvchi uyushmalarining birlashishi sinflovchilar uchun birlashish qoidalaridan foydalangan holda birlashma sinflovchilarining birlashishi orqali amalga oshiriladi va uyushma oxirining tegishli o'ziga xos xossalarning birlashishi uyushmalarining xossalari va uchlari qoidalariga muvofiq amalga oshiriladi;

✓ uyushmalarining mos keladigan oxirlari uchun – agar uyushmaning biron bir oxiri yo'naltirilmagan bo'lsa, unda uyushmaning natijaviy oxiri ham yo'naltirilmagan bo'ladi. Boshqa barcha holatlarda, uyushmaning natijaviy oxiri yo'naltirilgan bo'ladi.

### 25.11. Amallar uchun qoidalar

Amal ko'rinishidagi elementlar – qaytish turining qiymatini hisobga olmaganida, nomi, parametr tartibi va parametr turlari bo'yicha ularga mos keladi.

#### Cheklovlar:

- amal parametrlari va turlari, xossalar uchun aniqlangan tur va karralik bo'yicha bir xil qoidalarga amal qilishi kerak;
- agar mos keladigan birlashtirish amali so'rov bo'lsa, qabul qiluvchi amal so'rov bo'lishi kerak.

#### O'zgarishlar:

➤ mos keladigan qabul qiluvchi amalga ega bo'lmagan birlashtirish amali uchun natijaviy amal birlashtirish sinfidagi kabi bir xil nom va imzoga ega amal hisoblanadi;

➤ mos keladigan qabul qilish amaliga ega bo'lgan birlashtirish amali uchun hosil bo'lgan amal mos keladigan birlashtirish va qabul qilish amalining birlashishi natijasidir. Bunda parametrlarni o'zgartirish ilgari aniqlangan xossalarni o'zgartirish qoidalariga muvofiq amalga oshiriladi.

### 25.12. Sanoq turi qoidalari

Literallarning turlari bo'lgan elementlar, o'z sanog'i va lite -

ralning nomi bilan mos keladi.

**Cheklov:** sanoq turidagi mos keladigan literallar bir xil tartibda bajarilishi kerak.

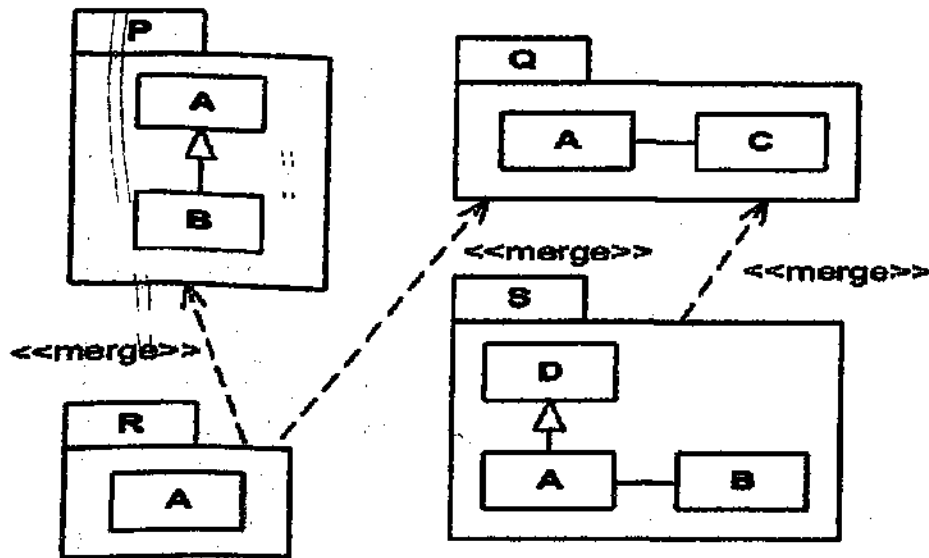
**O'zgarishlar:** birlashtirilgan sanoq turidagi mos kelmaydigan literallar sanoq turidagi qabul qiluvchi sanoq turida birlashadi.

### 25.13. Cheklovlar uchun qoidalar

**Cheklov:** cheklovlar bir-biriga mos kelishi kerak.

**O'zgarishlar:** modelning birlashtiriladigan elementlaridagi cheklovlar modelning qabul qiluvchi elementidagi mos keladigan cheklovlarga kon'yunktiv ravishda qo'shiladi.

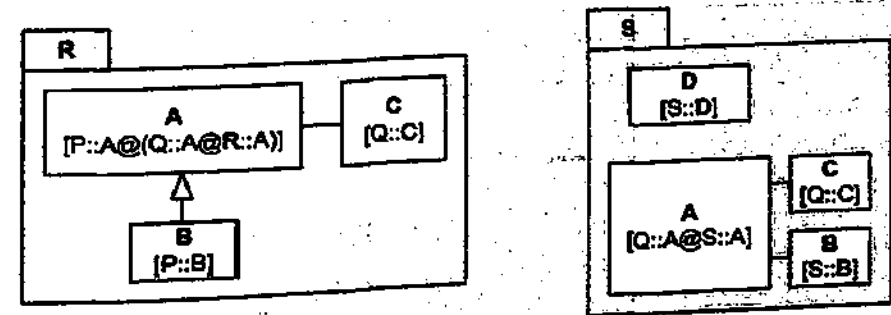
Taqdim etilgan birlashish qoidalari quyidagi diagramma parchasi bilan tasvirlanishi mumkin (25.12- rasm.), unda P va Q paketlari R paketda birlashadi, Q paketi esa faqat S paketida birlashadi.



