

O'ZBEKISTON RESPUBLIKASI OLIY VA O'RTA MAXSUS TA'LIM  
VAZIRLIGI

SAMARQAND DAVLAT UNIVERSITETI

AMALIY MATEMATIKA VA INFORMATIKA FAKULTETI

OPTIMAL BOSQARUV USULLARI KAFEDRASI

Ro'yhatga olingan:

«TASDIQLAYMAN»  
O'quv ishlari bo'yicha prorektor  
Prof.A.S. Soleev

№ \_\_\_\_\_  
2019 y. « \_\_\_\_ » \_\_\_\_\_

\_\_\_\_\_  
« \_\_\_\_ » \_\_\_\_\_ 2019 y.

### JAVADA GRAFIK DASTURLASH

fani bo'yicha

### O'QUV-USLUBIY MAJMUA

**Bilim sohasi:** 100000 - Gumanitary soha

**Ta'lim sohasi:** 110000 - Pedagogika

**Ta'lim yo'nalishi:** 5110700 – Informatika va o'qitish metodikasi

Samarqand - 2019

S. Axatqulov. “Javada grafik dasturlash” fanidan o’quv-uslubiy majmua.

Samarqand: SamDU nashri -2019. 110bet.

Ushbu o’quv-uslubiy majmua oliy o’quv yurtlari amaliy matematika va informatika, 5110700 – Informatika o’qitish metodikas ta’lim yo’nalishi talabalriga mo’ljallangan

**Taqrizchi:** A. Abdullayev – Axborotlashtirish texnologiyalari kafedrasida dotsenti, t.f.n..

SamDU o’quv-uslubiy kengashining 2019 yil \_\_\_\_\_ dagi \_\_\_\_\_ qarori bilan o’quv-uslubiy majmua sifatida nashrga tavsiya etilgan

**Tuzuvchilar:** S. Axatqulov- Optimal boshqaruv usullari kafedrasida katta o'qituvchisi, PhD.

Mazkur o'quv –uslubiy majmua Samarqand Davlat universiteti, Amaliy matematika va informatika fakulteti, "Optimal boshqaruv usullari" kafedrasida, 5110700 – informatika o'qitish metodikasi ta'lim yo'nalishining o'quv rejasidagi "Javada grafik dasturlash" fani bo'yicha Samarqand davlat universiteti tomonidan tasdiqlangan namunaviy fan dasturi asosida ishlab chiqilgan

"Optimal boshqaruv usullari" kafedrasining 2019 yil \_\_\_\_\_ 1-son yig'ilishida muhokamadan o'tgan va fakultet kengashida muhokama qilish uchun tavsiya etilgan.

**Kafedra mudiri:** \_\_\_\_\_ dots. Bozorov I.N.

Amaliy matematika va informatika fakulteti o'quv-uslubiy kengashining 2019 yil "\_\_\_" – avgustdagi 1-son qarori bilan tasdiqlangan.

**O'quv-uslubiy kengash raisi:** \_\_\_\_\_ dots. Mamatov Sh.S.

Fanning ishchi o'quv dasturi Amaliy matematika va informatika fakulteti ilmiy kengashining 2019 yil "\_\_\_" – avgustdagi 1-son qarori bilan tasdiqlangan.

**Fakultet dekani:** \_\_\_\_\_ dots. Babayarov.A.

Kelishildi:

O'quv-uslubiy

boshqarma boshlig'i

\_\_\_\_\_ dots. Aliqulov B.S.



ЎЗБЕКИСТОН RESPUBLIKASI ОЛИЙ ВА ЎРТА МАХСУС ТАЪЛИМ ВАЗИРЛИГИ  
САМАРКАНД ДАВЛАТ УНИВЕРСИТЕТИ

ЎЗБЕКИСТОН  
САМАРКАНД ДАВЛАТ УНИВЕРСИТЕТИ  
ректори

ИШЧИ ЎҚУВ РЕЖА

Таълим йуналиши: 5110700 – Информатика  
Ўқитиш методикаси

Академик даража - БАКАЛАВР  
Ўқиш муддати - 4 йил  
Таълим шакли - кундузги

2019 й.

I. ЎҚУВ ЖАРАЁНИНИНГ ЖАДВАЛИ

Курс	Хафтаалар												Ўқув жараёни:				Тажури	Хатмисиз																			
													ЖАМИ	интулки																							
	Сентябрь	Октябрь	Ноябрь	Декабрь	Январь	Февраль	Март	Апрель	Май	Июнь	Июль	Август		Назорат таълими	Аттестация	Малака амалиёти			мулкисий																		
I					Т	Т	А	А							А	А	Т	Т	Т	Т	Т	Т	Т	43	39	4		9	52								
II					Т	Т	А	А								А	А	Т	Т	Т	Т	Т	Т	Т	43	39	4		9	52							
III					Т	Т	А	А				М	М							А	А	Т	Т	Т	Т	Т	43	37	4	2	9	52					
IV				М	М	М	М	Т	Т	А	А	М	М	М	М	М	М	М	М	А	Б	Б	Б	Б	Д	Д	Т	Т	Т	Т	43	21	5	12	5	5	48
	Жами																						172	136	17	14	5	32	204								

Назарий таълим   
  А   
  Аттестация   
  М   
  Малака амалиёти   
  Д   
  Давлат аттестацияси   
  Б   
  Битирув малакавий иши   
  Т   
  Таътил

II. ЎҚУВ РЕЖАСИ

Т/р	Ўқув блоклари, фаилар ва фаолит турларининг номлари	Талабанинг ўқув юкламаси (соатларда)									Соатларнинг курс, семестр ва хафтаалар бўйича тақсимоти							
		Умумий юкламанинг ҳажи		Аудитория машғулотлари (соатларда)							Мустақил таълим	1-курс	2-курс	3-курс	4-курс			
				Жами	Маъруза	Амалий	Лаборатория	Семинар	Курс лойиҳаси (иши)	Курс лойиҳаси (иши)		Курслардаги хафтаалар сони						
		соат	%	Жами	Маъруза	Амалий	Лаборатория	Семинар	Курс лойиҳаси (иши)	Курс лойиҳаси (иши)	1	2	3	4	5	6	7	8
		Семестрдаги аудитория машғулотлари хафтааларининг сони																
19	20	19	20	19	18	14	7											
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1.00	Гуманитар ва ижтимоий-иқтисодий фаилар	1584	21,6	1040	280	486		274		544	210	198	192	156	134	38	72	40
1.01	Ўзбекистон тарихи	116		76	38			38		40	76							
1.02	Ҳуқуқшунослик. Ўзбекистон Республикаси Конституцияси	116		60	30			30		56	60							
1.03	Фалсафа	148		96	48			48		52			48	48				
1.04	Маънавият асослари. Дипшунослик	92		58	28			30		34			58					
1.05	Маданиятшунослик	56		40	20			20		16				40				
1.06	Иқтисодиёт назарияси	116		58	28			30		58					58			
1.07	Социология	56		38	18			20		18				38				
1.08	Миллий гоъ асосий тушунча ва тамойиллар	60		42	20			22		18								42
1.09	Фуқаролик жамияти Ўзбекистонда демократик жамият куриш назарияси ва амалиёти	112		70	34			36		42								70
1.10	Ўзбек (рус) тили	116		78		78				38	78							
1.11	Чет тили	360		268		268				92	78	38	38	38	38	38		
1.12	Жисмоний маданият ва спорт *	186		116		116				70	78		38					
	Валеология асослари	50		40	16	24				10				40				
2.00	Математика ва табиий-илмий фаилар	776	10,6	458	206	192	60			318	154	118	38		74	74		
2.01	Математика	296		196	94	102				100	38	158						

3.10	Маълумотлар базаси	316		190	66	42	82			126				76	56	58			
3.11	Тармоқ технологиялари	170		100	30	24	46			70				100					
3.12	Компьютер таъминоти	442		270	86	68	116			172		114	116	40					
3.13	Танлов фаилари	154		98	30	22	46			56	40	20				38			
3.13.1	Масофавий таълим технологиялари	96		60	18	12	30			36	40	20							
3.13.2	Т.ф. Комбинаторика ва графлар назариясининг амалий талбиклари	58		38	12	10	16			20						38			
4.00	Ихтисослик фаилари	852	11,6	486	150	116	208	12	1кн	366	12			38	20	80	190	146	
4.01	Компьютерли моделлаштириш	180		112	32	40	40			68						32	44	36	
4.02	Педагогик дастурий воситалар	216		106	30	28	48			110						36	56	14	
4.03	Информатикашн ўқитиш технологиялари ва лойиҳалаштириш	152		92	40		40	12	кн	60						36	56		
4.04	Ахборот хавфсизлиги	96		58	18	18	22			38				38	20				
4.05	Танлов фаилари	208		118	30	30	58			90	12			24				42	40
	А)Компютер математикаси пакетлари																		
	Б)Информатикани ўқитишнинг замонавий технологиялари																	42	40
	С)Компютерларга техник хизмат кўрсатиш																		
	Мобил технологиялари	22		12	4	4	4			10	12								
	Интерактив дастурий воситалар яратиш технологиялари	44		24	6	6	12			20				24					
5.00	Кушимча фаилар	394	5,4	204	76	84	44			190	38		22			32	62	36	
	Амалий инглиз тили	78		42		42				36							28	14	
5.00.1	Компютерларга техник хизмат кўрсатиш	62		38	18	6	14			24	38								
5.00.2	Java дастурлаш тилига кириш	40		22	8	4	10			18			22						
5.00.3	Javaда график дастурлаш	72		32	16	10	6			40						32			
	Кушимча фан	116		56	28	18	10			60								34	22
	жами	7344	100	4352	1458	1334	1036	524	3кн	2992	608	640	608	640	608	576	448	224	
	Малака амалиёти	756														108	270	378	
	Битирув малакавий иши	270									108	108	108	108	108	108	108	162	
	Аттестациялар	918																	
	Жами	1944																	
	Хаммаси	9288																	

Ўқув режа Ўзбекистон Республикаси Олий ва ўрта махсус таълим вазирлигининг ракамли буйруғи билан тасдиқланган ўқув режа асосида тузилди.

йилдаги

Ўқув жараёнининг таркибий қисмлари	Хафталар сони	Семестр	Давлат аттестацияси
Назарий таълим	136	1-8	1. Гуманитар ва ижтимоий-иқтисодий фанлардан
Малака амалиёти	14	6,7,8	2. Чет тили
Аттестациялар	15+2(ДА)	1-8	3. Битирув малакавий ишини ҳимоя қилиш
Битирув малакавий иши	5	8	
Таътил	32	1 - 8	
Жами	204		

Самарқанд давлат университети  
Ўқув режаси кенгаши томонидан маъқулланди.  
2019 йил «АТ» \_\_\_\_\_ соғли баённома.

Самарқанд давлат университети  
Илмий кенгаши томонидан маъқулланди.  
2019 йил «АТ» \_\_\_\_\_ соғли баённома.

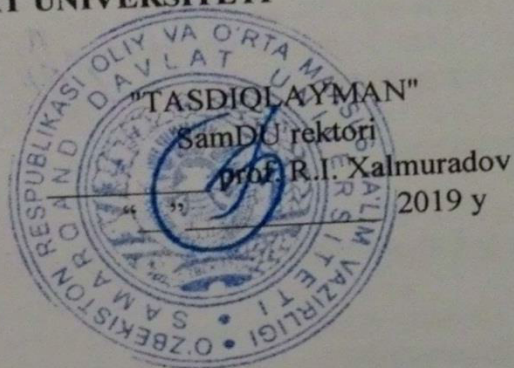
O'ZBEKISTON RESPUBLIKASI  
OLIV VA O'RTA MAXSUS TA'LIM VAZIRLIGI

SAMARQAND DAVLAT UNIVERSITETI

Ro'yxatga olindi:

№ BD-5110700-S.00.3

“ ” \_\_\_\_\_ 2019 y



JAVADA GRAFIK DASTURLASH

qo'shimcha fanining

O'QUV DASTURI

Bilim sohasi:

4100000 – Gumanitar soha

Ta'lim sohasi:

110000 – Pedagogika

Ta'lim yo'nalishi:

5110700 – Informatika o'qitish metodikasi

Samarqand – 2019

O'ZBEKISTON RESPUBLIKASI  
OLIV VA O'RTA MAXSUS TA'LIM VAZIRLIGI  
SAMARQAND DAVLAT UNIVERSITETI

Ro'yxatga olindi:

No 1355

2019 y " " "



JAVADA GAFIK DASTURLASH  
tanlov fanining  
ISHCHI O'QUVDASTURI

Bilim sohasi:

4100000 – Gumanitar soha

Ta'lim sohasi:

110000 – Pedagogika

Ta'lim yo'nalishi:

5110700 – Informatikao'qitish metodikasi

Samarqand – 2019

## MUNDARIJA

Mavzu 1. Dastur va Java dasturlash tili haqida umumiy tushunchalar . . . . .	9
Mavzu 2. JDK va Eclipse larni o'rnatish.. . . . .	11
Mavzu 3. Java dasturlash tilining asosi va uning strukturasi . . . . .	19
Mavzu 4. Swing haqida asosiy tushunchalar . . . . .	25
Mavzu 5. Ma'lumotlarni Panelda ko'rsatish. . . . .	31
Mavzu 6. Xodisa va amal. . . . .	34
<b>Amaliy mashg'ulot mavzulari. . . . .</b>	<b>40</b>
Mavzu 1. Checkbox va Radio tugmalar. . . . .	42
Mavzu 2. Ro'yxat va Combobox. . . . .	50
Mavzu 3. Slayder va Spinner. . . . .	53
Mavzu 4. Menyular. . . . .	59
Mavzu 5. Qurollar paneli. . . . .	63
Mavzu 6. Kontekst menyu. . . . .	65
Mavzu 7. Dialog oynalari . . . . .	68
Mavzu 8. JOptionPane va JFileChooser klasslari yordamida oynalar yaratish. . .	71
Mavzu 9. Joylashuv menedjerlari . . . . .	77
Mavzu 10. Xatolik va istisnolar . . . . .	83
Mavzu 11. Istisnolar bilan ishlash. . . . .	85
Mavzu 12. Klass a'zolari va metodlarni ko'rinishi. . . . .	87
Glossariy . . . . .	92
Adabiyotlar ro'xati. . . . .	110

# 1-Mavzu

## 1.1 Dastur va Java dasturlash tili haqida umumiy tushunchalar

**Java** dasturlash tili — eng yaxshi dasturlash tillaridan biri bo'lib unda korporativ darajadagi mahsulotlarni (dasturlarni) yaratish mumkin. Bu dasturlash tili **Oak** dasturlash tili asosida paydo bo'ldi. Oak (ma'nosi eman daraxti) dasturlash tili 90-yillarning boshida **Sun Microsystems** (hozirda **Oracle** nomidan ish yuritadi) tomonidan platformaga (operatsion tizimga) bog'liq bo'lmagan holda ishlovchi yangi avlod aqlli qurilmalarini yaratishni maqsad qilib harakat boshlagan edi. Bunga erishish uchun Sun hodimlari **C++** ni ishlatishni rejalashtirdilar, lekin ba'zi sabablarga ko'ra bu fikridan voz kechishdi. Oak muvofaqiyatsiz chiqdi va 1995-yilda Sun uning nomini **Java** ga almashtirdi, va uni WWW rivojlanishiga hizmat qilishi uchun ma'lum o'zgarishlar qilishdi.

Java 1990 yillarda ishlab chiqarila boshlangan bo'lsa ham, uning birinchi versiyasi (**Java 1.0**) 1996 yil ommaga taqdim etilgan. Undan so'ng keyingi versiyalar sekin-astalik bilan chiqa boshladi: **1998 yil — Java 2, 2004 yil — Java 5.0, 2006 yil — Java 6, 2011 yil — Java 7, 2014 yil — Java 8.**

Java **Obyektga Yo'naltirilgan Dasturlash** (OOP-object oriented programming, OOI) tili va u **C++** ga ancha o'xshash. Eng ko'p yo'l qo'yildigan xatolarga sabab bo'luvchi qismlari olib tashlanib, Java dasturlash tili ancha soddalashtirildi.

Java texnologiyasi o'ta sodda, xavfsizlikni yuqori darajada ta'minlab bera oladigan, kuchli, to'la obyektga yo'naltirilgan dasturlash tili bo'lib, muhit (platforma)ga bo'liq bo'lmagan holda ishlaydi. U bilan xatto eng kichik qurilmalarga ham dasturlar yozish mumkin. Java texnologiyasi to'laligicha **Java Virtual Machine(JVM)** ga asoslangan. JVM ning vazifasi tarjimonlik ya'ni, dastlab biz yozgan **\*.java** fayl kompilyator yordamida **bayt kodga** o'giriladi va JVM yordamida esa mashina tiliga aylantiriladi. Bu degani JVM qaysi platformaga tegishli bo'lsa, kodlarni ham o'sha platformaga moslab beradi.

### **JAVA imkonyatlari:**

— WORA — Write Once, Run Anywhere (portable). Platforma tanlamaydi;

— xavfsizlik (ishonch yuq kodni havfsiz ishga tushirish);

— xotirani havfsiz boshqarish (avtomat ravishda musorlarni yig'adi);

— tarmoq uchun dasturlar yozish ;

— ko'p oqimli (Multi-thread) dasturlash;

class lar alohida fayllarda saqlanadi. Kerak bo'lsa ishlatiladi.

## **1.2 Java imkonyatlari va texnologiyalari**

**Java SE (Java Standart Edition)** — serverda, shaxsiy kompyuterda desktoplarda ishlovchi dasturlar, appletlar yaratish uchun ishlatiladi. Bu texnologiya yordamida yaratilgan dasturlar deyarli barcha operatsion tizimlarda ishlay oladi (**Windows NT, Macintosh, Linux va Solaris**). Shu bilan birga JavaSE boshqa Java turlarining asosi hisoblanadi.

**Java EE (Java Enterprise Edition)** — Java texnologiyalari orasida eng keng tarqalgan turi xisoblanib unda asosan serverda ishlovchi dasturlar yaratiladi, masalan ko'p foydalanuvchili web-saytlar yaratishda keng qo'llaniladi va asosan internetda ishlovchi dasturlarda qo'llaniladi. Java SE ni Java EE dan eng asosiy farqi Java EE o'z tarkibiga Java SE ni olibgina qolmay shu bilan birga ko'pgina boshqa qo'shimcha kutubxonalarni(odatda\*.jar) ham o'z ichiga oladi ya'ni: **Servlet, JavaMail, JSF(Java Server Face)** va boshqa ko'pgina internetga asoslangan qoshimcha kutubxonalar.

**Java ME (Java Micro Edition)** — Java SE ning ba'zi qismlarini o'z ichiga oladi, JavaME yordamida kichik qurilmalar uchun dastrular yozish mumkin, masalan, mobil telefon uchun o'yinlar, dasturlar yaratish mumkin.

Javada kompilyator aytib o'tganimizdek biz yozgan kodni bayt-kodga o'giradi, odatda kompilatsiyadan o'tgan klasslar \*.class qisqartirmasi bilan tugaydi va

kompilatsiyadan o'tgan klassni Java Virtual Machine(JVM) ga yuklanadi va bayt-kodli fayllarni interpretatsiya qiladi, ya'ni mashina tiliga o'giradi va shu bilan birga undagi kodni imkoni boricha optimallashtiradi.

### **Mustahkamlash uchun savollar:**

1. Dastur va algoritm deganda nima tushuniladi?
2. Java dasturlash tili qanday xususiyatlarga ega?
3. Birlamchi kod va bayt kod orasidagi farq nimada?

## **2-Mavzu**

### **2.1 JDK va Eclipse larni o'rnatish.**

Java dasturlash tilida dastur tuzish uchun, dastlab, kompyuterga kerakli dasturlarni o'rnatish lozim. Birinchidan, Java dasturlarni ishga tushirish uchun, bizga **Java-mashina** kerak bo'ladi. Gap shundaki, barcha Java dasturlar faqat Java-mashina o'rnatilgan kompyuterlarda ishlaydi. Java-mashina Java dastur uchun muhit hisoblanadi. Ikkinchidan, Java dasturlarni yozish uchun maxsus muhit (**IDE**) kerak bo'ladi. Ikkala dasturni ham oraclening saytidan yuklab olishingiz mumkin. Yuklab olish uchun [Oracle](#) saytiga kiramiz va Java Platform (**JDK**) ni yuklab olamiz.

#### **Java SE Downloads**



Java Platform (JDK) 8u20



JDK 8u20 & NetBeans 8.0.1

JDKni yuklab oladigan oynaga o'tamiz va litsenziyaga rozi bo'lamiz(**Accept License Agreement**). Undan so'ng kompyuteringizga mos keladigan JDKni tanlaymiz, agar 32 bitlik tizim bo'lsa, «**jdk-8u20-windows-i586.exe**»ni tanlaymiz va yuklashni boshlaymiz.