25.12- rasm. Paketlarni birlashtirish qoidalari tasviri

Ushbu birlashmalarning natijalari o'zgartirilgan R va S paket-

lari shaklida quyidagi diagrammada ko'rsatilgan (25.13- rasm.). Kvadrat qavs ichidagi iboralar qaysi shaxsiy ortirmalarni yakuniy natija olish uchun birlashtirilishini bildiradi. Bunda @ belgisi birlashma bayonotini ko'rsatish uchun ishlatiladi, bu umuman UML 2.0ning standart yozuvining bir qismi emas. Ushbu belgidan foydalanish faqat qulaylik sabablari bilan belgilangan.



25.13- rasm. Paketlarni birlashtirish natijalari

Paket diagrammasining tavsifi paket diagrammalarini tuzishda ishlatilishi mumkin bo'lgan barcha ko'rib chiqilgan grafik belgilarning qisqacha mazmuni bilan yakunlaymiz (25.1- jadval). Paket diagrammalarida keltirilgan grafik tugunlar va yo'llar boshqa diagrammalarda ham ishlatilishi mumkin.

25.1- jadval. Paket diagrammasidagi grafik tugunlar va yo'llar

Asbob	Grafik element turi
	Paket
	Sinf
	Paket importi, umumiy (package import, public)
	Paket importi, yopiq (package import, private)
	Paketlarni birlashtirish (package merge)

## GLOSSARIY

**"Bori qanaqa" (AS-IS) modeli** – tashkilotdagi mavjud tijorat jarayonlarini aks ettirish.

**Abstrakt arxitektura** – domen vazifalarining quyi tizimlarga yoki quyi tizimlar shajarasiga dekompozitsiyalangan tuzilish, ularning har bir darajasida tanlangan qism-tizimlar tarkibi variantlarini belgilaydigan tanlangan parametrlar va cheklovlar qayd etiladi.

**Abstraktsiya** – muhim narsani ikkilamchi narsadan ajratish, amalga oshirishni belgilaydigan g'oyani ko'rish qobiliyati.

**Agregatsiya** – bir qator tushunchalarni yangi g'oyaga birlashtirish ("qism-butun" kabi munosabat), uning umumiy xususiyatlari tarkibiy qismlarning belgilarining yig'indisi yoki sezilarli darajada yangi xususiyatlar bo'lishi mumkin.

**Aktor (harakatlanuvchi shaxs)** – bu tashqarida joylashgan mohiyat bo'lib, u modellashtiriladigan tizimdan tashqarida va u bilan bevosita o'zaro harakat qiladi.

**Aloqa (Relationship)** – ko'rib chiqilayotgan mavzu sohasi uchun muhim bo'lgan ikki mohiyat o'rasidagi nomlangan yushma.

**Aloqa diagrammasi** – semantik jihatdan ketma-ketliklar diagrammasiga ekvivalent bo'lgan xulq-atvor tavsiflaridan iborat usul.

**Amaliy tizim** – bu oxirgi foydalanuvchining aniq vazifalarini bajarish uchun mo'ljallangan dasturiy ta'minot muhandisligi mahsuloti.

**Amaliy tizimlar oilasi** – umumiy funktsional xususiyatlarga va boshqaruvga ega bo'lgan ko'plab amaliy tizimlar.

**Amallar g'oyasi** – bu missiyasi oldidan, topshiriqdan va keyingi bosqich natijalariga asoslangan ishlash maqsadlariga erishish uchun zarur bo'lgan tizimning ketma-ket va/yoki parallel amallaridagi ish jarayonining tavsifi.

**Aniqlashtirish** – muhim xususiyatlarning qo'shilishi, buning natijasida g'oyaning mazmuni kengayadi va hajmi esa torayadi.

**Artefakt (artifact)** – bu dasturiy ta'minotni ishlab chiqish bo'yicha mutaxassislar faoliyatining har qanday mahsuloti.

**Arxitektura loyihasi** – qurilayotgan tizimning tarkibiy xususiyatlarini aniqlash.

**Atribut** – tizimning funktsional yoki jismoniy xususiyatlarini tasniflaydi.

**Avtomat diagrammasi (holat diagrammasi)** – bu UML tilida holatni har tomonlama tavsiflash usullaridan biri.

**Avtomatizatsiyalashtirilgan tizimlarni yaratish jarayoni** – bu belgilangan talablarga javob beradigan avtomatizatsiyalashtirilgan tizimni yaratish uchun zarur va etarli bo'lgan ish bosqichlari davrida birlashtirilgan, vaqt o'tishi bilan buyurtma qilingan va o'zaro bog'liq bo'lgan to'plam.

**Axborot ta'minoti** – foydalanuvchilarga uning mazmuni va foydalanish shartlari to'g'risida ma'lumot berish uchun vositalar to'plami.

**Axborot tizimi** – qo'yilgan maqsadga nisbatan berilganlarni to'plash, saqlash, ishlov va chiqarish uchun vosita, usul va xodimlarning o'zaro bog'langan majmuasi.

**Aylanma (spiral, evolyutsion yoki yaqinlashuvchi) model** – ishlab chiqarilgan mahsulot elementlarini qayta ishlash uchun har qanday oldingi jarayonga qaytishga imkon beradigan tizimni ishlab chiqishning hayotiy siklidagi jarayonlar modeli.

**Ayon funktsional talablar** – bu foydalanuvchi tomonidan amalga oshiriladigan funktsiyalar.

**Batafsil loyihalash** – ma'lum bir muhit uchun funktsiyalarni amalga oshirish tafsilotlarini va tizimning tegishli tarkibiy qismlari o'rtasidagi aloqalarni aniqlash.

**Belgilangan ish muhiti** – bu iqtisodiy, natija va omon qolish sabablariga ko'ra har bir tizim belgilagan, ya'ni chegaralangan ish muhitiga ega bo'lish.

**Boshqarish ob'yektlari** – bu interfeys ob'yektlarini ob'yekt ob'yektlariga aylantirish funksiyalari, ko'pincha tizimdagi berilganlarni qayta ishlash algoritmlarini aks ettiradi.

**Dasturiy ta'minot hayot siklining modeli** – bu jarayonlar, ishlar va dasturiy mahsulotni ishlab chiqish, ishlatish va qo'llab-quvvatlashni o'z ichiga olgan vazifalardan iborat bo'lib, unda talablarni belgilashdan boshlab, toki foydalanishni to'xtatishgacha bo'lgan tizim hayotini qamrab oladi.

**Dasturiy ta'minot hayot siklining modeli (Software Life Cycle Model, SLCM)** dasturiy ta'minotni ishlab chiqish bo'yicha harakatlar qanday amalga oshirilishini, ushbu harakatlarning "ketma-ketligini" tavsiflash orqali vositalar chizma shaklida tushuntiradi.

**Dasturiy ta'minot muhandisligi** – ommaviy takror ishlab chiqarishga qodir bo'lgan dasturiy ta'minotni rejalashtirish, ishlab chiqish, ishlatish va kuzatish usullari va vositalari tizimi.

**Dasturiy ta'minot sifati** – dasturiy ta'minotning ishlab chiqish talablariga muvofiq buyurtmachini qondirish uchun yaroqliligini aniqlaydigan xususiyatlar to'plami.

**Dasturiy ta'minot sifatini kafolatlash** – hayot siklining har bir bosqichida standartlar va protseduralarga muvofiq erishilgan sifatini tekshirish va tasdiqlash bo'yicha harakatlar.

**Dasturiy ta'minot tizimini amalga oshirish** – loyihaviy yechimlarini ishlaydigan tizimga aylantirish.

**Dasturiy ta'minot tizimining arxitekturasi** – tizimning hisoblash qism-tizimlari va ular orasidagi interfeyslar nuqtayi nazaridan ta'rifi, muammoning parchalanish qoidalarini aks ettirish.

**Dasturiy ta'minot tizimining ishonchliligi** – bu tizimning ma'lum bir ish sharoitida ma'lum vaqt davomida dastlabki ma'lumotlarni natijalarga aylantirish jarayonida o'z xususiyatlarini (ishonchliligi, barqarorligi va boshqalar) saqlab qolish qobi-

liyati.

**Diagramma** – bu sinflar, ssenariylar, holatlar va boshqalar yordamida tizimni modellashtirishning grafik tasviri.

**Dinamik sinov** – xatolarni aniqlash, ularning sabablarini aniqlash va yo'q qilish uchun dasturlarni bajarish.

**Disk raskadrovkasi** – dasturda xatolar mavjudligini tekshirish va yangilarini kiritmasdan ularni yo'q qilish.

**Domen** – hal qilishning o'xshash usullariga imkon beradigan muammoli sohadagi muammolar doirasi.

**Domen harakati** – domen elementlarining vaqt o'tishi bilan holatdan holatga o'tishi.

**Ekvayer** – bu savdo-xizmat ko'rsatish tarmog'idagi terminallar va bankomatlardan tashkil topgan kartalarga xizmat ko'rsatish punktlari bilan o'zaro aloqada bo'lgan barcha amallarni amalga oshiradigan bank yoki kompaniya.

**Eski tizim** – bu ba'zi bir tijorat jarayonlarini qo'llab-quvvatlash uchun har qanday usul va texnologiyalar asosida yaratilgan mavjud operatsion tizim.

**Faoliyat (activity)** holatining alohidagi xususiy voqeligi, ya'ni uzoq vaqt davomida, atomar bo'lmagan hisob-kitoblar bilan tavsiflanish holati.

**Faoliyat diagrammasi** – bu xatti-harakatni tasvirlashning yana bir usuli bo'lib, u algoritmnning blokli chizmalar ravishdagi namoyishini eslatadi.

**Fasad** – tizim foydalanuvchisi nuqtayi nazaridan qayta muhandislik qilishda zarur bo'lgan tarkiblarni tizimining xususiyatlariga kiritish mexanizmi.

**Foydalanish diagrammasi** – tizim funktsionalligining eng umumiy ifodasi.

**Foydalanuvchiga missiyaga yo'naltirilgan amaliy ehtiyojlarni bashoratli ravishda qondirish imkoniyatini berish** – har bir tizimning maqsadi (ya'ni, mavjudlik sababi) va foydalanuvchi-

(lar) uchun qiymati borligini nazarda tutish.

**Funksional bo'lmagan talablar** – bu muayyan tizim funksiyalari bilan bog'liq bo'lishi mumkin bo'lgan cheklovlar yoki sifatlar. Amalga oshirish muhitida dasturiy ta'minot tizimining tashkiliy, ijro etuvchi, operatsion jihatlarini tavsiflash.

**Funksional talablar** – shaxsiy funksiyalarning yuqori darajadagi foydalanish holatlari o'z tuzilishiga ega bo'lganda, foydalanish holatini kengaytirish. Tizimning maqsadlari va funksiyalarini va ularni kompyuterda bajarish tamoyillarini belgilash.

**Funksionallik** – atributlari o'lchanadigan ishlash parametrlari, shu jumladan ishonchlilik, barqarorlik va xavfsizlikdan iborat holat.

**Funksional to'liqlik** – dasturiy ta'minotning maqsadiga muvofiq maxsus muammolarni hal qilish uchun asosiy funksiyalarning etarlilik darajasini ko'rsatuvchi atribut.