Product / File Description	File Size	Download
Linux x86	135.24 MB	<a href="#">jdk-8u20-linux-i586.rpm</a>
Linux x86	154.87 MB	<a href="#">jdk-8u20-linux-i586.tar.gz</a>
Linux x64	135.6 MB	<a href="#">jdk-8u20-linux-x64.rpm</a>
Linux x64	153.42 MB	<a href="#">jdk-8u20-linux-x64.tar.gz</a>
Mac OS X x64	209.11 MB	<a href="#">jdk-8u20-macosx-x64.dmg</a>
Solaris SPARC 64-bit (SVR4 package)	137.02 MB	<a href="#">jdk-8u20-solaris-sparcv9.tar.Z</a>
Solaris SPARC 64-bit	97.09 MB	<a href="#">jdk-8u20-solaris-sparcv9.tar.gz</a>
Solaris x64 (SVR4 package)	137.16 MB	<a href="#">jdk-8u20-solaris-x64.tar.Z</a>
Solaris x64	94.22 MB	<a href="#">jdk-8u20-solaris-x64.tar.gz</a>
Windows x86	161.08 MB	<a href="#">jdk-8u20-windows-i586.exe</a>
Windows x64	173.08 MB	<a href="#">jdk-8u20-windows-x64.exe</a>

Yuklab olingandan so‘ng, dasturni ishga tushiramiz va berilgan savollarga qarab o‘rnatamiz(unchalik qiyin ish emas va ko‘p vaqt ham kerak emas).

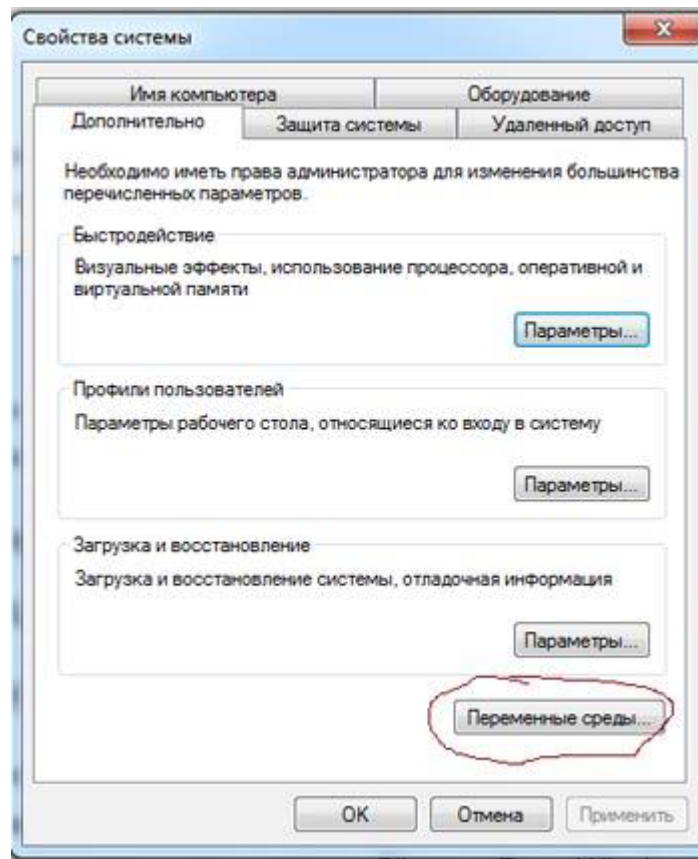
Odatda JDK «C:\Program files\Java» adresiga o‘rnatiladi.

Keling endi, java bajaruvchi utilitlarini topamiz. Uning uchun quyidagi papkaga kiramiz:

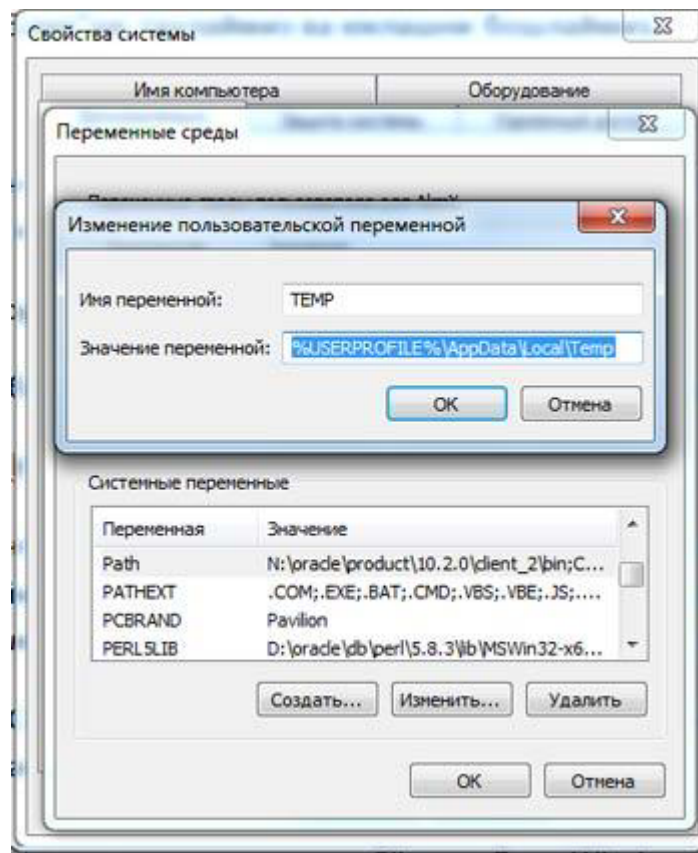
**C:\Program files\jdk\*\bin\**

Bu adresda JDKning barcha bajariluvchi fayllari joylashgan. Biz ularni, operatsion tizimga ham ma’lum qilib qo‘yishimiz lozim. Operatsion tizim bularni bilib olsa, bemalol «Команднаястрока» orqali ham ishlatishimiz mumkin bo‘ladi. Buni **Windows 7** misolida ko‘rsataman.

**Мойкомпьютер->Свойства,** chap tomonda «Дополнительныепараметрысистемы» bo‘limiga kiramiz va quyidagi oynani ko‘ramiz.



Bu oynadan «**Переменные среды**» tugmasini bosamiz, hosil bo‘lgan oynaning «**Системные переменные**» bo‘limidan «**PATH**» o‘zgaruvchisini(переменная)qidirib, uni belgilaymiz va «**Изменить**» tugmasini bosamiz.



Shundan so‘ng, «**Значение переменной**»dagi qiymatlarning oxiriga o‘tamiz va » **C:\Program Files\Java\jdk\*\bin**» adresini kiritamiz(\* o‘rniga o‘zingizning jdk adresingiz bo‘ladi, ya’ni mavjud papka nomi). «**OK**» tugmalarini bosib, bu oynalardan chiqib ketamiz va kompyuterni qayta yuklaymiz.

Hozircha shu, xulosa qiladigan bo‘lsak, biz bu maqolada, o‘zida Java kompilyatorni saqlaydigan, qaysiki biz yozgan Java kodlarni bayt kodga(kompyuter tushunadigan kodga) o‘tkazib beradigan, undan tashqari yozilgan kodlarni ishlatish uchun kerak bo‘ladigan Java-mashinani ham o‘zida mujassam etgan **JDK(Java Development Kit)**ni yuklashni, o‘rnatishni va operatsion tizimga o‘rnatilgan joyini e’lon qilishni ko‘rib chiqdik.

Oldingi maqolada JDK paketini o‘rnatishni ko‘rsatib o‘tdim, keling endi shu paketni ishlayotganini tekshirib olamiz. Uning uchun «**Командная строка**» oynasini ochamiz (**win+R->CMD**) va «**javac**» buyrug‘ini teramiz, so‘ng **ENTER** tugmasini bosamiz.

```

Администратор: Командная строка
Microsoft Windows [Version 6.1.7601]
(c) Корпорация Майкрософт (Microsoft Corp.), 2009. Все права защищены.
C:\Users\АкмХ>javac
Usage: javac <options> <source files>
where possible options include:
-g Generate all debugging info
-g:none Generate no debugging info
-g:<lines,vars,source> Generate only some debugging info
-nowarn Generate no warnings
-verbose Output messages about what the compiler is doing
-deprecation Output source locations where deprecated A
-classpath <path> Specify where to find user class files and
-cp <path> Specify where to find user class files and
-sourcepath <path> Specify where to find input source files
-bootclasspath <path> Override location of bootstrap class files
-extdirs <dirs> Override location of installed extensions
-endorseddirs <dirs> Override location of endorsed standards pa
-processor:<none,only> Control whether annotation processing and/
-processor <class1>[,<class2>,<class3>...] Names of the annotation pr
-processorpath <path> Specify where to find annotation processor
-d <directory> Specify where to place generated class fil
-s <directory> Specify where to place generated source fi
-implicit:<none,class> Specify whether or not to generate class f
-encoding <encoding> Specify character encoding used by source
-source <release> Provide source compatibility with specific
-target <release> Generate class files for specific VM versi
-version Version information
-help Print a synopsis of standard options
-Akey[=value] Options to pass to annotation processors
-X Print a synopsis of nonstandard options
-J<flag> Pass <flag> directly to the runtime system
-Werror Terminate compilation if warnings occur
@<filename> Read options and filenames from file
C:\Users\АкмХ>

```

**Javac** — bu java kompilyatoridir, qaysiki java kodni bayt kodga aylantirib beradi. Javac boshqa utilitlar kabi «bin» katalogida joylashgan bo‘ladi. Agar hamma narsa to‘g‘ri qilingan bo‘lsa, yuqoridagi rasmda keltirilgan yozuvlar chiqishi lozim, aks hold oldingi maqolani qaytadan o‘qib chiqing(agar «**Javac не является внутренней или внешней командной, исполняемой программой или пакетным файлом**» kabi xatolik chiqsa, «**Переменные среды**»ni tekshirib chiqing). Bu yozuvlar kompilyatorning kalit so‘zlari bo‘lib, bizga yordam tariqasida tushuntirib chiqilmoqda. Tekshirib oldik, endi Java tilida biror kod(dastur) yozib, «**Командная строка**»da ishlatib, natija olib ko‘ramiz. Dastur yozish uchun, istalgan yozuv muhitidan foydalanishimiz mumkin, masalan «**Блокнот**». «**Блокнот**»ni ochamiz va ichiga quyidagi kodlarni yozamiz:

```
1 class test_java{
```

```
2 public static void main(String[ ] args){
3 System.out.println("Hello World");
4 }
5 }
```

Bu bizning birinchi kodimiz. Bunda konsolda ekranga chiqarish komandasi(**System.out.println**) ishlatilmoqda. Dastur ekranga «**Hello World**» yozuvini chiqaradi. Bu kodni chuqur muhokama qilib o'tirmaymiz, hozir bu dastur ekranga yozuv chiqarishini bilsak bo'ldi. Javada dasturlar klaslardan tashkil topgan bo'ladi, misolda, klas nomi «**test\_java**»dir. Yozilgan kodni **test\_java.java** deb, biror joyga masalan, «**D:\test**» ichiga saqlaymiz. Bu yerda eng muhim jihat shuki, asosiy klas nomi bilan, fayl nomi bir xil bo'lishi lozim, registr(katta-kichik harflar)larga ham ahamiyat berish shart.

Bu kod oddiy yozuv bo'lib, hali dastur ko'rinishiga kelmadi, protsessor ham bu yozuvni tushunmaydi. Bu kodni bayt kodga kompilyatsiya qilgandan so'ng, kod, dastur ko'rinishiga keladi va kompyuter uni tushunadi. JDK papkasida Javac nomli kompilyator mavjud bo'lib, kodni kompyuter tushunadigan kodga o'zgartirib beradi. Uning uchun «**Командная строка**»ni ochib, «**test\_java.java**» faylimiz turgan adresga o'tib olamiz, so'ng «javac va kod yozilgan faylni(test\_java) ko'rsatamiz».

```
C:\> cd D:\test
```

```
D:\test> javac test_java.javac
```

Shundan so'ng, fayl joylashgan papkada yangi «**java\_test.class**» nomli fayl paydo bo'ladi. Mana shu fayl dastur hisoblanadi. Oldin aytib o'tganimizdek, java dasturini ishga tushirish uchun, java-mashina kerak bo'ladi, bu ham «**bin**» papkada joylashgan bo'lib, «**java**» deb nomlanadi. Yaratilgan dasturni ishga tushirish uchun, «java» va fayl nomini(kengaytmasiz) yozib, **ENTER**ni bosamiz(paydo bo'lgan fayl adresiga o'tib olamiz).

```
D:\test>java test_java
Hello World
D:\test>_
```

Biz yozgan yozuv ekranga chiqdi. Bu maqolada yozilgan kodni bayt kodga o'tkazish va java-mashina yordamida dasturni ishga tushirishni «**Командная строка**»da ko'rsatib o'tdim, lekin biz kodlarni bu muhitda yozmaymiz, chunki dasturchilar uchun qulay bo'lishi uchun, juda ko'plab vizual muhitlar yaratilgan. Keyingi maqolalarda ulardan birini o'rnatishni ko'rsatib o'taman.

## **2.2 Eclipse ni o'rnatish**

[www.oracle.com](http://www.oracle.com) web saytidan Eclipse IDE faylini kompyuterlarimizga yuklab olamiz. Eslatib o'tamiz, Eclipse IDE fayli yuqoridagi web saytda bepul holda mavjud va biz o'zimizning kompyuterimizning operatsion sistemasiga mos bo'lgan eclipse

faylni yuklab olishimiz kerak, yani operatsion sistemamiz 32 bitlikmi yoki 64.

The screenshot shows the Eclipse website's download page. At the top, there is a navigation bar with the Eclipse logo, a search bar, and a prominent orange 'DOWNLOAD' button. Below the navigation bar, a dark blue banner indicates the current path: 'HOME / DOWNLOADS'. A secondary navigation bar lists 'Packages', 'Developer Builds', and 'Java™ 8 Support'. The main content area features a dropdown menu for 'Eclipse Luna (4.4.1) Release for Windows'. The primary product card is for 'Eclipse IDE for Java Developers', which is 154 MB and has been downloaded 2,995,692 times. It lists features like Java IDE, CVS client, Git client, XML Editor, Mylyn, and Maven integration. To the right of this card, there are options for 'Windows 32 Bit' and 'Windows 64 Bit'. Below the main card is a 'Package Solutions' section with a search bar and a list of categories including BIRT Framework, C/C++ Development Tools, Code Recommenders Tools For Java Developers, Data Tools Platform, Git Team Provider, Java Development Tools, Java EE Developer Tools, JavaScript Development Tools, Jubula Functional Testing, and Maven Integration For Eclipse. On the right side of the page, there is a promotional banner for 'INTRODUCING F-TYPE for ECLIPSE BIRT APPS' with a 'Download Now' button. Below that is a 'RELATED LINKS' section with links to 'Compare & Combine Packages', 'Install Guide', 'Documentation', 'Updating Eclipse', and 'Forums'. At the bottom right, there is a 'MORE DOWNLOADS' section.

The screenshot shows the Eclipse IDE download page. At the top, there is a list of plugins including Maven Integration For Eclipse, Mylyn Task List, PHP Development Tools (PDT), Parallel Tools Platform, Plug-in Development Environment, Remote Application Platform, Remote System Explorer, Scout, WindowBuilder Core, XML Editors And Tools, Xtend, and Xtext. To the right, there is a 'Hint' section stating: 'You will need a Java runtime environment (JRE) to use Eclipse (Java SE 7 or greater is recommended). All downloads are provided under the terms and conditions of the Eclipse Foundation Software User Agreement unless otherwise specified.' Below the plugin list, there are three main IDE options:

- Eclipse IDE for Java EE Developers** (254 MB, Downloaded 1,796,312 Times): Tools for Java developers creating Java EE and Web applications, including a Java IDE, tools for Java EE, JPA, JSF, Mylyn... Available for Windows 32 Bit and Windows 64 Bit.
- Free Enterprise-Grade BIRT Report Server (Not a Trial)**: Instantly enhance your BIRT reports and web applications with interactive viewing, flexible scheduling, native Excel export, and more... Promoted Download.
- Eclipse IDE for C/C++ Developers** (164 MB, Downloaded 565,744 Times): An IDE for C/C++ developers with full integration. Available for Windows 32 Bit and Windows 64 Bit.

Yuklab olgan faylimiz .zip fayl bo'lin o'z ichida bir qancha fayllarni saqlaydi qaysiki ayrimlari popkalarda joylashgan

## 3-Mavzu

### 3.1 Java dasturlash tilining asosi va uning strukturasi

Navbat, Java dasturlash tilining asosi va uning strukturasi keldi. Barcha narsaning asosini bilib olsangiz, har qanday murakkab holni ham tushunish mumkin bo'ladi.

Nafaqat dasturlash, balki istalgan yo'nalish, soha yoki biror bir tushunchani

mukammal bilish uchun, dastlab uning strukturasi (nimalardan tashkil topganini) bilish, o'rganishning asosiy va to'g'ri harakati bo'ladi. Java dasturlash tilida dastlabki dasturimizni NetBeans IDE muhitida yaratdik, keling shu dasturimizni esga olaylik.

```
1 class test {  
2     public static void main(String[ ] args){  
3         System.out.println("Hello World");  
4     }  
5 }
```

Shu misol asosida, Java tili haqida, dastlabki bilimlarni egallaymiz.

**1.** Java dasturlash tili registrga bog'liq bo'lgan dasturlash tili hisoblanadi. Agar siz misoldagi «**main**» so'zini, «**mAin**» deb yozsangiz, NetBeansning quyi qismida xatolik haqida ma'lumot chiqadi. Demak, so'zlarni katta-kichikligiga ahamiyat bering.

**2.** Ikkinchi qatordagi «**public**» kalit so'zi, ruxsatni anglatadi, ya'ni dasturning istalgan joyidan shu nomli klassga murojaat qilish mumkin bo'ladi. To'liqroq keyingi maqolalarda.

**3.** «**class**» kalit so'zi, bu degani, dasturdagi hamma kodlar klass ichida joylashadi va bu klass «**test**» deb nomlanadi. **Klass** bu — dastur uchun konteyner hisoblanadi. Barcha Java dasturlarida kamida bitta klass mavjud bo'ladi, murakkab dasturlarda bittadan ko'p bo'lishi ham mumkin. Klass nomi dasturchi tomonidan quyidagi qoidalarga rioya qilgan holda beriladi:

- klass nomi harfdan boshlanishi lozim;
- klass nomida faqat raqam va harflar ishtirok etishi mumkin;
- uzunligi siz hohlaganingizcha bo‘lishi mumkin(kichikroq va tushunarli bo‘gani ma’qul);
- klass nomi sifatida maxsus zaxira so‘zlari ishtirok etmasligi lozim;
- odatda klass nomi katta harfdan iborat bo‘ladi(bizning holda bunday emas), agar bir necha so‘zdan iborat nom bo‘lsa, ularning bosh harflari katta harfda yozilishi(tuya stili) odat bo‘lib qolgan(**TestVariable**).

**4.** Klass nomi, shu klass saqlangan fayl nomi bir xil bo‘lishi lozim. Agar asosiy klass nomi «**test**» bo‘lsa, kod saqlangan fayl nomi ham «**test**» bo‘lishi lozim, shu bilan birga registr(katta-kichik harf) ham bir xil bo‘lishi shart. Fayl kengaytmasi «**.java**» bo‘lishi kerak.

**5.** Agar dastur to‘g‘ri yozilgan va to‘g‘ri nomlangan bo‘lsa, kompilyatsiyadan so‘ng, bayt-kod hosil bo‘ladi va u asosiy klass nomi kabi nomlanadi, lekin kengaytmasi «**\*.class**», ya’ni «**test.class**» ko‘rinishida bo‘ladi.

**6.** Yozilgan kod «**Javac**» kompilyatori yordamida bayt-kodga aylantiriladi va «**Java**» interpretatoridan foydalanib, dastur ishga tushadi. Interpretator o‘z ishini «**main**» metodidan boshlaydi, shuning uchun har bir kodda bu metod bo‘lishi lozim.

**7.** Koddagi figurali qavslar({, }) alohida blokka olish uchun ishlatiladi. Barcha dasturlar alohida blok hisoblanadi, shuning uchun klass nomidan keyin qavs ochiladi va oxirida yopiladi.

8. Dastur ichidagi alohida bloklar, dastur metodlari hisoblanadi. Misolda, «**main**» metodi dastur ichida joylashgan va qavs bilan ajratilgan. Bu qavslar, metod boshlanishi va tugashini bildiradi.