**Funksional tuzilma** – elementlari korxonada bo'linmalari tomonidan amalga oshiriladigan funksiyalar va munosabatlar-mehnat ob'yektlarining uzatilishini ta'minlaydigan aloqalar bo'lgan tuzilma.

**Funksiya** – belgilangan samaradorlik chegaralari doirasida muayyan maqsadga erishish uchun tizim elementi tomonidan bajariladigan amal, faoliyat, jarayon yoki harakat.

**Hamkorlik (collaboration)** – bu biron bir maqsadga erishish uchun o'zaro harakatlanuvchi ob'yektlar to'plami.

**Har bir element aniq belgilangan va chegaralangan imkoniyatlarga egaligi** – har bir element qandaydir yuqori darajadagi maqsad yoki maqsadli vazifani bajarish uchun ishlashi kerakligi.

**Harakat (action)** – bu eng sodda atomar hisobdir.

**Hayotiy sikl modeli** – dasturiy mahsulotni ishlab chiqish jarayonlaridagi ishlarning ketma-ketligining odatiy chizmasi.

**Hodisa** – bu ob'jekt holatning o'zgarishiga va tizimdagi boshqa holatga o'tishga olib keladigan hodisa.

**Holat (state)** – bu ob'jektning yashash davri bo'lib, bunda ob'jekt ma'lum bir shartni qondirishi va o'z faoliyatini amalga oshirishi yoki biron bir hodisaning sodir bo'lishini kutadigan mavjudlik vaqti.

**Holat modeli** – sinf ob'jektining holatini o'zgartirish dinamikasini ko'rsatish, bu uning xatti-harakatlarni o'zgartirish.

**Homaki loyiha** – tizim va uning qismlari uchun dastlabki loyihaviy yechimlarni ishlab chiqish.

**Ilova** – bu tamoyil va amaliyotlarning eng yaxshi qo'llanilishini ifodalovchi dastur.

**Imkoniyat** – bu funksiya (harakat)ni tashqi buyruqlar bilan tugatilgunga qadar, belgilangan muddatda bajarish yoki zaxiralar tugaguniga qadar ishlashning belgilangan darajasida bajarish uchun tashqi stimuly tomonidan faollashtirilgan yoki qo'zg'atilgan aniq, o'ziga xos xususiyat.

**Instansiya** – parametrlangan mavhum andoza sinf va abdoza parametrlarini aniqlash orqali haqiqiy sinf o'rasidagi bog'liqlik.

**Integratsiyalashgan terma** – tizimdagi jismoniy elementlar, ob'yektlar yoki tarkiblarning shajaraviy darajalaridan tashkil topishi.

**Interfeys (interface)** – bu nomlangan amallar termasi bo'lib, iste'molchi tomonidan so'ralishi mumkin bo'lgan xizmatlar termasini va xizmat ko'rsatuvchi provayder tomonidan shu xizmatlarni taqdim etilishini aniqlovchi.

**Interfeys ob'yektlari** – bajarish uchun xabarlar orqali uzatiladigan parametrlarning tavsifi sifatida taqdim etilgan dasturlar to'plami.

**Ish** – amal yoki funksiyani bajarish bilan bog'liq jismoniy yoki funksional atributni tavsiflovchi miqdoriy o'lchov. Ish atributlari – miqdor (qancha) va sifat (qanchalik).

**Ish jarayoni** – foydalanuvchi tomonidan tizimga xizmat ko'rsatish bo'yicha harakatlar.

**Ishlab chiqish jarayoni** – ishlab chiquvchining talablar muhandisligi, dasturiy mahsulotni loyihalash, kodlash va sinovdan

o'tkazish bo'yicha harakatlari yetkazib berish jarayoni, ishlab chiqilgan mahsulotni xaridorga topshirish bo'yicha harakatlar.

**Ishlash darajasi** – ob'ektiv, o'lchanadigan parametr, tizimning ssenariy farazlari, boshlang'ich shartlari va ish sharoitlari to'plamiga asoslangan funksiyani bajarish qobiliyatini chegaralash xizmati.

**Istalgan** – rivojlanish jarayoniga katta muammo to'sqinlik qilmasa, olinishi mumkin bo'lgan talablar.

**Jarayon modeli** – bu dasturiy ob'yektlar holatining o'zgarishiga hamroh bo'ladigan muayyan harakatlar ketma-ketligi.

**Jismoniy xususiyatlar** – tarkib, o'lchamlar, tugatish, shakl, moslik va ularning tegishli tolerantliklari kabi moddiy xususiyatlarning miqdoriy va sifatli ifodalari.

**Joylashtirish diagrammasi** – tizim elementlarining tarkibi va ulanishlarini ko'rsatish bilan birga, ularning ishlashi vaqtidagi jismoniy jihatdan hisoblash zaxiralarida qanday joylashganligini ko'rsatadi.

**Kamchilik** – bu talab tasniflari, dastlabki yoki loyiha tasniflari, hujjatlar va boshqalarning noto'g'ri tavsifi natijasida yuzaga keladigan noto'g'ri hodisa.

**Ketma-ketlik diagrammasi** – bu tizimning harakatini "misol orqali" tasvirlash usuli.

**Kodni tekshirish** – tizim loyihasida ishlatiladigan ma'lumotlar turlari va tuzilmalarining tavsiflarini ularning talablariga muvofiqligini rasmiy tekshirish.

**Konfiguratsiya** – tarkiblar va qism-tizimlarning alohida nusxalaridan tayyorlangan dasturiy ta'minot tizimining naqli.

**Konseptual loyiha** – tizimga qo'yiladigan talablarning tafsilotlarini tushunish va muvofiqlashtirishni aniqlashtirish.

**Konseptual modellashtirish** – bu muammoni inson tomonidan tushunishga qaratilgan modelni yaratish jarayoni.

**Korporativ korxonalarni boshqarish** – muammolarini vertikal (birlamchi berilganlardan tortib yuqori boshqaruv qarorla-

rini qo'llab-quvvatlashgacha) va gorizontal (faoliyatning barcha sohalari va texnologik amallarni) majmuaviy hal qilishni ta'minlash.

**Majburiy talablar** – har qanday yo'l bilan olinishi kerak bo'lgan talablar.

**Manfaatdor shaxs** – o'ziga yuklangan vazifani bajarishda tizim tomonidan ishlab chiqarilgan natijalardan (masalan, do'stona, raqobatbardosh yoki qarama-qarshi) manfaatdor shaxs yoki tashkilot.

**Mavzu** – tizim bilan o'zaro aloqada bo'lgan tizimdan tashqarida kimdir yoki biror narsa.

**Menejment** – xodimlar jamoasini professional boshqarish.

**Meros** – boshqa super sinf ob'yektlari tomonidan foydalanish uchun shahsiy xususiyatlarning kichik sinfida aniqlashtirish.

**Metrika** – miqdoriy o'lchov va dastur xususiyatlarini o'lchash shkalasi.

**Metrikani baholash elementi** – miqdoriy yoki sifat o'lchovini baholash tizimidagi og'irligini hisobga oluvchi ko'rsatkich.

**Mezon** – maqsadga erishish darajasini baholash va maqsadga erishish vositalarini (usullarini, texnologiyalarini) tanlashning hal qiluvchi qoidalarini shakllantirishga imkon beradigan tizim holatining miqdoriy yoki sifat tavsifi.

**Moslashish diagrammasi** – bu ketma-ketlik diagrammasining maxsus shakli bo'lib, unda sinflovchi nusxalarining turli holatlardagi o'zgarishlari va ularning vaqt bo'yicha moslashishiga alohida e'tibor beriladi.

**Moslik** – ob'yektning boshqa element bilan belgilangan chegaralar to'plamida osonlik bilan va hech qanday aralashuvsiz ulanishi uchun mosligi.

**Muayyan natija** – tizimdan manfaatdor tomonlar (foydalanuvchilar, hissadorlar, egalar va boshqalar) tizimlardan natijalar kutishi.

**Muhandislik** – bu to'qish, tajriba va amaliyot natijasida olingan matematika va tabiiy fanlar bo'yicha bilimlarni insoniyat manfaati uchun tabiat materiallari va kuchlaridan tejamkorlik bilan foydalanish yo'llarini ishlab chiqish uchun oqilona qo'llanadigan kasb.

**Muvaffaqiyat ehtimoli** – ma'lum bir natijaga erishish noaniqlik yoki xavf darajasini o'z ichiga olishi.

**Muvaffaqiyatsizlik darajasi** – bu dasturiy ta'minot tizimida uni sinovdan o'tkazish yoki ishlatish paytida nosozliklar yoki nuqsonlarning paydo bo'lish chastotasi.

**Nazorat yoki bosqichma-bosqich punkt** – bu to'g'ri qaror qabul qilish mezonlari to'plami bajarilmaguncha, ish jarayonining borishini keyingi maqsadli amallar to'plamiga o'tkazishni cheklab qo'yadigan asosiy qaror nuqtasi.

**O'zaro faoliyat yurituvchi elementlar** – tizim tuzilmasidagi elementlar shakli, mosligi va funksiyasi bo'yicha bir-biriga mos bo'lishi kerakligi.

**O'zaro harakat diagrammasi** – bu faoliyat diagrammasining kengaytirilgan sintaksisiga ega bo'lgan bir turi.

**Ob'yekt (object)** – bu noyoblik xossasiga ega bo'lgan va atrofidagi begona holat va xatti-harakatni qamrab olgan mohi-yatdir.

**Ob'yekt diagrammasi** – bu sinf diagrammasining nusxasi.

**Ob'yektga yo'naltirilgan model** – bir-biri bilan o'zaro ta'sir qiladigan, xususiyatlarga va xatti-harakatlarga ega bo'lgan ob'yektlar to'plamining tuzilishi.

**Ob'yektlarning o'zaro harakati** – bir-biriga xabarlarni yuborish orqali ob'yektlar o'rasidagi aloqa.

**Ontologiya** – ishlab chiqilishi kerak bo'lgan muammo muhitida ularni talqin qilish uchun elementar tushunchalar, atamalar va paradigmalarni to'plami.

**Operatsion** – tayyor dasturiy ta'minot tizimini bajarish bo'yicha harakatlar.

**Oq quti usuli** – boshqaruvning barcha yo'llari va oqimlarini to'liq sinovdan o'tkazish orqali xatolarni aniqlash uchun dasturning ichki tuzilishini o'rganish.

**Paket (package)** – modeldagi elementlar guruhi (shu jumladan paketlar).

**Paketlar diagrammasi** – bu modelning murakkabligini boshqarishga imkon beruvchi yagona vosita.

**Pog'onali (waterfall) model** – dasturiy ta'minotni ishlab chiqish jarayonining modeli bo'lib, unda ishlab chiqish jarayoni talablarini tahlil qilish, loyihalash, amalga oshirish, sinovdan

o'tkazish, integratsiya va qo'llab-quvvatlash bosqichlaridan ketma-ket o'tadigan oqimiga o'xshashi.

**Pog'onali-takroriy yondoshuv** – istalgan natijaga erishilgunga qadar har bir jarayonning ketma-ket takrorlanishini ta'minlash.

**Pretsedent** – ma'lum bir mavzu uchun qimmatli natijani beradigan tizim tomonidan bajariladigan harakatlar ketma-ketligi.