9. Metod ichida ekranga chiqaruvchi buyruq(**System.out.println**) berilgan, bu **operator** deyiladi. Bizning dasturda operatorlar soni bitta va oxirida nuqta-vergul(;) bilan tugatilmoqda.

Java dasturlash tilida, boshqa tillarda bo'lgani singari **kommentariyalar** mavjud. Kommentariyalar kompilyator tomonidan inobatga olinmaydi, ulardan dasturchilar o'z kodlarini tushuntirishda foydalanishadi. Kommentariyadan foydalanilgan kodlarni, keyinchalik ochib ko'rganda, nima vazifa bajarilganini tez anglab olish mumkin.

Kommentariyaning quyidagi turlari mavjud:

1. Bir qatorli kommentariya(**//**). Faqat bir qatorni kommentariyaga oladi, misol

```
1 class test { //bu yerda klass e'lon qilinmoqda.  
2 public static void main(String[ ] args){  
3 System.out.println("Hello World");  
4 }  
5 }
```

Berilgan izoh kompilyator tomonidan qaralmaydi, shuning uchun xatolik bo'lmaydi.

2. Ko‘p qatorli kommentariya. «/\*» belgilar bilan boshlanib, «\*/» belgilar bilan tugaydigan kommentariya turi. Bunda, shu blok orasidagi barcha yozuv kommentariyaga o‘tadi. Misol,

```
1 class test {  
2     /* public static void main(String[ ] args){  
3     System.out.println("Hello World");  
4     } */  
5 }
```

Kompilyator klass ichidagi barcha kodlarni inobatga olmaydi.

Java dasturlash tili, tiplarga qat’iy e’tibor beradigan til hisoblanadi, ya’ni dasturda ishlatiladigan barcha ma’lumotlar uchun tip e’lon qilinishi lozim.

Java dasturlashda, ma’lumotlar uchun 8 turdagi oddiy tiplar mavjud, keling ular bilan tanishib chiqamiz.

**1. Butun sonlar.** Bu tipdan, dasturda butun sondagi ma’lumotlarni ishlatish uchun foydalaniladi. Bu tipdagi qiymat **musbat** va **manfiy** bo‘lishi mumkin. Java dasturlash tilida, quyidagi 4 ta butun sonli tiplar mavjud:

Тип	Хотира ҳажми(байт)	Оралиқ қиймат
int	4	-2147483648 дан 2147483647 гача
short	2	-32768 дан 32767 гача
long	8	-9 223 372 036 854 775 808 дан 9 223 372 036 854 775 807 гача
byte	1	-128 дан 127 гача

Dasturda qo‘yilgan vazifa asosida, kerakli butun tiplardan biri tanlanadi. Kichik butun sonlar uchun «**long**» tipini ishlatishdan ma’no yo‘q. Agar katta hajmdagi projekt bo‘lsa, tezkor xotiradan ancha joy faqat tiplarning o‘ziga ketib qoldi. Misol uchun, temperaturani aniqlash uchun, hattoki «**short**» tipi ham kerak emas, «**byte**» tipi bilan ham qo‘yilgan vazifani bajarish mumkin.

**2. Xaqiqiy tip.** Dastur yozish mobaynida, haqiqiy tiplar kerak bo‘lib qolsa, ya’ni butun va kasr qismdan iborat sonlar ishlatilsa, bu tipdan foydalanish mumkin. Butun tiplar kabi, xaqiqiy tiplarning ham bir necha turlari mavjud:

Тип	Хотира ҳажми(байт)	Оралиқ қиймат
float	4	3.4e-038 дан 3.4e+038 гача
Double	8	1.7-308 дан 1.7e+308 гача

«**float**» tipi ba’zi protsessorlarda tezroq ishlaydi, kichikroq aniqlikdagi qiymatlar uchun ishlatiladi, agar qiymat juda kichik yoki juda katta bo‘lsa bu tipni ishlatib bo‘lmaydi. Bunday holda «**double**» tipidan foydalaniladi. Matematik amallarni bajarishga mo‘ljallangan protsessorlarda, bu tip hattoki «**float**» dan ham tez ishlaydi. Shunday ekan, har doimgidek, qo‘yilgan vazifaga qarab tip tanlanaveradi.

**3. Simvulli tip — char.** Bu tip simvollarni saqlash uchun ishlatiladi. Misol uchun, sonlar, raqamlar, belgilar va hakazo. Java tilida simvollarni saqlash uchun **Unicode** kodirovkasidan foydalaniladi. Bu kodirovkada, barcha simvollar raqamlar ko‘rinishida bo‘ladi. Misol uchun, «**A**» simoli unicodeda **0410** raqami bilan belgilangan. Demak, barcha simvollarni(barcha davlat tillaridagi harflarni ham) raqam orqali ifodalash mumkin ekan.

Bu tip tezkor xotiradan **2** bayt joy oladi va uning oraliq qiymati **0** dan **65536** gachadir. Dastur tuzishda simvollar uchun, shu tipning o'zi kifoya qiladi.

**4. Logik tip — boolean.** Bu tip 2 xil qiymatni qabul qilishi mumkin: **true**, **false**. Juda kerakli va ahamiyatli tip, odatiy qiymati «**false**» hisoblanadi. Yuqoridagi tiplar Java dasturlash tilining asosiy tiplari hisoblanadi.

### **Mustahkamlash uchun savollar:**

1. Java Dasturlash Vositasini o'rnatilish tartibi xaqida gapirib bering.
2. Windows operatsion tizimining *path* muhit o'zgaruvchisi nimani belgilaydi?
3. Java dasturlash tilida dastur yozish uchun qanday muhitlar mavjud?
4. "Salom dunyo" matnini konsolga chiqarib beruvchi dastur tuzing.
5. Birlamchi kod fayli qanday kengaytmaga ega bo'lishi kerak?
6. Bayt kod fayli qanday kengaytmaga ega?

## **4-Mavzu**

### **4.1 Swing haqida asosiy tushunchalar**

Swing tushunchasi javaning eski avlodlarida mavjud bo'lmagan. Unig o'rniga ma'lum kamchiliklarga ega bo'lgan AWT(Abstract Window Toolkit) qo'llanilgan. AWTlar foydali ammo ma'lum kamchiliklarga ega ya'ni cheklangan grafik interfeysga ega. Javaning ommalashishi bilan ko'p o'tmay bunday kamchilikka boshqa yondoshuv zarurligi ayon bo'ldi. Yechim esa swing bo'lib chiqdi. 1977 yilda e'lon qilingan swing JFS(Java Foundation Classes) qisimini ichiga o'rnatilgan. Swing komponenta va konteynerlarni o'z ichiga oladi. Umuman olganda , java komponentalari JComponent klasidan voris qilib olingan.

**Java/SWING nima** – JAVA uchun GUI Framework

- JAVA dasturlarini "look and feel" holatiga keltirish
- Java Foundation Classes (Sun Microsystems) bir qismi

- IFC (Netscape) va JFC (Sun Microsystems) qo' shilgani
- Java Standard Edition 1.2 versiyasidan boshlab paket shaklida qo' shilgan
- GUI controllar java dastur orqali generatsiya qilinadi
- Ko'rinishi va funkcionallkin jihatidan Windows, Mac va Linux OS'laridaka 8

### Swing komponentalari

JApplet	JButton	JCheckBox	JCheckBoxMenuItem
JColorChooser	JComboBox	JComponent	JDesktopPane
JDialog	JEditorPane	JFileChooser	JFormattedTextField
JFrame	JInternalFrame	JLabel	JLayeredPane
JList	JMenu	JMenuBar	JMenuItem
JOptionPane	JPanel	JPasswordField	JPopupMenu
JProgressBar	JRadioButton	JRadioButtonMenuItem	JRootPane
JScrollBar	JScrollPane	JSeparator	JSlider
JSpinner	JSplitPane	JTabbedPane	JTable
JTextArea	JTextField	JTextPane	JToggleButton
JToolBar	JToolTip	JTree	JViewport
JWindow			

## SWING Layots

AWT va Swing klaslarining layout manager'lari:

BorderLayout

BoxLayout

CardLayout

FlowLayout

GridBagLayout

GridLayout

GroupLayout

SpringLayout

## **Java/SWING BorderLayout**

### **BorderLayout**

Har bir content pane boshlang'ichda BorderLayout inisalizatsiya qilinadi. BorderLayout componentlarni 5 ta maydonga qo'yadi: top, bottom, left, right, va center.

Swing bu Java dasturlash tilining grafikli foydalanuvchi interfeysi qurollar to'plamidir. `javax.swing` o'rami interfeys elementlari xisoblanadigan oynalar, tugmalar, combo qutilari, katalog daraxtlari, jadvallar, menyular va zamonaviy grafikli dastur tuzish uchun kerak bo'ladigan boshqa elementlarni o'z ichiga oladi.

Swing qurollar to'plamidan oldin grafikli foydalanuvchi interfeysini yaratish uchun Java dasturlash tilida Abstract Window Toolkit (AWT) nomli qurollar to'plami ishlatilgan. AWT qurollar to'plami grafikli foydalanuvchi elementlarini yaratishni dastur mo'ljallanayotgan platformaning (Windows, Macintosh, Solaris) grafikli foydalanuvchi interfeysi qurollar to'plamiga yo'naltirgan. Natijada xar bir platforma uchun dastur ko'rinishi ushbu platformaga xos bo'lgan. Shu orqali Java dasturlarni portativligi ta'minlangan.

AWT qurollar to'plami qo'llaydigan printsipni o'ziga xos kamchiliklari xam mavjud. Ushbu qurollar to'plami grafikli foydalanuvchi interfeys elementlarini yaratishni dastur bajarilayotgan platformaga yo'naltirayotganligi sababli xar xil platformalarda interfeys elementlari bajarayotgan amallari farq qiladi. Bundan tashqari, xamma platformalar xam bir xil grafikli interfeys elementlar to'plamiga ega emas. Shu va boshqa kamchiliklarni oldini olish maqsadida grafikli foydalanuvchi interfeys elementlari mustaqil xosil qilish bo'lib ushbu printsip Swing qurollar to'plamida qo'llanilgan.

Swing qurollar to'plami yordamida dasturning foydalanuvchi interfeysini yaratishda oldindan tayyorlangan komponentlardan foydalaniladi. Ushbu komponentalar maxsus konteynerlar ichiga joylashtiriladi. Konteynerlar ichida komponentlarni tartibli

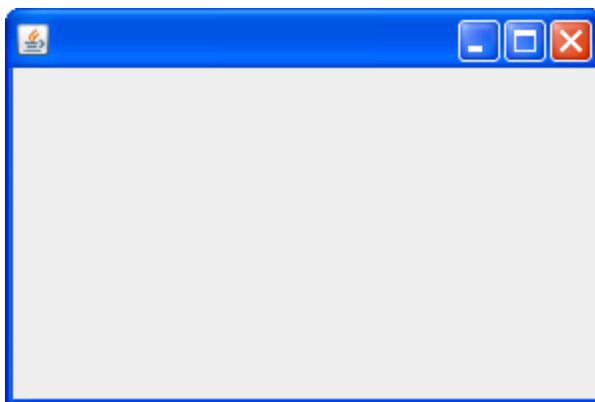
joylashtirish uchun joylashuv menedjerlar ishlatiladi. Joylashuv menedjerlari foydalanuvchi interfeys elementlarini joylashish strategiyasini belgilab beradi.

## 4.2 Frame yaratish va joylashtirish

Frame Java dasturlash tilining yuqori darajali oynasi xisoblanadi. Ushbu oyna barcha grafikli foydalanuvchi interfeysi elementlarini o'z ichiga oladi. Frame Swing o'ramining JFrame klassi yordamida yaratiladi. Frame bilan ishlashni quyidagi misol orqali ko'rib chiqamiz.

```
import javax.swing.*;
public class SoddaFrameTest{
public static void main(String[] args){
JFrame frame = new JFrame();
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.setSize(300, 200);
frame.setVisible(true);
}}
```

Yuqoridagi dastur ekranga bo'sh oynani chiqarib beradi.



*JFrame yordamida yaratilgan bo'sh oyna*

Ushbu dasturni xar bir qatorini ko'rib chiqamiz.

Birinchi qatorda `Swing` o'rami `import javax.swing.*;` ifodasi orqali import qilingan. Keyingi qatorlarda `SoddaFrame` klassi va uning `main()` metodi e'lon qilingan. Oltinchi qatorda oynani xosil qiluvchi `frame` obyektini `Frame` klassi andozasi asosida yaratilgan. Yettinchi qatorda foydalanuvchi ushbu oynani yopganda qanday amal bajarilishi `frame` obyektini `setDefaultCloseOperation()` metodi orqali berilgan. Bizning misolda dasturdan chiqib ketish amali berilgan. Sakkizinchi qatorda oynani yangi o'lchamlari `frame` obyektini `setSize()` metodi orqali berilgan. Yangi `frame` obyektini yaratilganda oyna o'lchami `0x0` piksel bo'ladi. Yangi oyna yaratilganda u avtomatik ravishda ekranda chiqarilmaydi. Buning uchun `frame` obyektini `setVisible()` metodini chaqirish kerak bo'ladi.

`JFrame` klassi aksariyat metodlarini o'zining superklasslaridan meros qilib oladi.

Asosiy metodlari quyidagilardan iborat:

- `dispose()` – oynani yopadi va u egallab turgan sistema resurslarini bo'shatadi;
- `setIconImage()` – oynaning sarlavxa qismiga yorliq chiqarib beradi;
- `setTitle()` – sarlavxa qismidagi matnni o'zgartiradi;
- `setResizable()` – oyna o'lchamini foydalanuvchi tomonidan o'zgartirilish imkonini belgilaydi;
- `setLocation()` – oynani ekranda joylashishini o'zgartiradi, metod argument sifatida  $(x, y)$  qiymatlarini qabul qilib  $x$  qiymati gorizontal va  $y$  qiymati vertikal koordinata o'qlarini bildiradi.  $(0, 0)$  koordinatali nuqta ekranni yuqori-chap qismida joylashgan.
- `setBounds()` – metod argument sifatida  $(x, y, kenglik, balandlik)$  qiymatlarini qabul qilib  $x$  va  $y$  qiymatlari oyna joylashish nuqtalarini, kenglik va balandlik qiymatlari oyna kengligi va balandligini belgilaydi.

Quyidagi misolda ushbu metodlardan ba'zilarini qo'llagan xolda oyna yaratib olamiz.

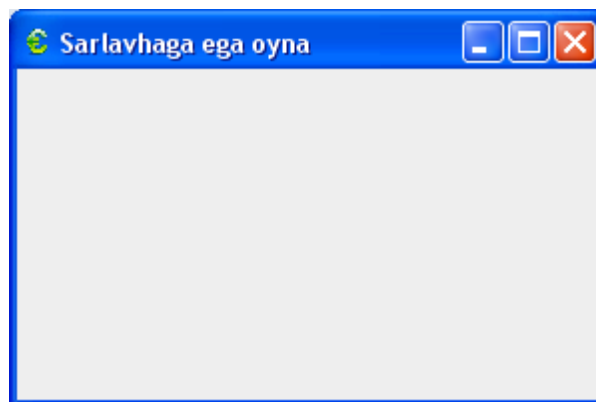
```
import javax.swing.*;
```

```

import java.awt.*;
public class SoddaFrame2{
public static void main(String[] args)
{
JFrame frame = new JFrame();
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.setTitle("Sarlavhaga ega oyna");
Toolkit kit = Toolkit.getDefaultToolkit();
Image img = kit.getImage("euro.png");
frame.setIconImage(img);
frame.setSize(300, 200);
frame.setVisible(true);
} }

```

Dastur quyidagi sarlavha matni va sarlavha yorlig'iga ega oynani ekranga chiqarib beradi:



*Sarlavhaga ega oyna*

Shuni eslatib o'tish kerakki sarlavha yorlig'i uchun ishlatiladigan rasm fayli dastur klassi bilan bitta papkada joylashishi kerak.

## 5-Mavzu

### 5.1 Ma'lumotlarni Panelda ko'rsatish

Java dasturlash tilida Frame boshqa foydalanuvchi interfeysi elementlari uchun konteyner vazifasini bajaradi. Foydalanuvchi interfeysi elementlari Panel elementiga chiziladi keyin esa ushbu Panel Frame oynasiga joylashtiriladi.

Panellar JPanel klassi yordamida yaratilib ular quyidagi xususiyatlarga ega:

-ular ustiga elementlar chizish mumkin;

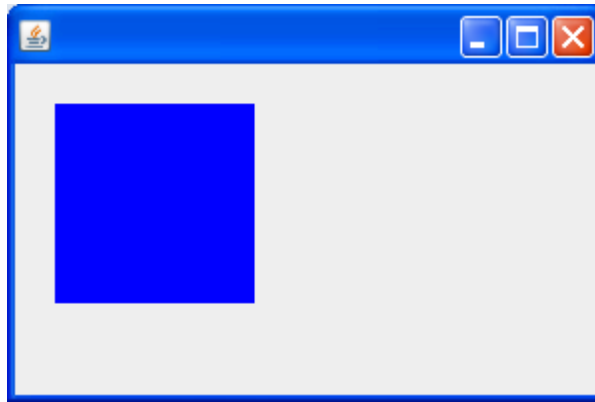
-ular o'z navbatida konteyner xisoblanib boshqa foydalanuvchi interfeysi elementlarni joylashtirishi mumkin.

Panel elementida biron bir rasm joylashtirish uchun JPanel klassini kengaytirib paintComponent() metodini qayta yaratish kerak bo'ladi. paintComponent() metodi JPanel klassini tomonidan JComponent klassidan meros qilib olingan bo'lib ushbu metod Graphics turidagi elementni argument sifatida qabul qiladi. Quyidagi misolni ko'rib chiqamiz:

```
import java.awt.*;
import javax.swing.*;
public class MeningRasmim{
public static void main(String[] args){
JFrame f = new JFrame();
f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
f.setSize(300,200);
MyPanel p = new MyPanel();
f.add(p);
f.setVisible(true);}}
class MyPanel extends JPanel{
public void paintComponent(Graphics g){
```

```
g.setColor(Color.orange);  
g.fillRect(20,20,100,100);  
} }
```

Ushbu dastur ekranga quyidagi oynani chiqarib beradi:



*JPanel klassining paintComponent() metodi yordamida chizilgan rasm*

MyPanel klassi JPanel klassini kengaytirib meros qilib olingan paintComponent() metodini qayta yaratadi. Graphics elementi yordamida ko'k rangli, 20,20 nuqtada boshlanadigan va 100x100 o'lchamga ega to'rtburchak chizib olinadi. MeningRasmim klassi Frame oynasini yaratadi va MyPanel klassi asosida panel yaratib Framega joylashtiradi.

JPanel klassining paintComponent() metodini qayta yaratib panelga tayyor rasmlarni xam joylashtirish mumkin. Yuqoridagi MeningRasmim dasturda MyPanel klassining paintComponent() metodini quyidagiga o'zgartiramiz:

```
public void paintComponent(Graphics g){  
Image rasm = new ImageIcon("sumka.png").getImage();  
g.drawImage(rasm,3,4,this); }
```

Dastur quyidagi oynani ekranga chiqarib beradi:

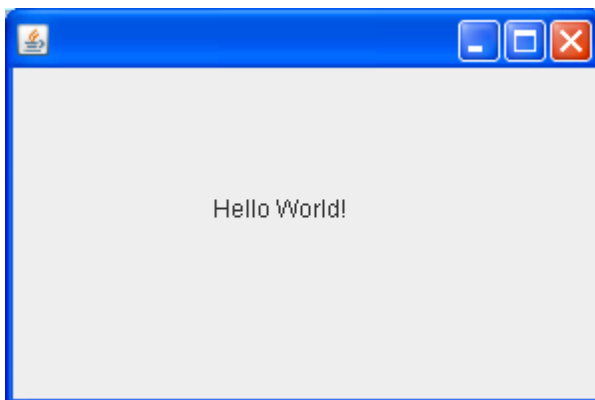


*JPanel klassining paintComponent() metodi yordamida aks ettirilgan rasm*

JPanel klassining paintComponent() metodi yordamida Panelga matn yozish xam mumkin. Buning uchun MeningRasmim klassida paintCpmponent() metodini quyidagiga o'zgartiramiz:

```
public void paintComponent(Graphics g){  
g.drawString("Hello World!",100,75);  
}
```

Dastur quyidagi oynani ekranga chiqarib beradi:



*JPanel klassining paintComponent() metodi yordamida aks ettirilgan matn*

## 6-Mazvu

### 6.1 Xodisa va amal

Java dasturlash tilida foydalanuvchi xosil qilgan xodisalarni qabul qiluvchi obyektlar qabul qilib ularga belgilangan amallarni bajaradi. Obyektlarga ixtisoslashgan dasturlash tiliga xos bo'lganidek, xodisa java.util o'ramida joylashgan `EventObject` klassi tomonidan yaratiladigan obyekt orqali ifodalanadi. Xar bir xodisa turi uchun o'zining ostki klasslari mavjud, masalan, tugmalar uchun `ActionEvent`, oynalar uchun esa `WindowEvent`. Xodisani xosil qiluvchi foydalanuvchi interfeysi elementi xodisa manbasi deyiladi.