**Pretsedent sinf** – pretsedent misollar to'plami.

**Qanday bo'lishi kerak (TO-BE) modeli** – ATni joriy etishni hisobga olgan holda tijorat jarayonlardagi zaruriy o'zgarishlarni aks ettirish.

**Qayta foydalanish** – dasturiy ta'minot tizimlarini amalga oshirish jarayonida olingan har qanday rasmiylashtirilgan bilimlardan tayyor qism sifatida foydalanish.

**Qayta ishlatiladigan tarkib** – bu nafaqat ishlab chiquvchilar, balki foydalanuvchilar tomonidan yangi muhitga mos ravishda moslashgandan so'ng foydalanish uchun taqdim etilgan tizimni dasturlash bo'yicha o'tgan tajriba haqidagi bilimlarning bir qismi.

**Qo'llab-quvvatlash** – foydalanuvchiga foydalanish uchun topshirilgandan so'ng dasturiy ta'minot tizimiga o'zgartirishlar kiritish bo'yicha ishlar.

**Qo'llab-quvvatlash jarayoni** – tizim funksiyalarini bajarishda yoki tizimni foydalanishdan olib tashlashda tizimning o'zgarishi va dolzarbligini boshqarish bo'yicha harakatlar.

**Qo'shimcha qiymatli qayta ishlash** – amaliy xarajat, foydali-lik, yaroqlilik, mavjudlik va samaradorlik kabi omillar, tizimdagi har bir amal va masaladan kiruvchi berilganlarning kirishiga qiymat qo'shishini va tizim umumiy vazifasiga erishishiga hissa qo'shadigan natijalarni ishlab chiqarishni talab qilish.

**Qo'shimcha talablar** – bu nafaqat shahsiy funksiyalarga, balki butun tizim bilan bog'liq barcha turdagi cheklovlar va sifatlar.

**Qora quti usuli** – talablarning tasniflari asosida funksiyalarning haqiqiy xatti-harakatlarining kutilgan xatti-harakatlarga muvofiq-ligini tekshirish orqali amalga oshirilgan funksiyalarni sinash.

**Rad etish** – xatolar yoki nuqsonlar tufayli dasturni ish holatidan ishlamaydigan holatga o'tkazish.

**Samaradorlik mezon** – bu turli xildagi zaxiralarning sarfxarajatlarini hisobga olgan holda maqsadga erishish darajasini baholashga imkon beradigan mezon.

**Shakl** – bir yoki bir nechta interfeys chegarasi maqsadlarini qoʻllab-quvvatlash uchun moʻljallangan elementning belgilangan ifodasi.

**Sifat menejmenti** – dasturiy taʼminot sifatini rejalashtirish, boshqarish va baholash usullari va tizimli faoliyati majmuasi.

**Sifat modeli** – dasturiy taʼminotning xususiyatlari, atributlari, oʻlchovlari va baholash elementlarini aks ettiruvchi toʻrt darajali tuzilma.

**Sifat xususiyatlari** – funktsionallik, ishonchlilik, qulaylik, samaradorlik, hamrohlik, ixchamlik va boshqalar.

**Sifatni baholash** – dasturiy taʼminotning maqsadiga mos keladigan talablarga javob berish darajasini aniqlashga qaratilgan harakatlar.

**Sinergik tarzda ishlash** – elementlar termasini integratsiya qilishdan maqsad, alohida elementlar sifatida erishib boʻlmaydigan yuqori darajadagi qobiliyatni amalga oshirish uchun, alohida elementlarning imkoniyatlaridan foydalanish.

**Sinf (class)** – bu obʼyektlar toʻplamining umumiy holatini belgilovchi atributlar va xususiyatini aniqlovchi amallar tavsifidir.

**Sinf diagrammasi** – tizim tuzilishini tasvirlashning asosiy usul.

**Sinflovchi (Classifier)** – bu tuzilmaviy va xulqiy xususiyatlarini tavsiflovchi mexanizm, yaʼni oʻz nusxasiga ega boʻlgan model elementlari.

**Sinov** – bu dasturni semantik disk raskadrovka (tekshirish) usuli boʻlib, u turli xil test toʻplamlari ketma-ketligini bajarish va maʼlum natija bilan taqqoslashdan iborat.

**Sinov maʼlumotlari** – dasturiy taʼminot tizimining ishlashini tekshirish uchun dastur hujjatlari yoki tasniflari asosida tayyorlangan maʼlumotlar.

**Sinov rejasi** – shaxsiy tarkiblar va umuman tizimning strategiyasi, zaxiralari va sinov jadvalining tavsifi.

**Sotib olish jarayoni** – xaridor tomonidan dasturiy taʼminot tizi-

mi yoki xizmatini aniqlash uchun maʼlum bir tahlil siklini boshlaydigan harakatlar.

**Statik sinov** – tarliablarning texnik xususiyatlarini kompyuterda bajarmasdan toʻgʻri taqdim etish uchun tahlil qilish va koʻrib chiqish.

**Ssenariy** – bu pretsedent namunasining xatti-harakati va bajarilishini aks ettiruvchi muayyan harakatlar ketma-ketligi.

**Talab** – buyurtmachi va tizim ijrochisi oʻrtasida uning ishlashi toʻgʻrisida kelishuv yoki shartnoma.

**Tamoyil** – butun loyihalash va dasturlash sohasi asosida yotadigan asosiy tushunchalar.

**Taqlid** – dinamik xususiyatlarni tahlil qilish va resurslarni taqsimlash tahlili bilan tizimning xatti-harakatlarini turli jihatlarida va turli tashqi va ichki sharoitlarda modellashtirish.