Xodisalarni boshqarish quyidagicha amalga oshiriladi:

1. Xodisani qabul qiluvchi obyekt xodisa nazorat qiluvchi interfeysni bajaradi;
2. Xodisa manbasi xodisani qabul qila oladigan va unga xodisa obyektini yubora oladigan obyektini belgilaydi;
3. Xodisa xosil bo'lganda xodisa manbasi xodisa obyektini belgilangan barcha xodisani qabul qiluvchi obyektlarga jo'natadi;
4. Xodisani qabul qiluvchi obyekt xodisa obyektidagi ma'lumotni bajariladigan amalni xal qilishda ishlatadi.

Xodisani qabul qiluvchi obyektini xodisa manbasiga belgilash misoli quyida keltirilgan:

```
ActionListener listener = new TugmaListener();  
JButton button = new JButton("OK");  
button.addActionListener(listener);
```

Ushbu misolning birinchi qatorida xodisani qabul qiluvchi obyekt yaratilgan. Ikkinchi qatorda xodisa manbasi yaratilgan. Uchinchi qatorda xodisa manbasiga xodisani qabul qiluvchi obyekt belgilangan. Xodisa xosil qilinganda (ya'ni foydalanuvchi tugmani bosganda) xodisa manbasi xodisani xosil qilib (ushbu xolda `ActionEvent`

xodisasini) xodisa qabul qiluvchi obyektga uzatadi. Xodisani qabul qiluvchi obyekt klassi tegishli interfeysni (ushbu xolda `ActionListener` interfeysini) bajarishi kerak bo'ladi. `ActionListener` interfeysini bajarish uchun xodisani qabul qiluvchi obyekt klassi `ActionEvent` obyektini argument sifatida qabul qiluvchi `actionPerformed()` metodiga ega bo'lishi kerak.

```
public class TugmaListener implements ActionListener
public void actionPerformed(ActionEvent xodisa)
{
//bajariladigan amal kodi
}}
```

Foydalanuvchi tugmani bosganda `JButton` obyektini `ActionEvent` xodisa obyektini xosil qilib xodisani qabul qiluvchi obyektga uzatadi va `actionPerformed()` metodi chaqirilib undagi amal bajariladi.

## 6.2 Tugmalar va yozuvlar

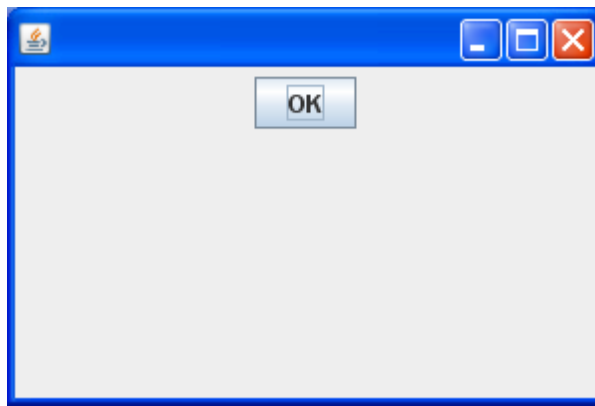
Tugma bu biron bir amalni bajarish uchun foydalanuvchi ruxsatini olish uchun ishlatiladigan element xisoblanadi. Java dasturlash tilida tugmalarni `JButton` klassi andozasi yordamida yaratib olish mumkin. Masalan:

```
JButton button = new JButton();
```

Dastur kodi yangi tugma yaratib uni `button` o'zgaruvchisiga biriktirib beradi. Yuqoridagi dastur kodi xech qanday yozuvga ega bo'lmagan tugmani yaratib beradi. Ammo, foydalanishni osonlashtirish maqsadida xar bir tugma bajaradigan amaliga asosan nomga ega bo'lishi kerak. Nomga ega tugmani yaratib olish uchun quyidagi dastur kodidan foydalanish mumkin:

```
JButton button = new JButton("OK");
```

Ushbu dastur kodi quyidagi tugmani yaratib beradi

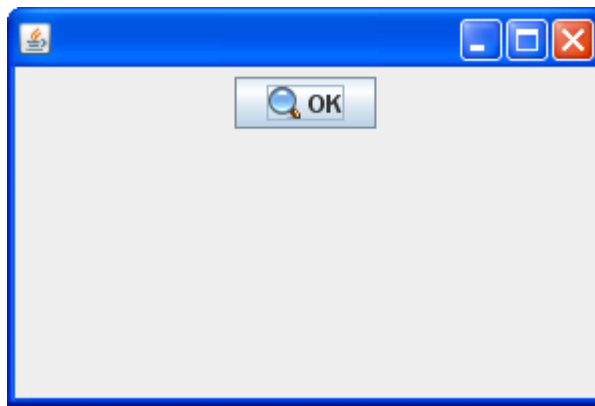


*JButton klassi yordamida yaratilgan tugma*

JButton klassi tugmalarga matndan tashqari rasm joylashtirish imkonini xam beradi. Rasmni fayldan yuklash uchun ImageIcon klassidan foydalaniladi. Masalan:

```
import java.awt.*;
import javax.swing.*;
public class RasmliTugma{
public static void main(String[] args){
JFrame frame = new JFrame();
JPanel panel = new JPanel();
Icon icon = new ImageIcon("search.png");
JButton button = new JButton("OK", icon);
panel.add(button);
frame.getContentPane().add(panel);
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.setSize(300,200);
frame.setVisible(true);
}}
```

Ushbu dastur quyidagi oynani ekranga chiqarib beradi:



*JButton klassi yordamida yaratilgan rasmi tugma*

Java dasturlash tilida tugma bosilganda `ActionEvent` xodisasi xosil qilinadi. Ushbu xodisani qabul qilish uchun dastur xodisalarini nazorat qiluvchi `ActionListener` interfeysini bajarishi va uning `actionPerformed()` metodini qayta yaratishi kerak bo'ladi. `actionPerformed()` metodi ichiga tugma bosilganda bajarilishi kerak bo'ladigan amallar kiritiladi. Masalan:

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

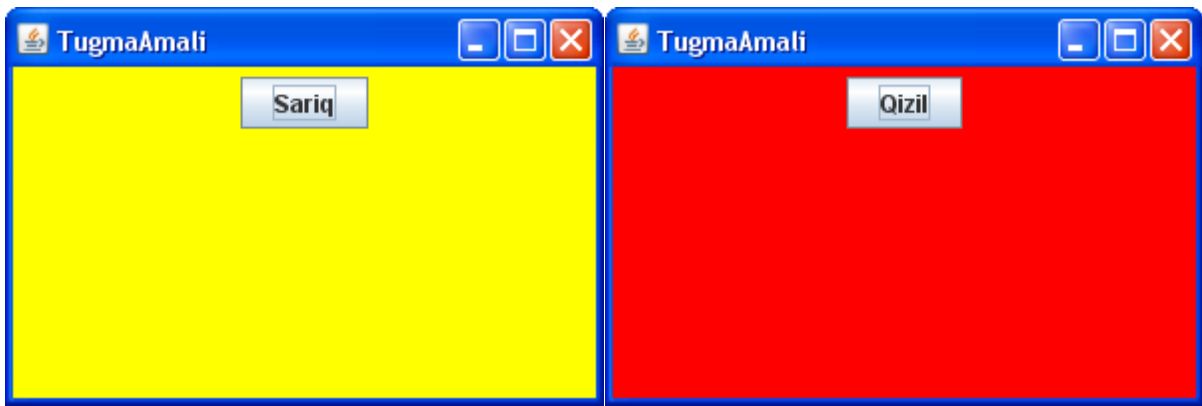
public class TugmaAmali{
    JPanel panel;
    Color backgroundColor = Color.YELLOW;
    JButton button;
    String tugmaMatni = "Sariq";
    public static void main(String[] args){
        TugmaAmali ta = new TugmaAmali();
        ta.go();
    }
    public void go(){
        JFrame frame = new JFrame("TugmaAmali");
```

```

panel = new JPanel();
panel.setBackground(backgroundColor);
button = new JButton(tugmaMatni);
button.addActionListener(new ButtonListener());
panel.add(button);
frame.getContentPane().add(panel);
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.setSize(300,200);
frame.setVisible(true);
}
public class ButtonListener implements ActionListener{
public void actionPerformed(ActionEvent e){
if(backgroundColor == Color.YELLOW){
backgroundColor = Color.RED;
tugmaMatni = "Qizil";}
else{
backgroundColor = Color.YELLOW;
tugmaMatni = "Sariq";
}
panel.setBackground(backgroundColor);
button.setText(tugmaMatni);}
}}

```

Yuqoridagi dastur tugma bosilganda oyna rangi sariq rangdan qizil rangga o'zgaradi.



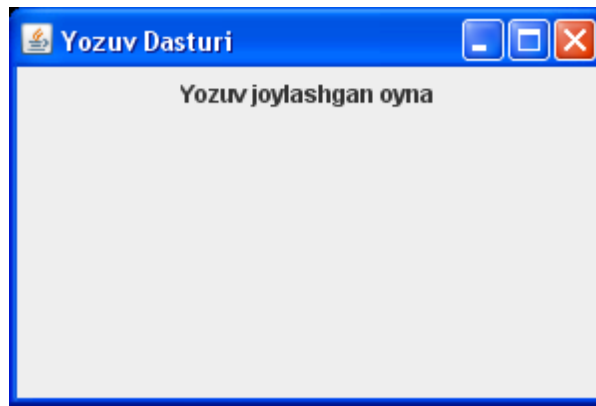
*Tugma xosil qilgan xodisa asosida amal bajarish*

### 6.3 Yozuvlar

Dastur oynasiga biron bir yozuvni qo'shish uchun JLabel klassidan foydalanish mumkin. Yozuvlar asosan oynada joylashgan elementlarga izoh berish yoki ularni nomlash uchun ishlatiladi. Masalan:

```
import java.awt.*;
import javax.swing.*;
public class YozuvDasturi{
public static void main(String[] args){
JFrame frame = new JFrame("Yozuv Dasturi");
JLabel label = new JLabel("Yozuv joylashgan oyna");
JPanel panel = new JPanel();
panel.add(label);
frame.getContentPane().add(panel);
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.setSize(300,200);
frame.setVisible(true);
}}
```

Ushbu dastur yozuv joylashgan quyidagi oynani ekranga chiqarib beradi:



*JLabel klassi yordamida yaratilgan yozuv*

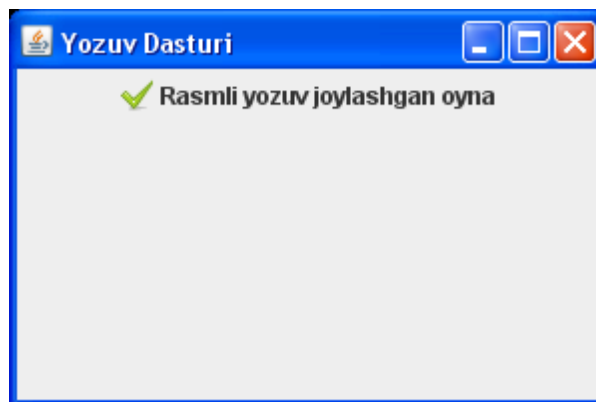
Yozuvni oldida joylashgan rasm bilan xam yaratish mumkin. Buning uchun ImageIcon klassi yordamida rasmini fayldan import qilib JLabel klassi konstruktoriga uzatish kerak bo'ladi. Masalan, YozuvDasturi dasturga quyidagi o'zgarishlar kiritish kerak:

...

```
Icon icon = new ImageIcon("note.png");  
JLabel label = new JLabel("Rasmlı yozuv joylashgan oyna",  
icon, SwingConstants.CENTER);
```

...

O'zgartirilgan dastur quyidagi oynani ekranga chiqarib beradi:



*JLabel klassi yordamida yaratilgan rasmlı yozuv*

Shuni yodda tutish lozimki, note.png rasm fayli YozuvDasturi klass fayli joylashgan papkada joylashgan bo'lishi kerak.

Yozuvni oynada joylashishi SwingConstants interfeysining konstantalari orqali belgilanadi. Masalan, bizning xolda SwingConstants.CENTER konstantasi yozuvni markazga tekislab joylashtiradi.

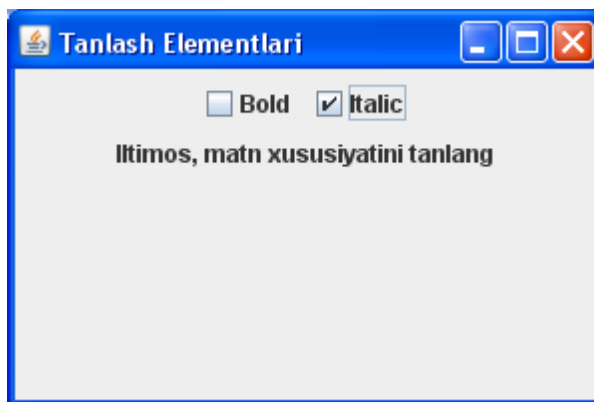
# AMALIY MASHG'ULOT MAVZULARI

## 1-Mavzu

### 1.1 Checkbox va Radio tugmalar

Java dasturlash ptili foydalanuvchidan ma'lumot yig'ish uchun bir necha turdagi elementlarni taklif etadi. Checkbox va Radio tugmalar grafikli foydalanuvchi interfeysining tanlash elementlari xisoblanadi va foydalanuvchidan belgilangan variantlar orasidan tanlash imkonini beradi.

Checkbox elementi foydalanuvchiga “xa” yoki “yo’q” variantlarini tanlash imkonini beradi. Checkbox elementi yozuvga ega bo’lib ushbu yozuv Checkbox maqsadini belgilaydi. Foydalanuvchi Checkbox elementini sichqon bilan bosish orqali tanlaydi yoki tanlanishni olib tashlaydi. Checkbox elementi JCheckBox klassi yordamida xosil qilinadi. Quyidagi oynada Checkbox elementlari ko’rsatilgan:



*JCheckBox klassi yordamida yaratilgan chekbox elementi*

Ushbu oynani quyidagi dastur xosil qiladi:

```
import java.awt.*;  
import javax.swing.*;  
public class TanlashElementlari{  
public static void main(String[] args){  
TanlashElementlari te = new TanlashElementlari();  
te.go();  
}  
}
```

```

public void go(){
JFrame frame = new JFrame("Tanlash Elementlari");
JPanel panel = new JPanel( );
JCheckBox ch1 = new JCheckBox("Bold");
JCheckBox ch2 = new JCheckBox("Italic");
JLabel label = new JLabel("Iltimos, matn xususiyatini
tanlang");
panel.add(ch1);
panel.add(ch2);
panel.add(label);
frame.getContentPane().add(panel);
frame.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
frame.setSize(300,200);
frame.setVisible(true);
}}

```

Checkbox elementi tanlanganlik xolatini bilish uchun Checkbox klassini isSelected() metodi ishlatiladi. Agar Checkbox tanlangan bo'lsa metod true qiymatini, tanlanmagan bo'lsa false qiymatini qaytaradi.

Checkbox elementi bosilganda ActionEvent xodisasi xosil qilinadi va ushbu xodisani ActionListener orqali qabul qilib amal bajarish mumkin. Masalan, quyidagi dastur Checkbox tanlanganda Label elementi ustida amal bajariladi:

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
public class TanlashElementlari{
JLabel label;
JCheckBox ch1;
JCheckBox ch2;

```

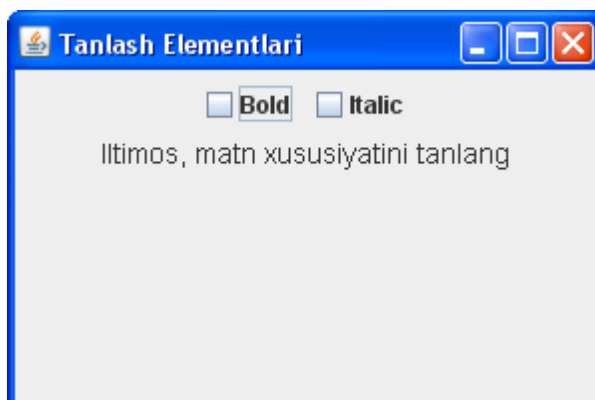
```

public static void main(String[] args){
    TanlashElementlari te = new TanlashElementlari();
    te.go();
}
public void go(){
    JFrame frame = new JFrame("Tanlash Elementlari");
    JPanel panel = new JPanel( );
    ActionListener Listener = new Listener();
    ch1 = new JCheckBox("Bold");
    ch1.addActionListener(Listener);
    ch2 = new JCheckBox("Italic");
    ch2.addActionListener(Listener);
    label = new JLabel("Iltimos, matn xususiyatini tanlang");
    label.setFont(new Font("Arial", Font.PLAIN, 14));
    panel.add(ch1);
    panel.add(ch2);
    panel.add(label);
    frame.getContentPane().add(panel);
    frame.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
    frame.setSize(300,200);
    frame.setVisible(true);
}
public class Listener implements ActionListener{
    public void actionPerformed(ActionEvent e){
        int fm = 0;
        if(ch1.isSelected()) fm = fm + Font.BOLD;
        if(ch2.isSelected()) fm = fm + Font.ITALIC;
        label.setFont(new Font("Arial", fm, 14));
    }
}

```

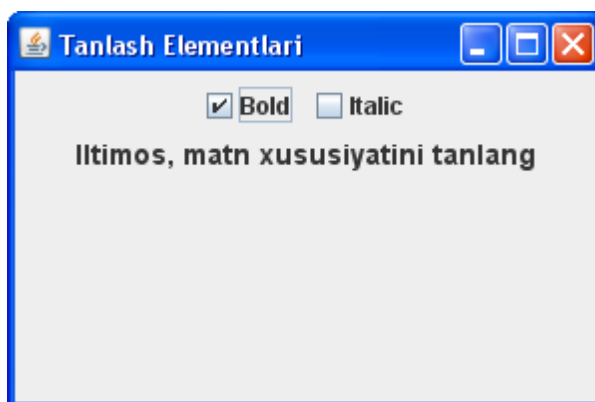
}}}

Ushbu dastur quyidagi oynani ekranga chiqarib beradi:



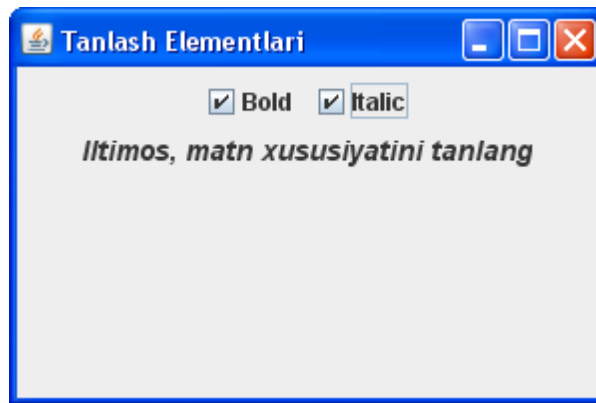
*Checkbox elementi va u xosil qilgan xodisa asosida amal bajarish*

Bold yozuviga ega Checkbox elementi tanlanganda “Iltimos, matn xususiyatini tanlang” matni qalin yozuvga o’zgaradi:



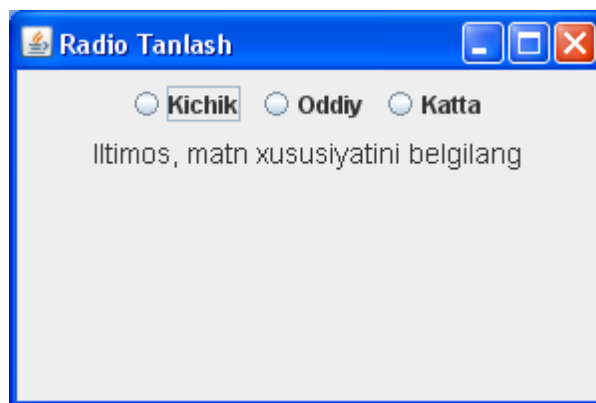
*Checkbox elementi va u xosil qilgan xodisa asosida matn ko’rinishini qalinroq qilish*

Bold va Italic yozuvlariga ega Checkbox elementlari tanlanganda matn qalin va kursiv yozuvga o’zgaradi:



*Checkbox elementi va u xosil qilgan xodisa asosida matn ko'rinishini qalinroq va kursiv qilish*

Checkbox elementi birdaniga bir nechta variantlarni tanlash imkonini berish uchun ishlatiladi. Bir nechta variantlardan faqat bittasini tanlash uchun Radio tugmalar ishlatiladi. Bir variant tanlanganda boshqa tanlangan variantlar belgilanishi bekor qilinadi. Buning uchun Radio tugmalar bir guruhda bo'lishi kerak. Radio tugmalar `JRadioButton` klassi yordamida yaratiladi. Radio tugmalar guruhi `ButtonGroup` klassi yordamida xosil qilinadi. Quyidagi oynada Radio tugmalar ko'rsatilgan:



*JRadioButton klassi yordamida yaratilgan radio tugmalar*

Ushbu oyna quyidagi dastur yordamida yaratilgan:

```
import java.awt.*;  
import javax.swing.*;  
public class RadioTanlash{  
public static void main(String[] args){  
RadioTanlash rt = new RadioTanlash();
```

```

rt.go();
}
public void go(){
JFrame frame = new JFrame("Radio Tanlash");
JPanel panel = new JPanel( );
ButtonGroup guruh = new ButtonGroup();
JRadioButton rb1 = new JRadioButton("Kichik");
guruh.add(rb1);
JRadioButton rb2 = new JRadioButton("Oddiy");
guruh.add(rb2);
JRadioButton rb3 = new JRadioButton("Katta");
guruh.add(rb3);
JLabel label = new JLabel("Iltimos, matn xususiyatini
belgilang");
label.setFont(new Font("Arial", Font.PLAIN, 14));
panel.add(rb1);
panel.add(rb2);
panel.add(rb3);
panel.add(label);
frame.getContentPane().add(panel);
frame.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
frame.setSize(300,200);
frame.setVisible(true); }}

```

Radio button element foydalanuvchi tomonidan tanlanganda `ActionEvent` xodisasi xosil qilinadi va `ActionListener` interfeysi yordamida ushbu xodisani qabul qilib biron bir amal bajarib olish mumkin. Masalan, quyidagi dastur tegishli Radio tugmasi tanlanganda oynadagi matn o'lchamini o'zgartirib beradi:

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
public class RadioTanlash {
    JRadioButton rb1;
    JRadioButton rb2;
    JRadioButton rb3;
    JLabel label;
    public static void main(String[] args){
        RadioTanlash rt = new RadioTanlash();
        rt.go();
    }
    public void go(){
        JFrame frame = new JFrame("Radio Tanlash");
        JPanel panel = new JPanel( );
        ActionListener Listener = new Listener();
        ButtonGroup guruh = new ButtonGroup();
        rb1 = new JRadioButton("Kichik");
        rb1.addActionListener(Listener);
        guruh.add(rb1);
        rb2 = new JRadioButton("Oddiy");
        rb2.addActionListener(Listener);
        guruh.add(rb2);
        rb3 = new JRadioButton("Katta");
        rb3.addActionListener(Listener);
        guruh.add(rb3);
        label = new JLabel("Iltimos, matn xususiyatini belgilang");
        label.setFont(new Font("Arial", Font.PLAIN, 14));
    }
}

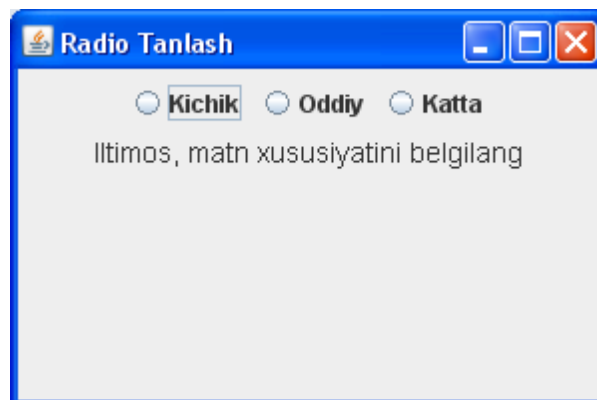
```

```

panel.add(rb1);
panel.add(rb2);
panel.add(rb3);
panel.add(label);
frame.getContentPane().add(panel);
frame.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
frame.setSize(300,200);
frame.setVisible(true);
}
public class Listener implements ActionListener{
public void actionPerformed(ActionEvent e){
if(rb1.isSelected()) label.setFont(new Font("Arial",
Font.PLAIN, 12));
if(rb2.isSelected()) label.setFont(new Font("Arial",
Font.PLAIN, 14));
if(rb3.isSelected()) label.setFont(new Font("Arial",
Font.PLAIN, 16));
}}
}

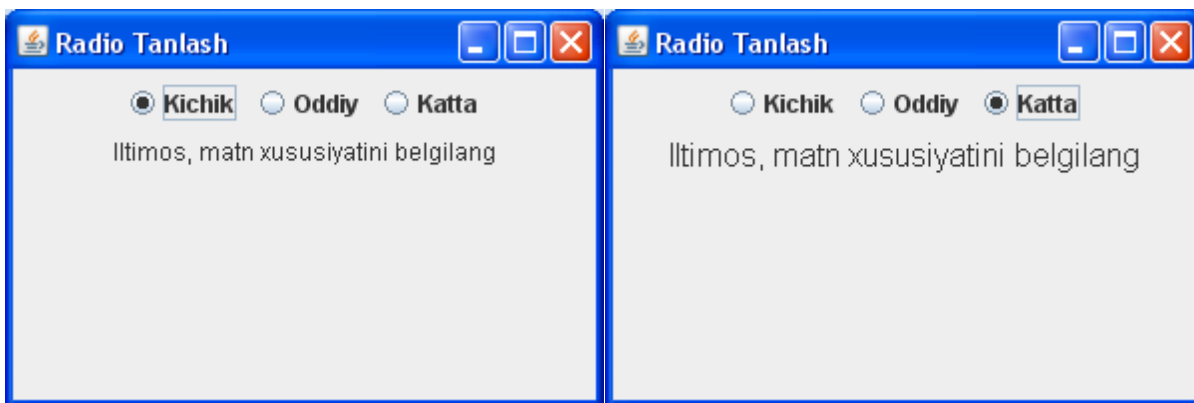
```

Ushbu dastur quyidagi oynani ekrangachiqarib beradi:



*Radio tugmalar va u xosil qilgan xodisa asosida amal bajarish*

Kichik va Katta yozuvlariga ega Radio tugmalari tanlanganda matn o'lchami o'zgarishi quyidagi oynalarda ko'rsatilgan:



*Radio tugma tanlanganda xosil qilingan xodisa asosida matn ko'rinishini o'zgartirish*  
 Guruhda bo'lgan Radio tugmalarning bir vaqtning o'zida faqatgina bittasi tanlangan bo'lishi mumkin.

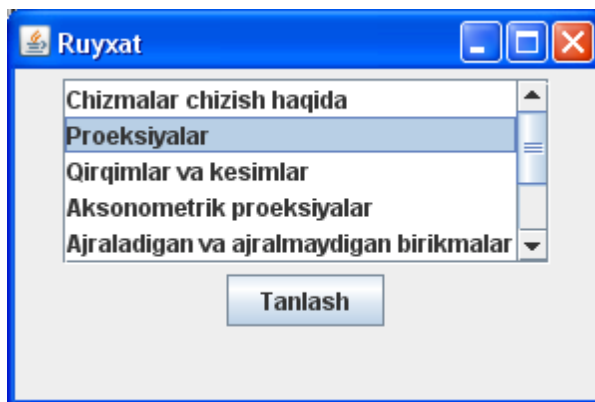
## 2-Mavzu

### 2.1 Ro'yxat va Combobox

Royxat va Combobox tanlash elementlari xisoblanib, ko'p variantlar orasidan tanlash kerak bo'lganda oynada joyni tejash uchun ishlatiladi.

Ro'yxat `JList` klassi tomonidan yaratiladi va foydalanuvchiga belgilangan variantlardan tanlash imkonini beradi. Ro'yxatda faqatgina bitta variantni tanlash yoki bir necha variantlarni tanlash imkoniyati mavjud. Ro'yxat yaratishda variantlarni ifoda etuvchi ma'lumotlar modeli ishlatiladi. Ushbu modelni massiv yordamida ifoda etish mumkin.

Quyidagi misolda ro'yxat joylashgan oyna keltirilgan:



*JList klassi yordamida yaratilgan ro'yxat*

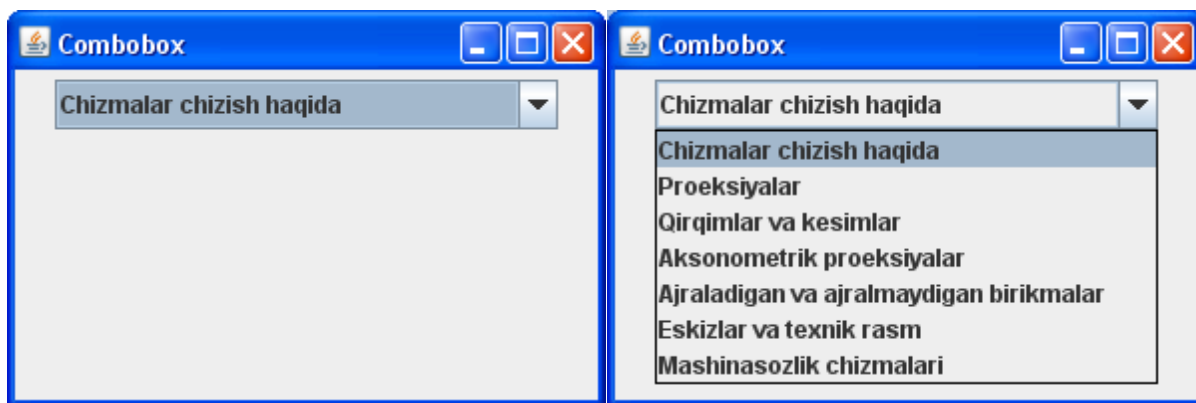
Ushbu oyna quyidagi dastur yordamida yaratilgan:

```
import java.awt.*;
import javax.swing.*;
public class Ruyxat{
public static void main(String[] args){
JFrame frame = new JFrame("Ruyxat");
JPanel panel = new JPanel();
String [] malumotModeli = {"Chizmalar chizish haqida",
"Proeksiyalar", "Qirqimlar va kesimlar", "Aksonometrik
proeksiyalar", "Ajraladigan va ajralmaydigan birikmalar",
"Eskizlar va texnik rasm", "Mashinasozlik chizmalari"};
JList list = new JList(malumotModeli);
list.setVisibleRowCount(5);
JScrollPane scrollPane = new JScrollPane();
scrollPane.getViewPort().setView(list);
JButton button = new JButton("Tanlash");
panel.add(scrollPane);
panel.add(button);
frame.getContentPane().add(panel);
frame.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
frame.setSize(300,200);
frame.setVisible(true);
}}
```

Ushbu dasturda ro'yxat yaratishda ma'lumotlar modeli maluoModeli massivi yordamida ifoda etilgan. JList klassini setVisibleRowCount() metodi ro'yxatni joriy ko'rinadigan qatorlar sonini belgilash uchun ishlatiladi. JScrollPane klassi ro'yxatni boshqa elementlarini ko'rib olish uchun ro'yxatga gorizontol o'tkazgich

biriktirib beradi. Tanlangan elementni indeksi yoki qiymatini olish uchun `JList` klassini `getSelectedIndex()` yoki `getSelectedValue()` metodlaridan foydalanish mumkin.

Combobox elementi ro'yxatdan farqli o'laroq tanlash variantlaridan faqatgina bittasini ko'rsatib beradi. Qolgan variantlarni ko'rib olish uchun Combobox yonboshida joylashgan tugmaga bosish kerak bo'ladi. Combobox `JComboBox` klassi yordamida yaratiladi. Quyidagi misolda Combobox joylashgan oyna va Combobox ruyxatini ochilgan xolati keltirilgan:



*JComboBox klassi yordamida yaratilgan combobox elementi*

Ushbu oyna quyidagi dastur yordamida yaratilgan:

```
import java.awt.*;
import javax.swing.*;
public class Combobox{
public static void main(String[] args){
JFrame frame = new JFrame("Combobox");
JPanel panel = new JPanel();
String [] malumotModeli = {"Chizmalar chizish haqida",
"Proeksiyalar", "Qirqimlar va kesimlar", "Aksonometrik
proeksiyalar", "Ajraladigan va ajralmaydigan birikmalar",
"Eskizlar va texnik rasm", "Mashinasozlik chizmalari"};
```

```

JComboBox combo = new JComboBox(malumatModeli);
panel.add(combo);
frame.getContentPane().add(panel);
frame.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
frame.setSize(300,200);
frame.setVisible(true);
}}

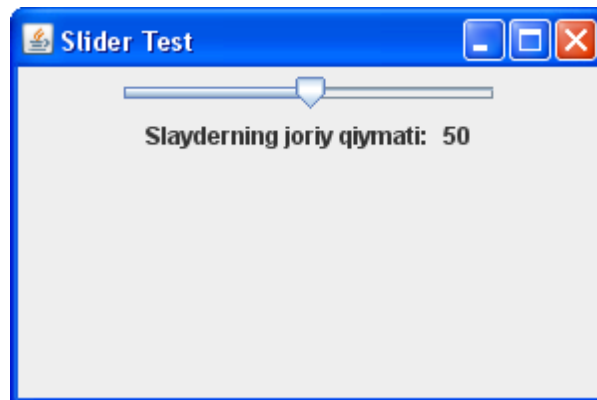
```

Tanlangan elementni yoki uning indeksini olish uchun JComboBox klassini `getSelectedItem()` yoki `getSelectedIndex()` metodlaridan foydalanish mumkin.

### 3-Mavzu

#### 3.1 Slayder va Spinner

Slayder elementi berilgan minimal va maksimal qiymatlar orasidan qiymat tanlash imkonini beradi. Slayder `JSlider` klassi yordamida yaratiladi. Quyidagi misolda Slayder joylashgan oyna keltirilgan:



*JSlider klassi yordamida yaratilgan slayder elementi*

Ushbu oyna quyidagi dastur yordamida yaratilgan:

```

import java.awt.*;
import javax.swing.*;

```

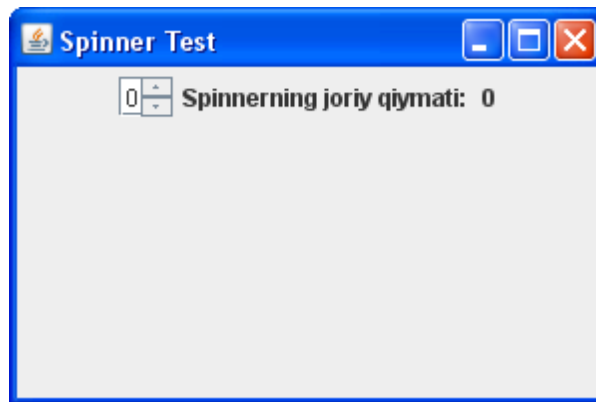
```

import javax.swing.event.*;
public class SliderTest{
JLabel qiymat;
JSlider slider;
public static void main(String[] args){
SliderTest st = new SliderTest();
st.go();
}
public void go(){
JFrame frame = new JFrame("Slider Test");
JPanel panel = new JPanel();
slider = new JSlider(0, 100, 50);
slider.addChangeListener(new sliderChange());
JLabel label = new JLabel("Slaydarning joriy qiymati: ");
qiymat = new JLabel("" + slider.getValue());
panel.add(slider);
panel.add(label);
panel.add(qiymat);
frame.getContentPane().add(panel);
frame.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
frame.setSize(300,200);
frame.setVisible(true);
}
public class sliderChange implements ChangeListener{
public void stateChanged(ChangeEvent e){
qiymat.setText("" + slider.getValue());
}}
}

```

Slayder yurgichi xolati o'zgartirilganda `ChangeEvent` xodisasi xosil qilinadi va ushbu xodisani `ChangeListener` interfeysini bajarib qabul qilish va amal bajarish mumkin. Slayderni joriy qiymatini olish uchun `JSlider` klassini `getValue()` metodidan foydalanish mumkin.

Spinner elementi qiymat tanlash uchun ishlatilib, Slayder elementidan farqli o'laroq minimal va maksimal qiymatlari bilan chegaralanmagan. Element yonboshida joylashgan yuqoriga va pastga qaragan strelkalar bilan Spinner qiymatini oshirish yoki kamaytirish mumkin. Spinner elementi `JSpinner` klassi yordamida yaratiladi. Quyidagi misolda Spinner joylashgan oyna keltirilgan:



*JSpinner klassi yordamida yaratilgan spinner elementi*

Ushbu oyna quyidagi dastur yordamida yaratilgan:

```
import java.awt.*;
import javax.swing.*;
import javax.swing.event.*;

public class SpinnerTest {
    JLabel qiymat;
    JSpinner spinner;

    public static void main(String[] args) {
        SpinnerTest st = new SpinnerTest();
        st.go();
    }

    public void go() {
```

```

JFrame frame = new JFrame("Spinner Test");
JPanel panel = new JPanel();
spinner = new JSpinner();
spinner.addChangeListener(new spinnerChange());
JLabel label = new JLabel("Spinnerning joriy qiymati: ");
qiymat = new JLabel("" + spinner.getValue());
panel.add(spinner);
panel.add(label);
panel.add(qiymat);
frame.getContentPane().add(panel);
frame.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
frame.setSize(300,200);
frame.setVisible(true);
}
public class spinnerChange implements ChangeListener{
public void stateChanged(ChangeEvent e){
qiymat.setText("" + spinner.getValue());
}}
}

```

Spinner elementi belgilangan elementlar orasidan tanlash imkonini xam beradi. Buning uchun Spinner konstruktoriga ma'lumotlar modelini uzatish mumkin. Ma'lumotlar modeli massiv yordamida yaratiladi. Masalan:

```

String[] qiymatlar =
{"sariq", "qizil", "oq", "qora", "zangor"};
SpinnerListModel model = new SpinnerListModel(qiymatlar);
JSpinner spinner = new JSpinner(model);

```

### 3.2 Matn kiritish

Foydalanuvchi matn kiritishi uchun matn qatori va matn maydoni elementlari ishlatiladi. Java dasturlash tilidi matn kiritish uchun `JTextField` va `JTextArea` klasslari mavjud. `JTextField` klassi bitta qatordan iborat matnni, `JTextArea` klassi bir nechta qatordan iborat matnni qabul qiladi.

Matn qatorini xosil qilish uchun quyidagi dastur kodidan foydalanish mumkin:

```
JTextField matnQatori=new JTextField("Kiritiladigan matn",  
20);
```

Ushbu dastur kodi yangi matn qatorini yaratib unga “Kiritiladigan matn” matnini qo’shib beradi. `JTextField` konstruktoriga uzatilayotgan ikkinchi qiyat matn qatorini kengligini bildiradi, ya’ni bizning xolda matn qatori 20 simvol kenglikka ega.

Agar matn kiritish uchun foydalanuvchiga bo’sh matn qatori berish kerak bo’lsa quyidagi dastur kodidan foydalanish mumkin:

```
JTextField matnQatori=new JTextField(20);
```

Bu xolda 20 simvol kenglikka ega bo’sh matn qatori yaratiladi.

Matn qatoridagi matnni o’zgartirish uchun `setText()` metodidan, foydalanuvchi kiritgan matnni olish uchun `getText()` metodidan foydalanish mumkin.

```
JTextField matnQatori = new JTextField(20);
```

Bu xolda 20 simvol kenglikka ega bo’sh matn qatori yaratiladi.

Matn qatoridagi matnni o’zgartirish uchun `setText()` metodidan, foydalanuvchi kiritgan matnni olish uchun `getText()` metodidan foydalanish mumkin.