**Tarkib** – bu tizimning modulli qismi boʻlib, talab qilinuvchi va taqdim etiluvchi interfeyslarning aniq termasidan iborat.

**Tarkib diagrammasi** – bu modellashtirilayotgan tizimni tashkil etuvchi sinflar va tarkiblarni, mantiqiy yoki fizik modullar oʻrasidagi munosabatlarni koʻrsatadi.

**Tarkibni ishlab chiqish** – kataloglarda saqlanadigan tayyor tarkiblar orqali dasturiy taʼminotni loyihalash.

**Tasdiqlash** – dasturiy taʼminot tizimini ishlab chiqish mijozning talablariga muvofiqligini tekshirish.

**Tasnif** – standartlar, sifat mezonlari va boshqalarni hisobga olgan holda algoritim, qoidalar, obʼyektlar harakatlarning qanday yaratilgani yoki ishlashining deklarativ tavsifidir.

**Tavsif** – har bir tizimni oʻziga xos tarzda aniqlaydigan xulq-atvor va jismoniy fazilatlarini anglatadi.

**Tekshirish** – tashkilotning tashkiliy tuzilmasi, uning faoliyati va mavjud axborotni qayta ishlash tizimini oʻrganish va tashxisli tahlil qilish, hayot siklining har bir bosqichida tizimning belgilangan talablarga muvofiqligini aniqlash.

**Texnik loyiha** – barcha tarkibiy elementlar va ularning kompozitsiyalarini aniqlash orqali tizimning ishlashi va rivojlanishi muhiti talablarini aks ettirish.

**Texnik topshiriq** – bu avtomatizasiyalashtirilgan texnologiya -

larni ishlab chiqish uchun zarur bo'lgan maqsadlar, talablar va asosiy kirish berilganlarini belgilaydigan hujjat.

**Tizim** – bu har biri aniq belgilangan va cheklangan imkoniyatlarga ega bo'lgan o'zaro ishlaydigan elementlarning integratsiyalashgan termasi bo'lib, foydalanuvchiga ma'lum natija va muvaffaqiyat ehtimoli bilan belgilangan amaliy muhitda missiyaga yo'naltirilgan tezkor ehtiyojlarni qondirishga imkon berish uchun qo'shimcha qiymatni qayta ishlashni amalga oshirish uchun sinergik tarzda ishlaydi.

**Tizim amallari lug'ati** – bu tizim ob'yektining amaliy aloqalari va muayyan bosqich va ish tartibotini qo'llab-quvvatlash uchun zarur bo'lgan o'zaro ta'sirlarni qamrab oladigan va tavsiflovchi hujjat.

**Tizim amallarining modeli** – bu ko'pgina tizimlar uchun ish jarayoni va amallarni aniqlash uchun boshlang'ich nuqta sifatida ishlatilishi mumkin bo'lgan tizim amallarining umumlashtirilishidan iborat tuzilma.

**Tizim foydalanuvchisi** – tizim, mahsulot yoki xizmatdan foydalanadigan jismoniy shaxs yoki tashkilot yoki missiyaga yo'naltirilgan maqsad yoki vazifani bajarish uchun ularning qo'shimcha mahsuloti.

**Tizimdagi missiyaning roli** – tizimdagi missiyaning rollari samaradorlikka asoslangan natija maqsadlariga erishadigan mahsulot va xizmatlarni yetkazib berish uchun maxsus vazifalar yuklangan tizimlar tomonidan amalga oshirish.

**Tizimdagi yordamchining roli** – tizimdagi yordamchining rollari, tizimdagi missiyalarning tayinlangan vazifalarni bajarishi va qo'llab-quvvatlashi, hamda keyingi texnik xizmat ko'rsatish va o'qitishni amalga oshirishga tezkor tayyor bo'lishini ta'minlashi uchun, boshqa tizimlar tomonidan bajarish.

**Tizimning ixchamligi** – modullarni yangi atrof-muhit yoki platforma sharoitlariga moslashtirish orqali tizim xizmatini o'zgartirish qobiliyati.

**Tizimning oxirgi foydalanuvchilari** – ehtiyojlari uchun kompyuter tizimi buyurtma qilingan professional shaxslar.

**Tizimning vazifasi** – aniq raqamli (shu jumladan vaqtinchalik)

xususiyatlarga ega bo'lgan maqsadni ko'rsatadigan maqsadga erishish usulini (texnologiyasini) tavsiflash.

**To'planning karraligi (multiplicity)** – bu sonlar to'plamidan iborat bo'lib, ular shu to'plam quvvatining mumkin bo'lgan barcha qiymatlarini belgilash.

**Tortishuv modeli** – empirik usulga asoslangan va ishlab chiqish murakkab muhitli sharoitlar uchun mo'ljallangan yuqori qiymatli mahsulotlarni yaratish.

**Tugun (node)** – bu artefaktlar joylashtiriladigan va kerak bo'lganda bajariladigan hisoblash zaxirasi.

**UML (Unified Modeling Language)** – bu tizimlarning dasturiy ta'minotini ishlab chiqish jarayonida yaratilgan barcha suniy dalillarni *tasniflash, namoyishlash, loyihalash va hujjatlashtirish* bo'yicha yaratilgan umumiy modellashtirish uchun grafik til.

**UML modeli** – bu cheklangan to'plamdagi til tuzilmalari bo'lib, uning asosiylari mohiyatlar va ular o'rtasidagi munosabatlar.

**Umumiy bilim** – domen oilasining barcha vazifalari to'g'risidagi bilimlar, ular ushbu oilaga tegishli har qanday muammoning yechimini ta'minlash uchun mos keladigan shaklda taqdim etish.