```
matnQatori.setText("Yangi matn");
```

```
String matn = matnQatori.getText();
```

Foydalanuvchi kiritadigan matn bir qatordan ortiq bo’lsa matn qatori o’rniga matn maydoni ishlatiladi. Matn maydoni `JTextArea` klassi tomonidan yaratiladi. Matn maydoniga foydalanuvchi istalgan sondagi matn qatorini kiritishi mumkin. Matn maydoni quyidagi dastur kodi orqali yaratiladi:

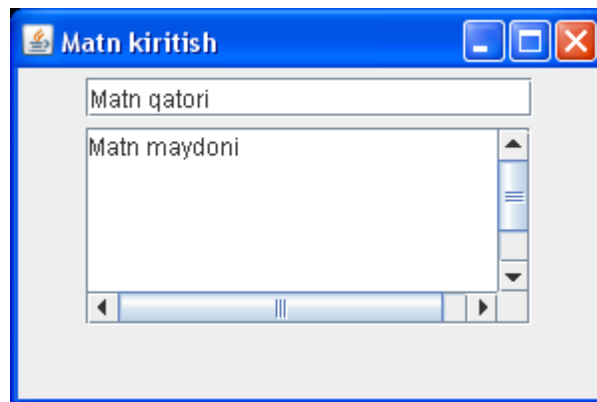
```
JTextArea matnMaydoni = new JTextArea(6, 20);
```

Ushbu dastur kodi 6 qator va 20 simvol uzunlikdagi bo'sh matn maydonini xosil qiladi. Oldindan kiritilgan matn maydonini xosil qilish uchun quyidagi dastur kodidan foydalanish mumkin:

```
JTextArea matnMaudoni = new JTextArea("Oldindan kiritilgan  
matn", 6, 20);
```

Matn qatori singari matn maydonidagi matnni o'zgartirish uchun `setText()` metodidan va foydalanuvchi kiritgan matnni olish uchun `getText()` metodidan foydalaniladi. Agar foydalanuvchi kiritayotgan matn qatorga sig'masa qolgan matnni keyingi qatorga avtomatik ravishda o'tkazish uchun `setLineWrap()` metodidan foydalanish mumkin. Biron bir matnni maydondagi matn oxiriga qo'shish uchun `append()` metodidan foydalanish mumkin. Matn maydonida joylashgan barcha matnni ko'rib olish uchun `JScrollPane` klassidan foydalanib maydon chetlariga gorizontal va vertikal surgichlar joylashtirib olish mumkin.

Quyidagi misolda matn qatori va matn maydoni joylashgan oyna keltirilgan:



*JTextField va JTextArea klasslari yordamida yaratilgan matn qatori va matn maydoni*

Ushbu oyna quyidagi dastur yordamida yaratilgan:

```
import java.awt.*;  
import javax.swing.*;  
public class MatnKiritish{  
public static void main(String[] args){
```

```

JFrame frame = new JFrame("Matn kiritish");
JPanel panel = new JPanel();
JTextField matnQatori = new JTextField("Matn qatori",
20);
JTextArea matnMaydoni = new JTextArea("Matn maydoni", 6,
20);
JScrollPane scrollPane = new JScrollPane();
scrollPane.getViewPort().setView(matnMaydoni);
panel.add(matnQatori);
panel.add(scrollPane);
frame.getContentPane().add(panel);
frame.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
frame.setSize(300,200);
frame.setVisible(true);}}

```

## **4-Mavzu**

### **4.1 Slayder va Spinner**

Java dasturlash tili oynaga foydalanuvchi interfeysi elementi xisoblanadigan menyular joylashtirish imkoniyatini beradi. Menyu xosil qilish uchun birinchi navbata menyular ro'yxati joylashgan menyu qatori yaratiladi. Menyu qatori JMenuBar klassi tomonidan yaratiladi.

```
JMenuBar menyuQatori = new JMenuBar();
```

Menyu elementi JMenu klassi yordamida yaratilib JMenuBar klassining add() metodi yordamida menyu qatoriga qo'shiladi.

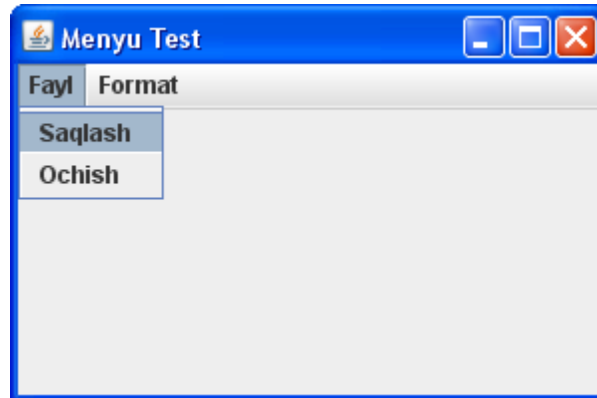
```
JMenu faylMenyusi = new JMenu("Fayl");
    menyuQatori.add(faylMenyusi);
```

So'ngra, menyu elementlari JMenuItem klassi yordamida yaratiladi va JMenu klassining add() metodi yordamida menyuga qo'shiladi.

```
JMenuItem saqlashItem = new JMenuItem("Saqlash");  
faylMenyusi.add(saqlashItem);
```

Menyu elementlari bosilganda `ActionEvent` xodisasi xosil qilinib uni `ActionListener` interfeysi yordamida qabul qilib ushbu menyu elementiga tegishli amal bajarish mumkin.

Quyidagi misolda menyu qatori joylashgan oyna keltirilgan:



*JMenuBar, JMenu va JMenuItem klasslari yordamida yaratilgan menyu*

Ushbu oyna quyidagi dastur yordamida yaratilgan:

```
import java.awt.*;  
import javax.swing.*;  
public class MenyuTest {  
    public static void main(String[] args) {  
        JFrame frame = new JFrame("Menyu Test");  
        JMenuBar menyuQatori = new JMenuBar();  
        JMenu faylMenyusi = new JMenu("Fayl");  
        menyuQatori.add(faylMenyusi);  
        JMenuItem saqlashItem = new JMenuItem("Saqlash");  
        faylMenyusi.add(saqlashItem);  
        JMenuItem ochishItem = new JMenuItem("Ochish");  
        faylMenyusi.add(ochishItem);  
        JMenu formatMenyusi = new JMenu("Format");
```

```

menuQatori.add(formatMenyusi);
JMenuItem shriftItem = new JMenuItem("Shrift");
formatMenyusi.add(shriftItem);
frame.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
frame.setSize(300,200);
frame.setJMenuBar(menuQatori);
frame.setVisible(true);}}

```

Menyu elementlari tugmalar singari yozuvga qo'shimcha rasmga xam ega bo'lishi.

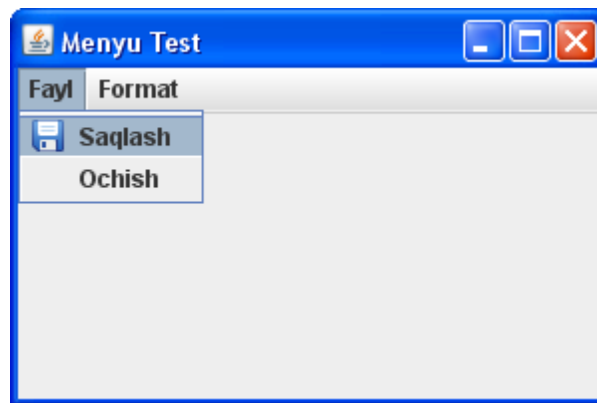
Buning uchun JMenuItem klassining quyidagi dastur kodidan foydalanish mumin:

```

Icon saqlashIcon = new ImageIcon("saqlash.png");
JMenuItem saqlashItem = new JMenuItem("Saqlash",
saqlashIcon);

```

MenyuTest dasturiga yuqoridagi o'zgarishni kiritib quyidagi o'zgarishni olamiz:



*Rasmi menyu elementi*

Shuni yodda tutish kerakki, saqlash.png fayli MenyuTest dasturi joylashgan papkada joylashisji kerak.

Menyuga uning elementi sifatida boshqa menyuni ostki menyu sifatida joylashtirish mumkin. Masalan, MenyuTest dasturida yangi Menyu yaratib uni Fayl menyusiga ostki menyu sifatida qo'shamiz.

```

JMenu importMenyu = new JMenu("Import");

```

```
JMenuItem xujjatItem = new JMenuItem("Xujjatni import qilish");
importMenyu.add(xujjatItem);
faylMenyusi.add(importMenyu);
```

MenuTest dasturiga ushbu o'zgarishlarni kiritib quyidagini olamiz:

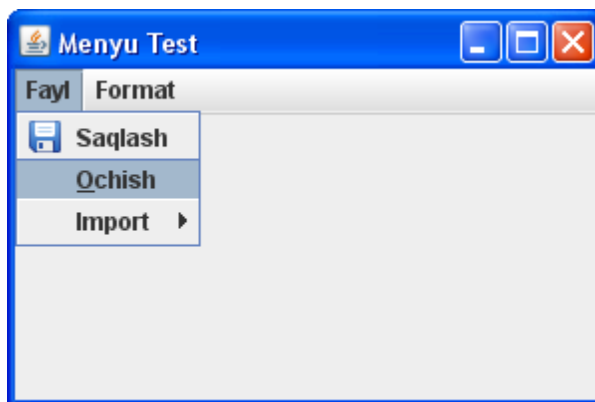


*Ostki meyuga ega menyu*

Foydalanuvchi menyu elementlarini faqatgina sichqon yordamida emas, balki klaviatura yordamida tanlashi mumkin. Buning uchun menyu elementlariga tegishli klaviatura qisqartmalarini biriktirish kerak. JMenuItem klassining quyidagi konstruktoridan foydalanish mumkin:

```
JMenuItem ochishItem = new JMenuItem("Ochish", 'O');
```

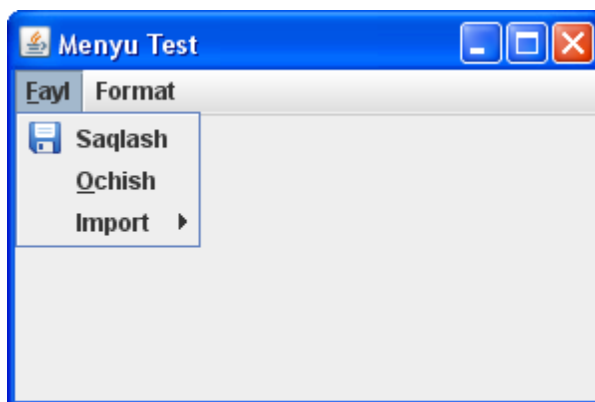
Natijada menyu elementining konstruktorning ikkinchi parametrda berilgan xarfi tagiga chizilib qo'yoladi va ushbu menyu tanlanishi uchun klaviaturada Alt va tagiga chizilib qo'yilgan xarf bosiladi



*Klaviatura qisqartmasiga ega menyu elementi*

Menu elementidan tashqari klaviatura qisqartmasini menyuning o'ziga xam belgilash mumkin. Buning uchun JMenu klassining setMnemonic() metodidan foydalanish mumkin.

```
faylMenyusi.setMnemonic('F');
```



*Klaviatura qisqartmasiga ega menyu*

Endi MenuTest dasturida Fayl menyusining Ochish elementini tanlash uchun Alt+F va Alt+O tugmalar kombinatsiyasidan foydalanish mumkin.

## **5-Mavzu**

### **5.1 Qurollar paneli**

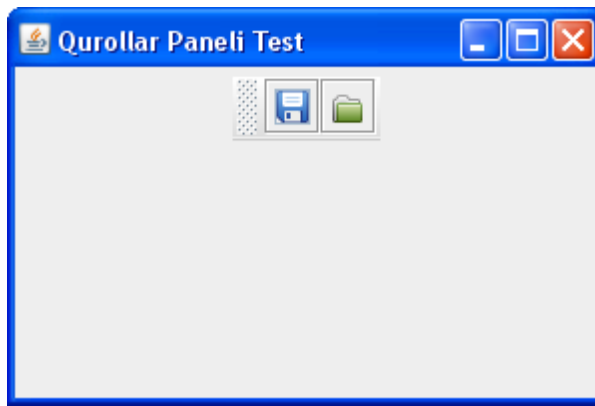
Qurollar paneli dasturning ko'p ishlatiladigan ammallarini tezda ishga tushirish imkonini beruvchi tugmalar to'plamidir. Qurollar paneli JToolBar klassi yordamida yaratiladi.

```
JToolBar qurollarPaneli = new JToolBar();
```

Qurollar paneliga tugma joylashtirish uchun JToolBar klassining add() metodidan foydalaniladi.

```
Icon saqlashIcon = new ImageIcon("saqlash.png");  
JButton saqlashButton = new JButton(saqlashIcon);  
qurollarPaneli.add(saqlashButton);
```

Quyidagi oynada qurollar paneli joylashgan oyna keltirilgan.



*JToolBar klassi yordamida yaratilgan qurollar paneli*

Ushbu oyna quyidagi dastur tomonidan yaratilgan:

```
import java.awt.*;
import javax.swing.*;

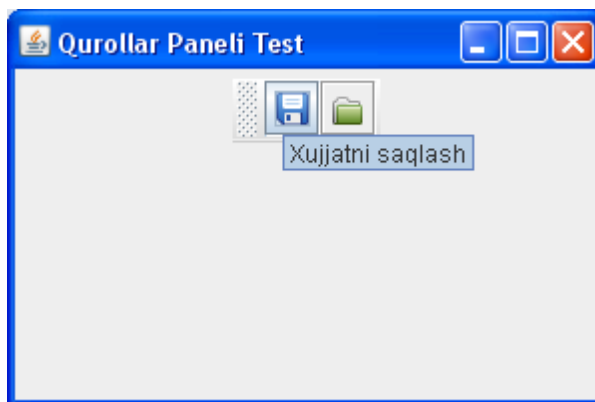
public class QurollarPaneliTest{
public static void main(String[] args){
JFrame frame = new JFrame("Qurollar Paneli Test");
JPanel panel = new JPanel();
JToolBar qurollarPaneli = new JToolBar();
Icon saqlashIcon = new ImageIcon("saqlash.png");
JButton saqlashButton = new JButton(saqlashIcon);
Icon ochishIcon = new ImageIcon("ochish.png");
JButton ochishButton = new JButton(ochishIcon);
qurollarPaneli.add(saqlashButton);
qurollarPaneli.add(ochishButton);
panel.add(qurollarPaneli);
frame.getContentPane().add(panel);
frame.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
frame.setSize(300,200);
frame.setVisible(true);
}}
```

Qurollar panelining boshqa foydalanuvchi interfeysi elementlaridan farqi shundaki uni oyna bo'ylab istalgan joyga siljitish mumkin.

Qurollar panelining kamchiligi shundaki unda joylashgan tugmalar yozuvi odatda faqatgina rasmdan iborat va foydalanuvchi qaysi rasm qanday amalga tegishliligini eslab qolishi qiyin. Ushbu kamchilikni oldini olish uchun tugmalarga biriktirilgan izohlardan foydalanish mumkin. Tugmaga biriktirilgan izoh sichqon ko'rsatkichi ushbu tugma ustiga joylashtirilganda paydo bo'ladi. Tugalarga biriktirilgan izohlarni JButton klassining meros qilib olingan setToolTipText() metodi yordamida biriktirish mumkin.

```
saqlashButton.setToolTipText("Xujjatni saqlash");
```

Tugmalarga biriktirilgan izoh quyidagi oynada ko'rsatilgan:



*Tugmaga biriktirilgan izoh*

## 6-Mazvzu

### 6.1 Kontekst menyu

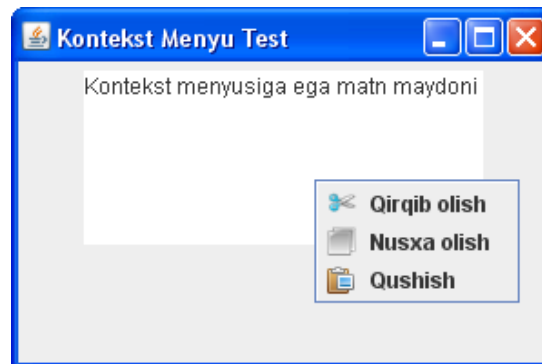
Kontekst menyu bu menyu qatoriga biriktirilmagan va foydalanuvchi tegishli sichqon tugmasini bosganda sichqon ko'rsatkichi joylashgan joyda paydo bo'luvchi menyusi xisoblanadi. Kontekst menyu JPopupMenu klassi tomonidan yaratiladi. Kontekst menyu oddiy menyu singari yaratiladi. Birinchi navbatda menyuning o'zi yaratilib keyin menyu elementlari yaratilib menyuga qo'shiladi.

```
JPopupMenu kontekstMenu = new JPopupMenu();  
JMenuItem qirqibOlish = new JMenuItem("Qirqib olish");  
kontekstMenu.add(qirqibOlish);
```

Kontekst menyu biron bir komponentga biriktiriladi. Kontekst menyusini paydo bo'lishi uchun foydalanuvchi ushbu elementga kontekst menyusini chiqarib beradigan sichqon tugmasini bosish kerak bo'ladi. Windows va Linux operatsion tizimida bu sichqonning asosiy bo'lmagan tugmasi, ya'ni o'ng tugma xisoblanadi. Masalan, kontekst menyu panel, matn maydoni, matn qatori, ro'yxat va boshqa elementlarga JComponent klassidan meros qilib olingan setComponentPopupMenu() metodi yordamida biriktirilishi mumkin.

```
matnMaydoni.setComponentPopupMenu(kontekstMenu);
```

Quyidagi oyna matn maydoniga biriktirilgan kontekst menyuni ko'rsatib beradi:



*JPopupMenu klassi yordamida yaratilgan kontekst menyu*

Ushbu oyna quyidagi dastur yordamida yaratilgan:

```
import java.awt.*;  
import javax.swing.*;  
public class KontekstMenyuTest {  
    public static void main(String[] args) {  
        JFrame frame = new JFrame("Kontekst Menyu Test");  
        JPopupMenu kontekstMenu = new JPopupMenu();  
        Icon qirqibOlishIcon = new ImageIcon("cut.png");
```

```

JMenuItem qirqibOlish = new JMenuItem("Qirqib olish",
qirqibOlishIcon);
Icon nusxaOlishIcon = new ImageIcon("copy.png");
JMenuItem nusxaOlish = new JMenuItem("Nusxa olish",
nusxaOlishIcon);
Icon qushishIcon = new ImageIcon("paste.png");
JMenuItem qushish = new JMenuItem("Qushish",
qushishIcon);
kontekstMenyu.add(qirqibOlish);
kontekstMenyu.add(nusxaOlish);
kontekstMenyu.add(qushish);
JPanel panel = new JPanel();
JTextArea matnMaydoni = new JTextArea("Kontekst menyusiga
ega matn maydoni", 6, 20);
panel.add(matnMaydoni);
matnMaydoni.setComponentPopupMenu(kontekstMenyu);
frame.getContentPane().add(panel);
frame.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
frame.setSize(300,200);
frame.setVisible(true);}}

```

Oddiy menyu kabi kontenst menyu tanlanganda `ActionEvent` xodisasi xosil qilinadi va `ActionListener` interfeysini bajarib biron bir amal bajarib olish mumkin.

## 7-Mavzu

### 7.1 Dialog oynalari

Dialog oynalar foydalanuvchi interfeysining standart elementi xisoblanib foydalanuvchiga biron bir ma'lumot berish yoki foydalanuvchidan biron bir savol so'rash uchun ishlatiladi.

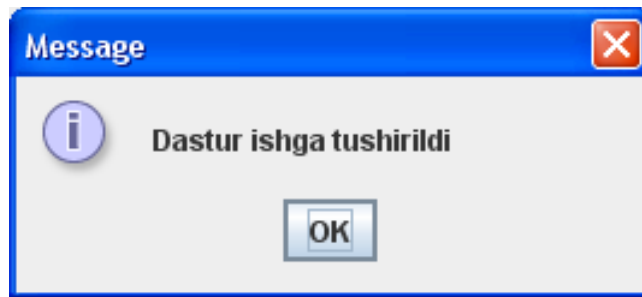
Dasturlarda dialog oynalar ko'p ishlatilganligi sababli Swing o'rami oldindan tayyorlangan dialoglar to'plamini o'z ichiga oladi. Dialog oynalar `JOptionPane` klassining statik metodlari yordamida yaratilishi mumkin. Ushbu klass dialog oynalarini to'rt guruhga ajratadi:

1. *Ma'lumot berish dialog oynalar* – foydalanuvchiga biron bir ma'lumot ko'rsatadi va odatda OK tugmasiga ega;
2. *Tasdiqlash dialog oynalar* – foydalanuvchidan biron bir savol so'raydi va odatda Yes, No, Cancel tugmalarga ega;
3. *Ma'lumot kiritish dialog oynalar* – foydalanuvchidan biron bir matn kiritishni so'raydi;
4. *Maxsus dialog oynalar* – istalgan elementni joylashtirish imkonini beradi.

Ma'lumot berish dialog oynasi `JOptionPane` klassining `showMessageDialog()` metodi yordamida yaratiladi. Masalan, quyidagi dastur ma'lumot berish dialog oynasini chiqarib beradi.

```
import javax.swing.*;
public class DialogTest1{
public static void main(String[] args){
JFrame frame = new JFrame();
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.setSize(300, 200);
frame.setVisible(true);
```

```
JOptionPane.showMessageDialog(frame, "Dastur ishga  
tushirildi");  
}}
```



*JOptionPane klassi yordamida yaratilgan ma'lumot berish dialog oynasi*

Tasdiqlash dialog oynasi `JOptionPane` klassining `showConfirmDialog()` metodi yordamida yaratiladi. `showConfirmDialog()` metodi foydalanuvchi bosgan tugmaga asosan butun son qaytaradi. Quyidagi dastur tasdiqlash dialog oynasini chiqarib, `tugmaLabel` yozuvini bosilgan tugma asosida o'zgartirib beradi.

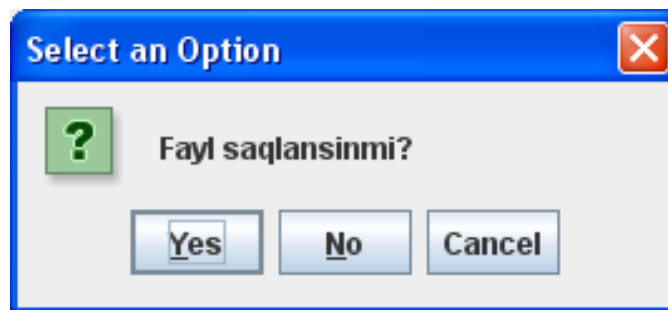
```
import javax.swing.*;  
public class DialogTest2{  
public static void main(String[] args){  
JFrame frame = new JFrame("Dialog Test 2");  
JPanel panel = new JPanel();  
JLabel label = new JLabel("Dialog oynasida bosilgan  
tugma: ");  
JLabel tugmaLabel = new JLabel(" ");  
panel.add(label);  
panel.add(tugmaLabel);  
frame.getContentPane().add(panel);  
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
frame.setSize(300, 200);  
frame.setVisible(true);
```

```

int i = JOptionPane.showConfirmDialog(frame, "Fayl
saqlansinmi?");
if(i==0) tugmaLabel.setText("Yes");
if(i==1) tugmaLabel.setText("No");
if(i==2) tugmaLabel.setText("Cancel");
}}

```

Dastur chiqarib bergan tasdiqlash dialog oynasi quyida keltirilgan:



*JOptionPane klassi yordamida yaratilgan tasdiqlash dialog oynasi*

Ma'lumot kiritish dialog oynasi `JOptionPane` klassining `showInputDialog()` metodi yordamida yaratiladi. `showInputDialog()` metodi foydalanuvchi kiritgan matnini qaytaradi. Quyidagi dastur ma'lumot kiritish dialog oynasini chiqarib, `matnLabel` yozuvini kiritilgan matn asosida o'zgartirib beradi.

```

import javax.swing.*;
public class DialogTest3{
public static void main(String[] args){
JFrame frame = new JFrame("Dialog Test 3");
JPanel panel = new JPanel();
JLabel label = new JLabel("Kiritilgan matn: ");
JLabel matnLabel = new JLabel(" ");
panel.add(label);
panel.add(matnLabel);
}
}

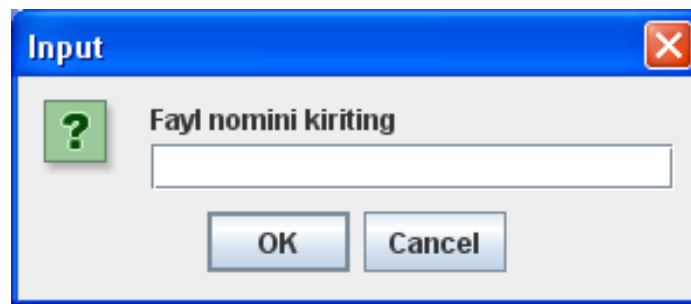
```

```

frame.getContentPane().add(panel);
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.setSize(300, 200);
frame.setVisible(true);
String s = JOptionPane.showInputDialog(frame, "Fayl
nomini kiriting");
matnLabel.setText(s);
}}

```

Dastur quyidagi ma'lumot kiritish dialogini chiqarib beardi:



*JOptionPane klassi yordamida yaratilgan ma'lumot kiritish dialog oynasi*

## 8-Mavzu

### 8.1 JOptionPane va JFileChooser klasslari yordamida oynalar yaratish

Maxsus dialog oynasi JOptionPane klassining showOptionDialog() metodi yordamida yaratiladi. Ushbu dialog oynasiga istalgan elementlarni joylashtirib olish mumkin. Quyidagi dastur avtorizatsia dialog oynasini yaratib beradi.

```

import javax.swing.*;
public class DialogTest4{
public static void main(String[] args){
JFrame frame = new JFrame("Dialog Test 4");
JPanel panel = new JPanel();
JLabel matnLabel = new JLabel(" ");

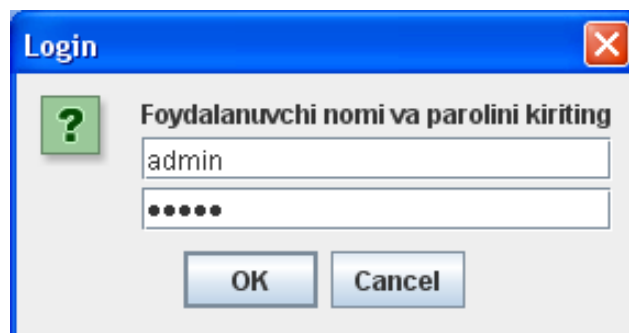
```

```

panel.add(matnLabel);
frame.getContentPane().add(panel);
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.setSize(300, 200);
frame.setVisible(true);
JTextField userField = new JTextField();
JPasswordField passField = new JPasswordField();
String message = "Foydalanuvchi nomi va parolini
kiriting";
JOptionPane.showOptionDialog(frame, new Object[]
{message, userField, passField}, "Login",
JOptionPane.OK_CANCEL_OPTION,
JOptionPane.QUESTION_MESSAGE, null, null, null);
String password = "";
for(char ch: passField.getPassword())
password = password + ch;
matnLabel.setText("Foydalanuvchi: " + userField.getText()
+ "; Parol: " + password);
}}

```

Dastur quyidagi dialog oynani chiqarib beradi:



*JOptionPane klassi yordamida yaratilgan va istalgan elementlarni joylashtirish mumkin bo'lgan dialog oynasi*

Dastur ishlashi davomida biron bir faylni ochish yoki saqlash extiyoji paydo bo'ladi. Ushbu maqsadda Java dasturlash tili fayllar bilan ishlovchi dialog oynasini yaratib beruvchi `JFileChooser` klassini taklif etadi. Faylni ochish dialog oynasini chiqarish uchun `JFileChooser` klassining `showOpenDialog()` metodidan, faylni saqlash dialog oynasini chiqarish uchun `showSaveDialog()` metodidan foydalanish mumkin.

```
JFileChooser chooser = new JFileChooser();
chooser.showOpenDialog(frame);
```

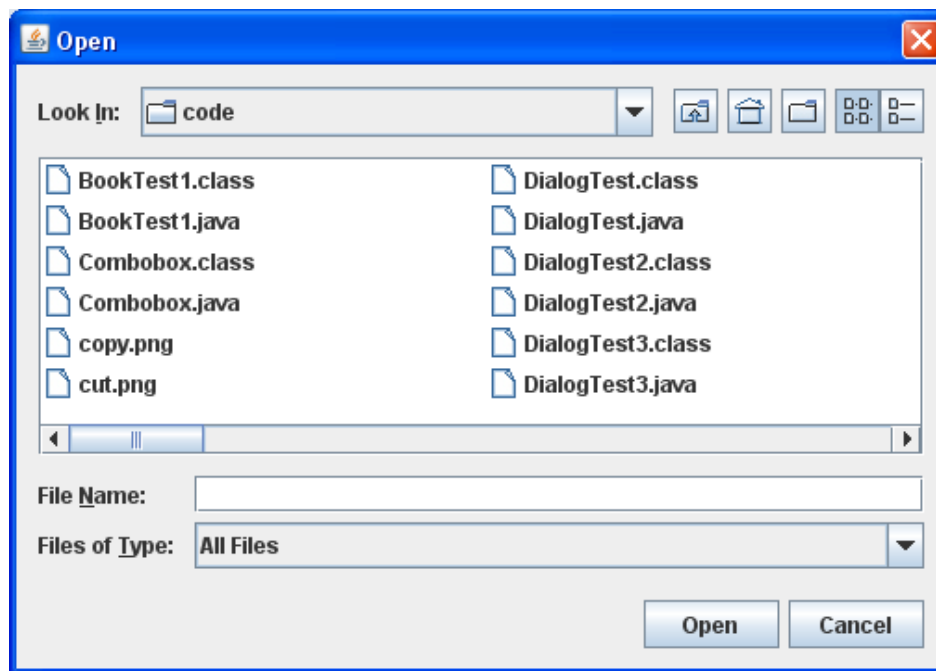
Fayl tanlash dialog oynasi ochilganda fayl tanlash uchun ma'lum papka ochilishi kerak bo'lsa `JFileChooser` klassining `setCurrentDirectory()` metodidan foydalanish mumkin.

```
chooser.setCurrentDirectory(new
    File("G:\\book\\code\\"));
```

Foydalanuvchi tanlagan faylni olish uchun `getSelectedFile()` metodidan foydalanish mumkin.

```
File file = chooser.getSelectedFile();
```

Quyida fayl tanlash dialog oynasi keltirilgan:



*JFileChooser klassi yordamida yaratilgan fayl tanlash dialog oynasi*

Ushbu fayl tanlash dialog oynasi quyidafi dastur tomonidan yaratilgan:

```
import java.awt.*;
import javax.swing.*;
import java.io.*;

public class FileTanlash{
public static void main(String[] args){
JFrame frame = new JFrame("File Tanlash");
JPanel panel = new JPanel();
JLabel label = new JLabel("");
JFileChooser chooser = new JFileChooser();
chooser.setCurrentDirectory(new
File("G:\\book\\code\\"));
panel.add(label);
frame.getContentPane().add(panel);
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.setSize(300,200);
```

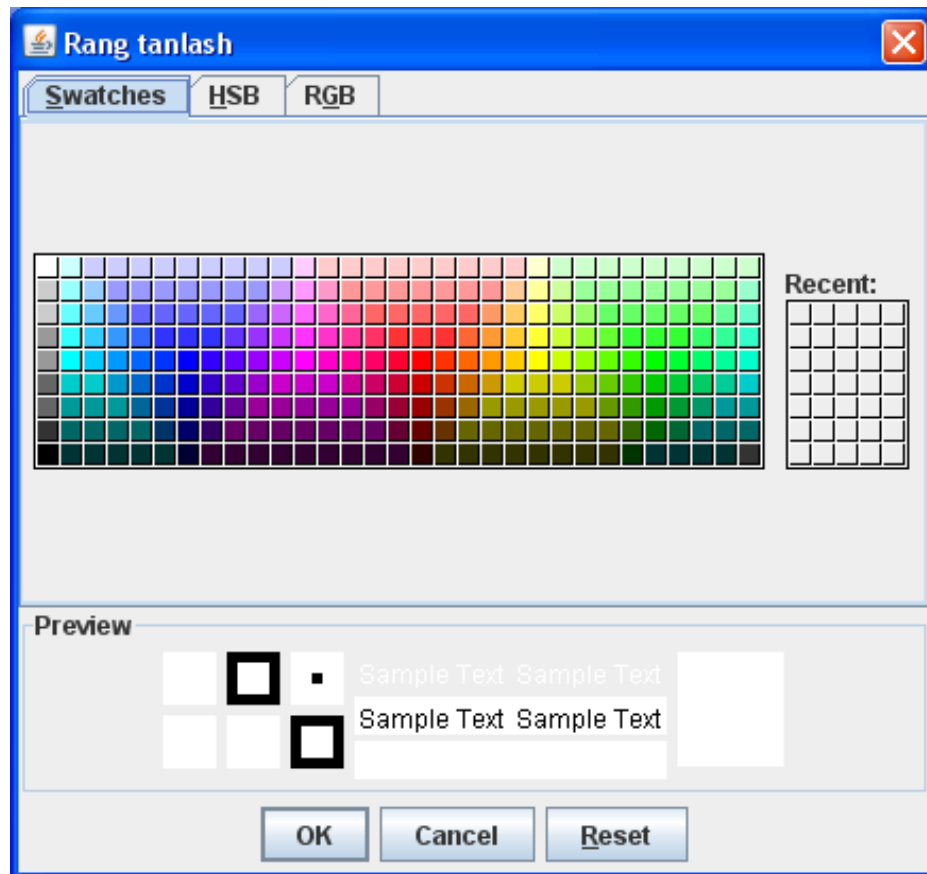
```
frame.setVisible(true);  
chooser.showOpenDialog(frame);  
String s = chooser.getSelectedFile().getPath();  
label.setText(s);  
}}
```

Bu yerda `Fayl` klassi diskda joylashgan faylni ifoda etadi. Fayl manzilini ko'rsatishda papkalarni ajratuvchi `\` simvoli `\\` ko'rinishda yozilishi kerak. `Fayl` klassi `java.io` o'ramida joylashgan bo'lib, ushbu o'ram dastur boshida import qilingan. `Swing` o'rami rang tanlash tayyor dialog oynasini xam taklif etadi. Rang tanlash dialog oynasini `JColorChooser` klassi yordamida yaratib olish mumkin. `JColorChooser` klassning `showDialog()` statik metodi dialog oynani chiqarib beradi.

```
Color c = JColorChooser.showDialog(frame, "Rang tanlash",  
null);
```

`showDialog()` metodi tanlangan rangni qiymat sifatida qaytaradi. Metodning uchunchi parametri joriy tanlangan rangni belgilaydi. Joriy belgilangan rang bo'lmasa `null` qiymati berilishi mumkin.

Quyida rang tanlash dialog oynasi keltirilgan:



*JColorChooser klassi yordamida yaratilgan rang tanlash dialog oynasi*

Ushbu oyna quyidagi dastur yordamida yaratilgan:

```
import java.awt.*;
import javax.swing.*;
public class RangTanlash{
public static void main(String[] args){
JFrame frame = new JFrame("Rang Tanlash");
JPanel panel = new JPanel();
frame.getContentPane().add(panel);
frame.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
frame.setSize(300,200);
frame.setVisible(true);
}
```

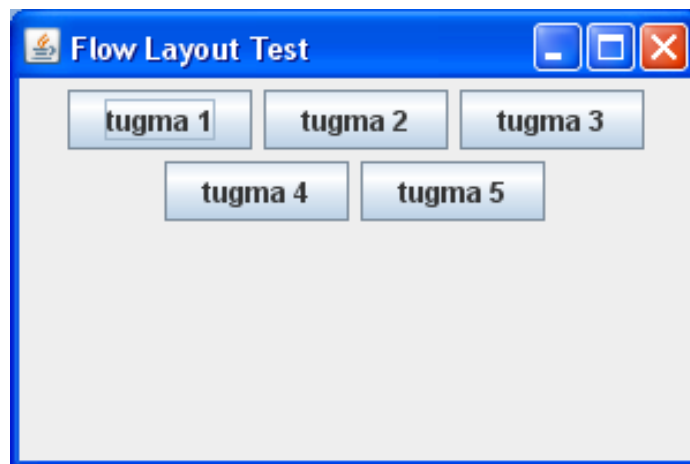
```
Color c = JColorChooser.showDialog(frame, "Rang tanlash",
null);
panel.setBackground(c);
}}
```

## 9-Mavzu

### 9.1 Joylashuv menedjerlari

Java dasturlash tilida foydalanuvchi interfeysi elementlarini panel yoki frameda joylashtirish usullarini joylashuv menedjerlari orqali belgilnadi.

Panelda elementlar avtomatik ravishda *flow layout* joylashuv menedjeri orqali joylashtiriladi. Ushbu menedjer elementlarni bir qatorga ketma-ket, o'rtaga tekislab joylashtiriladi. Agar elementlar bir qatorga sig'masa, sig'magan elementlar ikkinchi qatorga o'tkaziladi.



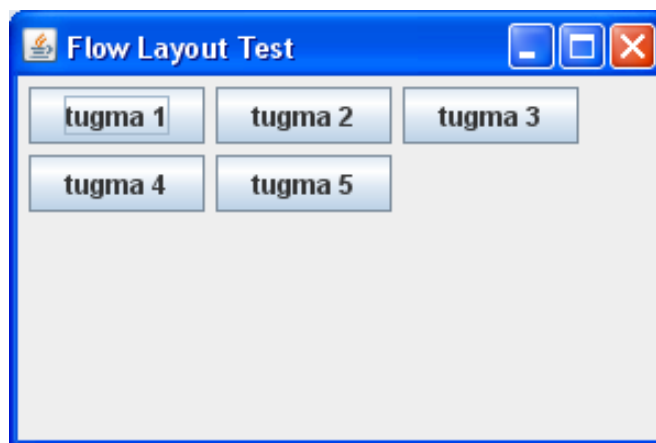
*Flow layout joylashuv menedjerining standart xolati*

Joylashuv menedjeri panelning `setLayout()` metodi orqali belgilanadi. Flow layout joylashuv menedjeri `FlowLayout` klassi yordamida yaratiladi. Panelda elementlar joylashuvini o'rtaga tekislashdan chapga tekislashga o'zgartirish uchun quyidagilarni bajarish kerak:

```
layout.setAlignment(FlowLayout.LEFT);
```

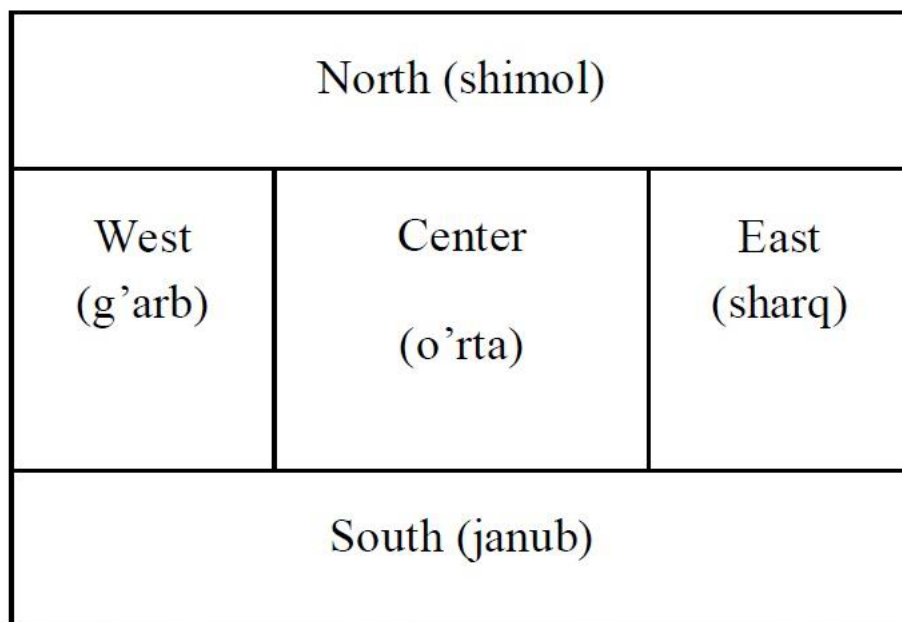
```
panel.setLayout(layout);
```

Natijada elementlar tekislanishi quyidagiga o'zgaradi.



*Flow layout joylashuv menedjerining chapga biriktirilgan xolati*

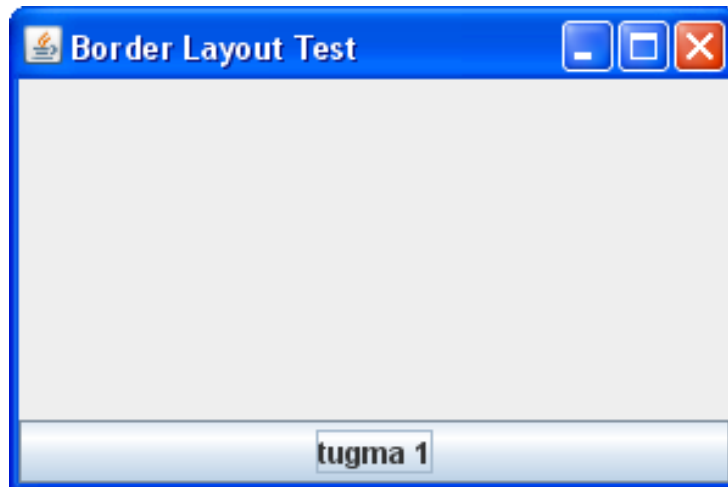
Boshqa joylashuv menedjeri bu *border layout* joylashuv menedjeridir. Ushbu joylashuv menedjeri flow layout manager joylashuv menedjeridan farqli o'laroq elementlarni panelni ma'lum joylariga joylashtirish imkonini beradi. Border layout joylashuv menedjeri panelni besh qismga (north, south, west, east) bo'lib, elementlarni ushbu qismlarga joylashtirish imkonini beradi.



*Border layout joylashuv menedjerining sohalari*

Border layout joylashuv menedjeri BorderLayout klassi tomonidan yaratiladi.

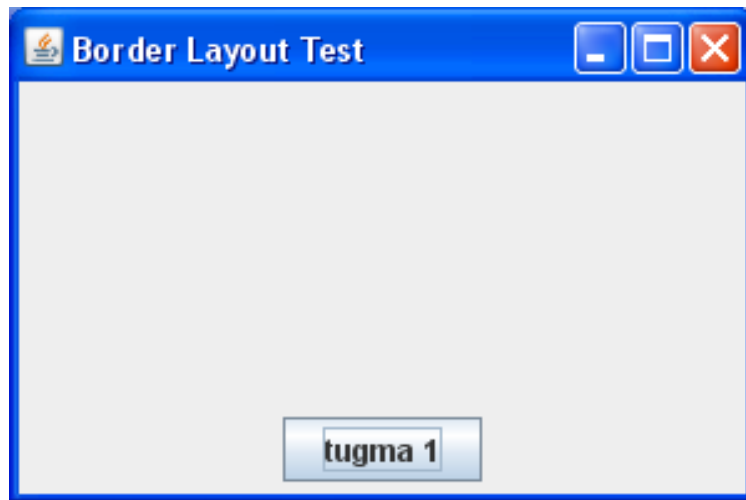
```
BorderLayout layout = new BorderLayout();  
panel.setLayout(layout);  
panel.add(button1, BorderLayout.SOUTH);
```



*Border layout joylashuv menedjerining janubiy qismiga joylashtirilgan element*

Flow layout joylashuv menedjeridan farqli o'laroq border layout joylashuv menedjeri joylashtirgan elementlarni asil o'lchamlarini mavjud maydonni to'ldirish uchun o'zgartiradi. Interfeys elementlarini asl o'lchamlarini saqlab qolish uchun ushbu elementlarni flow layout joylashuv menedjeri boshqaruvidagi panelga joylashtirib, keyin ushbu panelni border layout joylashuv menedjeri boshqaruvidagi panelning ma'lum qismiga joylashtirilishi mumkin.