**Umumlashtirish** – bu ikkita mohiyatlar o'rtasidagi munosabatlardan iborat bo'lib, ulardan biri ikkinchisining xususiy (maxsus) holati.

**Uyushma** – bu mohiyatlar o'rtasidagi, munosabatlarning eng ko'p qo'llaniladigan turi. Eng umumiy va muhim munosabat bo'lib, ularning mazmuni va hajmini aniqlashtirmasdan mohiyatlar o'rtasida bog'liqlik mavjudligini tasdiqlaydi.

**Xato** – dastur operatorlarida yoki uni ishlab chiqishning texnologik jarayonidagi kamchiliklar, bu asl ma'lumotlarning noto'g'ri talqin qilinishiga va noto'g'ri yechimga olib kelishi.

**Xossalar** – tizimning o'lchamlik xossalarini bildirish.

**Yakuniy foydalanuvchi** – natija yoki natijalardan bevosita foyda ko'radigan shaxs yoki tashkilot tizim, mahsulot yoki xizmat.

**Yashirin funktsional talablar** – bu foydalanuvchi bilmagan holda, tizim tomonidan bajariladigan funktsiyalar.

## Foydalanilgan adabiyotlar

1. Charles S. Wasson. System Analysis, Design, and Development. Concepts, Principles, and Practices. Published by John Wiley & Sons, Inc., Hoboken, New Jersey, 2006, 832 p.
2. Halimov M.X., Gaynazarov S.M. Berilganlar bazasini boshqarish tizimlari. OO'U uchun darslik. O'zgartirilgan va to'ldirilgan 3- nashr. "Fan va texnologiyalar"// T. 2021, 672 b.
3. Raul Sidnei Wazlawick. Object-Oriented Analysis and Design for Information Systems. Modeling with UML, OCL, and IFML. Waltham, MA, 02451, USA, 2013, 470 p.
4. Буч Г., Рамбо Д., Якобсон И. Язык UML. Руководство пользователя. 2-е изд.: Пер. с англ. Мухин Н. – М.: ДМК Пресс, 2006, 496 с.
5. Вейцман В. М. Проектирование информационных систем. Электронная книга, Изд. 2, стер. 2021. 316 с. ISBN 978-5-8114-8572-7.
6. Гвоздева Т.В., Баллод Б. А., Проектирование информационных систем. Санкт-Петербург: Лань, 2018. — 156 с.
7. Грекул В.И., Денищенко Г.Н., Коровкина Н.Л. Проектирование информационных систем. Учебное пособие. Интернет-университет университет информационных технологий, М., 2005, 304 с.
8. Заботина Н.Н. Методы и средства проектирования информационных систем. М, ИНФРА – М, 2023, 331 с.
9. Иванов Д. Ю., Новиков Ф. А. Основы моделирования на UML: Учеб. пособие. СПб.: Изд-во Политехнического университета, 2010, 249 с.
10. Избачков Ю.С., Петров В.Н. Информационные системы: Учебник для вузов. 2-е изд. — СПб.: Питер, 2006, 656 с.
11. Крэг Ларман. Применение UML 2.0 и шаблонов проектирования. Практическое руководство. 3-е изд. Издательский дом "Вильямс" М.: 2013, 736 с.
12. Леоненков А. В. Самоучитель UML 2. – СПб.: БХВ-Петербург, 2007, 576 с.
13. Орлов, С.А. Технологии разработки программного обеспечения. Учебник, Питер, 2002. – 464 с
14. Основные принципы Scrum гибкой методологии разработки. Режим доступа. <http://wnfx.ru/osnovnyie-printsipyi-scrum-gibkoy-metodologii-razrabotki/>
15. Рочев И.В. Информационные технологии. Анализ и проектирование информационных систем. Учебное пособие. 2-е изд. испр., Москва-Санкт-Петербург: Ланг., 2019, 128 с.
16. Стасышин В.М. Проектирование информационных систем и баз данных. Новосибирск: Изд-во НГТУ, 2012, 100 с.
17. Терещенко П.В., Астапчук В.А. Методы и средства проектирования информационных систем. Новосибирск: Изд-во НГТУ, 2021. 85 с
18. Флегонтов А.В., Матюшичев И.Ю. Моделирование информационных систем. Unified Modeling Language. Учебное пособие для вузов, 3-е изд., доп., Санкт-Петербург: Лань, 2023. 140 с.
19. Чистякова В.И., Белов В.В. Проектирование информационных систем. Учебное пособие, М, Издательский центр «Академия», 2013, 352 с.
20. <http://argouml.tigris.org>
21. <http://argouml.tigris.org/servlets/ProjectMailingListList>
22. [http://www.iso.org/iso/catalogue\\_detail?csnumber=35733](http://www.iso.org/iso/catalogue_detail?csnumber=35733)

M.X. HAKIMOV

1.  
2.  
3.  
4. II  
5. Б'  
6.  
7.  
8.  
9.  
1

**AXBOROT TIZIMLARINI  
LOYIHALASH  
(TAMOYIL VA AMALIYOT)**

DARSLIK

Muharrir Z.N. Buranov

Bosishga ruxsat etildi 24.09.2024y. Bichimi 60X84 1/16.  
Bosma tabog'i 26,5. Shartli bosma tabog'i 26,5. Adadi 20 nusxa.  
Buyurtma №109. Bahosi kelishilgan narxda.  
"Ma'rifat" nashriyoti. Toshkent, Salorbo'yi kochasi, 35A.  
O'zbekiston Milliy universiteti bosmaxonasida bosildi.  
Toshkent, Talabalar shaharchasi, O'zMU.

ISBN 978-9910-665-46-2



9 789910 665462