```
JPanel ichkiPanel = new JPanel();  
ichkiPanel.add(button1);  
panel.add(ichkiPanel);
```

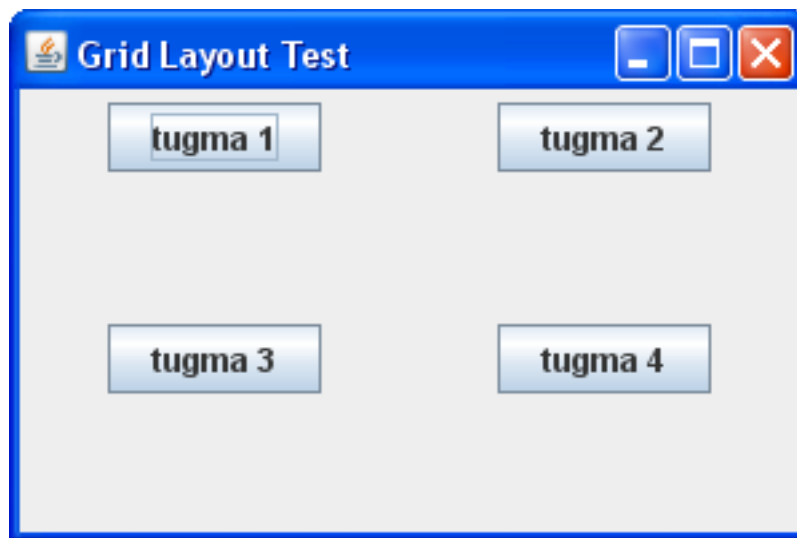


*Oldindan panelga joylashtirib olingan element*

Yana bir joylashuv menedjeri bu *Grid layout* joylashuv menedjeridir. Ushbu joylashtiruv menedjeri foydalanuvchi interfeysi elementlarini jadval sifatida qator va ustunlarga joylashtiradi. Jadval yacheykalari bir xil o'lchamga ega bo'ladi. Grid layout joylashuv menedjeri `GridLayout` klassi yordamida yaratiladi. Qator va ustunlar sonini klass konstruktori orqali berish mumkin. Border layout joylashuv menedjeri singari grid layout joylashuv menedjeri foydalanuvchi interfeysi elementlari o'lchamlarini mavjud joyni to'liq egallash uchun o'zgartiradi. Foydalanuvchi interfeysi elementlari o'lchamlarini saqlab qolish uchun ularni avval flow layout joylashuv menedjeri boshqaruvidagi panelga joylashtirish kerak.

```
GridLayout layout = new GridLayout(2, 2);  
panel.setLayout(layout);  
JPanel ichkiPanel1 = new JPanel();  
JPanel ichkiPanel2 = new JPanel();  
JPanel ichkiPanel3 = new JPanel();  
JPanel ichkiPanel4 = new JPanel();  
ichkiPanel1.add(button1);  
ichkiPanel2.add(button2);  
ichkiPanel3.add(button3);
```

```
ichkiPanel4.add(button4);  
panel.add(ichkiPanel1);  
panel.add(ichkiPanel2);  
panel.add(ichkiPanel3);  
panel.add(ichkiPanel4);
```

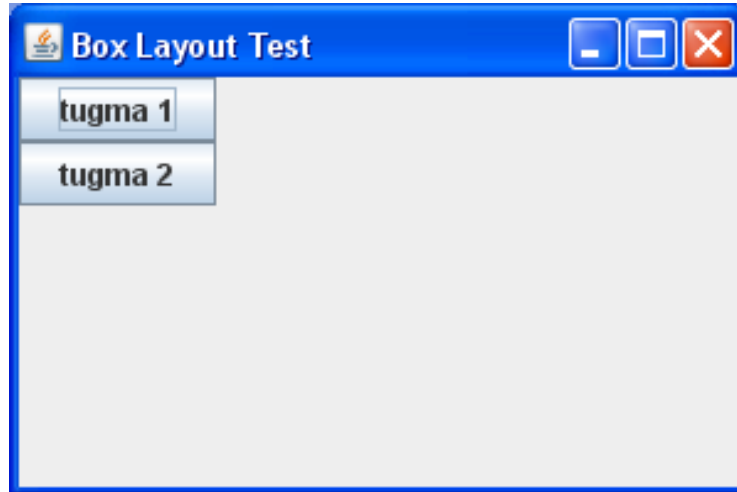


*Grid layout joylashuv menedjeri*

Yuqoridagi barcha joylashuv menedjerlari `java.awt` o'ramiga tegishli klasslar yordamida yaratiladi. `java.swing` o'rami ham joylashuv menedjerlariga ega. Bularning bittasi *Box layout* joylashuv menedjeridir. Ushbu joylashuv menedjeri foydalanuvchi interfeys elementlarini vertikal qator yoki gorizontal ustunga joylashtirish imkonini beradi. Box layout joylashuv `BoxLayout` klassi tomonidan yaratiladi.

```
BoxLayout layout = new BoxLayout(panel,  
BoxLayout.Y_AXIS);  
panel.setLayout(layout);  
panel.add(button1);
```

```
panel.add(button2);
```



*Box layout joylashuv menedjeri*

### **Takrorlash uchun savol va topshiriqlar:**

1. Frame deganda nima tushuniladi?
2. Panel nima uchun ishlatiladi?
3. Xodisa deganda nima tushuniladi?
4. Xodisalar qanday boshqariladi?
5. Tugalar qanday yaratiladi?
6. Yozuvlar deganda nima tushuniladi?
7. Tanlash elementlariga nimalar kiradi?
8. Checkbox va Radio tugma o'rtasida farq nima?
9. Matn kiritish uchun qanday grafik elementlardan foydalaniladi?
10. Menyu xosil qilish tartibi qanday?
11. Qanday dialog oynalari mavjud?
12. Joylashuv menedjerlarining vazifasiga nimalar kiradi?

## 10-Mavzu

### 10.1 Xatolik va istisnolar

Dastur ishlashi mobaynida ichki yoki tashqi sabablarga ko'ra xatoliklar yuzaga kelishi mumkin. Ichki sabablarga massivga murojat qilishda noto'g'ri indeksni ishlatish yoki obyekt biriktirilmagan obyekt o'zgaruvchisini ishlatish kabilar kiradi. Tashqi sabablarga noto'g'ri ma'lumot joylashgan fayl bilan ishlash yoki tarmoqdagi nosozliklar misol bo'la oladi. Agar dastur ishi xatolik tufayli to'liq bajarilmay qolsa, dastur quyidagi amallarni bajarishi lozim:

- xatosiz xolatga qaytish;
- foydalanuvchi ishini saqlab dastur ishini to'g'ri tugatish.

Istisnolarni boshqarish mexanizmining maqsadi boshqaruvni xatolik paydo bo'lgan joydan ushbu xatolik bilan ishlay oladigan dastur kodiga o'tkazishdan iborat.

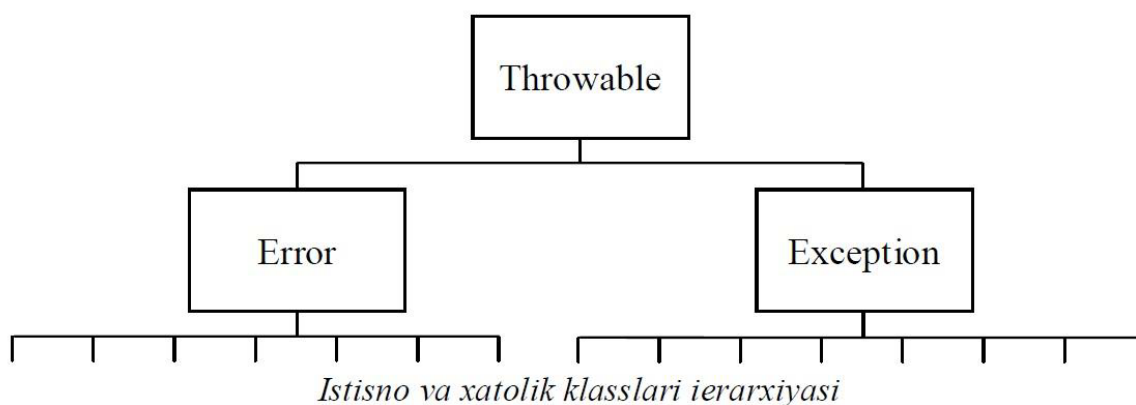
Istisnolar bilan ishlash uchun paydo bo'lishi mumkin bo'lgan xatoliklarni ko'rib chiqish kerak. Bular:

1. Foydalanuvchi kiritgan ma'lumotning xatoligi – kiritilgan matn xatoligi, noto'g'ri URL adresi, noto'g'ri son turi;
2. Qurilma xatoliklari – printer o'chiqligi yoki uning ishlashidagi xatolik, server ishlashidagi xatolik;
3. Resurslar cheklanishi – disk to'lib qolishi;
4. Dastur kodi xatoliklari – metodlarni noto'g'ri bajarilishi, mavjud bo'lmagan massiv indeksini ishlatish.

Java dasturlash tilida agar metod o'z ishini bajarish imkoni bo'lmasa (xatoliklar sababli) ushbu metod xatolik xaqida ma'lumotni o'z ichiga olgan obyektни xosil qiladi. Bundan tashqari, metodni chaqirgan dastur kodining bajarilishi to'xtatiladi va istisnolarni boshqaruvchi mexanizm dasturning istisnolarni boshqaruvchisini izlaydi.

Istisnolarni boshqaruvchi istisno obyektini qabul qilish va xatolik yuzaga kelganda bajariladigan amallarni belgilash imkonini beradi.

Java dasturlash tili obyektlarga ixtisoslashgan dasturlash tili bo'lganligi sababli istisnolar obyekt ko'rinishida bo'ladi. Barcha xatolik va istisnolar Throwable klassidan kelib chiqqan klasslar orqali xosil bo'ladi. Quyida istisnolar ierarxiya daraxtining yuqori qismi keltirilgan:



Throwable klassini ikkita, Error va Exception klasslari kengaytiradi. Ushbu klasslarni o'z navbatida aniq bir xatoliklarni ifoda etuvchi klasslar kengaytiradi.

Error klassi dasturning ichki xatoliklarini va resurslar cheklanishini belgilaydi. Ushbu xatoliklarni dastur yordamida to'g'irlash mushkul bo'lib, bu xolatda foydalanuvchini xatolik xaqida ogohlantirib dastur ishi tugatiladi.

Dastur tuzilayotganda Exception klassi va uni kengaytiruvchi klasslarga asosi e'tiborni qaratish kerak. Exception klassi o'z navbatida ikki turdagi klasslar tomonidan kengaytiriladi. Birinchi turdagi klasslar bu RuntimeException klassi va uni kengaytiruvchi klasslar. Ikkinchi turga boshqa klasslar kiradi.

Java dasturlash tilida Error va RuntimeException klasslaridan kelib chiqqan istisnolar *tekshirilmaydigan istisnolar* deyiladi. Barcha boshqa istisnolar *tekshiriladigan istisnolar* deyiladi. Tekshiriladigan istisnolarga dasturchi istisnolarni boshqaruvchi dastur kodini ta'minlab berishi kerak.

Metod amal bajarishi mobaynida uddalay olmaydigan xolat yuzaga kelsa istisno obykti xosil qilinadi. Agar metod bajarilishida istisno xosil bo'lish xolati mavjud bo'lsa metod e'lon qilish kodida ushbu xolat belgilanishi kerak. Istisno `throws` kalit so'zi bilan belgilanadi. Masalan, agar dasturdagi `faylniOchish()` metodi xotirada joylashgan faylni ochishi kerak bo'lsa, metod bajarilishi mobaynida ochilishi kerak bo'lgan fayl xotirada mavjud bo'lmasligi yoki mavjud fayl ochilmasligi mumkin. Ushbu xolda `IOException` istisnosi xosil bo'ladi.

```
public void faylniOchish() throws IOException
{
...
}
```

Istisnoni atayin xosil qilish xam mumkin. Istisno `throw` kalit so'zi bilan xosil qilinadi. Odatda bu dastur ishini tekshirish uchun amalga oshiriladi. Masalan,

```
IOException e = new IOException();
throw e;
```

## 11-Mavzu

### 11.1 Istisnolar bilan ishlash

Istisno xosil bo'lgandan keyin ushbu istisno ushlanishi kerak. Istisnoni ushlash uchun `try/catch` blokidan foydalaniladi.

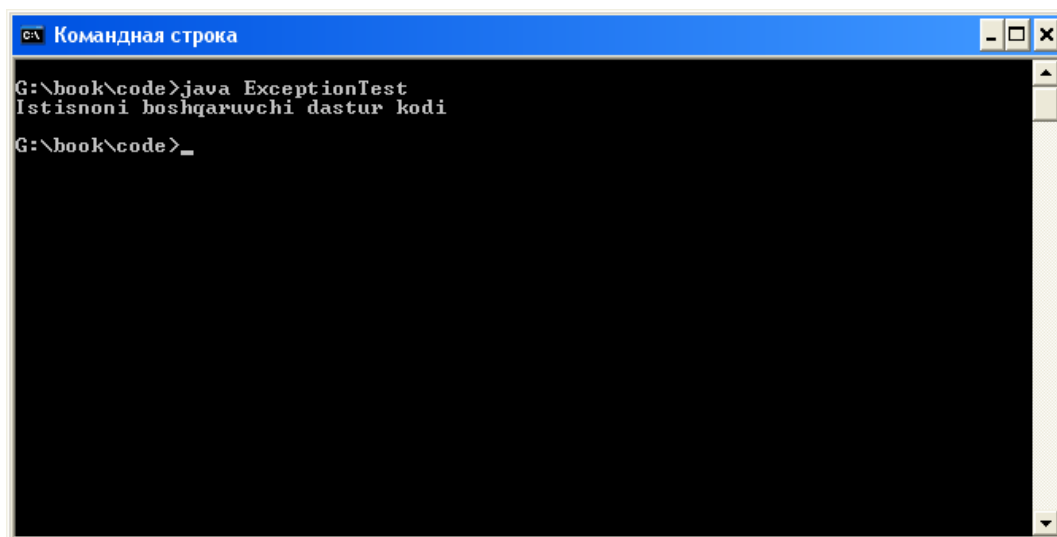
```
try {
//dastur kodi
//dastur kodi
} catch(ExseptionTuri e) {
//istisnoni boshqaruvchi dastur kodi
}
```

`try` bloki ichidagi dastur kodida istisno xosil bo'lsa dastur `try` bloki ichidagi qolgan dastur kod bajarilishi to'xtatilib `catch` bloki ichidagi istisnoni boshqaruvchi dastur kodi bajariladi, ya'no istisno ushlaniladi. Xosil bo'lgan istisno `catch()` ifodasida e'lon

qilingan istisno yoki uni kengaytiruvchi istisno turidan bo'lishi kerak. Masalan, quyidagi dastur try bloki ichida IOException istisnosi xosil qilinadi. Xosil qilingan istisno catch bloki tomonidan ushlanadi va istisno bajaruvchi kodi bajariladi.

```
import java.io.*;
public class ExceptionTest{
public static void main(String[] args){
try {
IOException istisno = new IOException();
throw istisno;
} catch(IOException e) {
System.out.println("Istisnoni boshqaruvchi dastur kodi");
}}}
```

Dastur komanda satriga quyidagi matnni chiqarib beradi:



```
Командная строка
G:\book\code>java ExceptionTest
Istisnoni boshqaruvchi dastur kodi
G:\book\code>_
```

#### *Istisnolarni boshqarish tizimi*

Agar try bloki ichidagi dastur kodida hech qanday istisno xosil bo'lmasa catch bloki bajarilmay ushbu blokdan keyingi dastur bajariladi. Agar xosil bo'lgan istisno catch() ifodasida ko'rsatilgan istisnodan farqli bo'lsa catch bloki bajarilmaydi va xosil bo'lgan istisno ushbu metodni chaqirgan dastur kodiga uzatiladi.

Metod xosil bo'ladigan istisnoni `try/catch` bloki bilan ushlashi yoki `throws` kalit so'zi bilan xosil bo'lishi xaqida e'lon qilinishi kerak.

### **Takrorlash uchun savol va topshiriqlar:**

1. Istisno deganda nima tushuniladi?
2. Istisnolar qaysi xollarda xosil bo'ladi?
3. Istisnolar qanday boshqariladi?
4. Xatolik deganda nima tushuniladi?

## **12-Mavzu**

### **12. Klass a'zolari va metodlarni ko'rinilishi**

Obyektlarga ixtisoslashgan dezaynni asosiy xususiyatlaridan biri bu ma'lumotni berkitish, ya'ni *enkapsulyatsiya* xisoblanadi. Enkapsulyatsiyaning asosiy maqsadi bu obyekt o'zgaruvchilarini tashqi obyektlarni to'g'ridan to'g'ri murojaatidan berkitish xisoblanadi. Bu bilan obyekt o'zgaruvchilariga noto'g'ri qiymatlarni berilishi oldi olinadi. Masalan, `Avtomobil` obyektini tezlik o'zgaruvchisiga manfiy qiymat berilishi. Enkapsulyatsiyaning yana bir afzalligi bu obyekt bajaradigan amallarini qay tarzda bajarilishini berkitishdir. Masalan, `Telefon` obyektini aloqaniUrnatish() metodi yordamida tarmoq orqali aloqani o'rnatish, ushbu metod qay tarzda aloqani o'rnatishi (ya'ni qanday tarmoqda, qaysi tarmoq protokolini qo'llab, ma'lumotlarni qay tarzda uzatib) tashqi muxitdan berkitiladi. Boshqa obyektlar `Telefon` obyektini yordamida aloqa o'rnatishi uchun aloqaniUrnatish() metodini chaqirishi lozim xolos.

Obyekt a'zolarini boshqa obyektlardan yashirish uchun o'zgartiruvchilar qo'llaniladi. Java dasturlash tilida to'rt turdagi o'zgartiruvchilar qo'llaniladi, bular:

1. `private`
2. o'zgartiruvchisiz (default)
3. `protected`
4. `public`

`private` o'zgartiruvchisi klass a'zosini boshqa klasslarga ko'rinmaydigan qilib qo'yadi. `private` o'zgartiruvchili klass a'zosiga faqatgina ushbu klassni boshqa a'zolari murojaat qilishi mumkin.

O'zgartiruvchisi bo'lmagan klass a'zosi klass joylashgan o'ramning boshqa klasslariga ko'rinadi. Ya'ni bunday klass a'zosiga murojaat qila olish uchun klasslar bir o'ramda bo'lishi lozim.

`protected` o'zgartiruvchisi klass a'zosini boshqa o'ramda joylashgan ushbu klassni ostki klassiga va klass joylashgan o'ramning boshqa klasslariga ko'rinarli qiladi. `public` o'zgartiruvchisi klass a'zosini barcha klasslarga ko'rinarli qiladi.

Quyidagi rasmda klass a'zolarini o'zgartiruvchisiga qarab ko'rinish darajasi keltirilgan.

*Klass a'zolarining o'zgartiruvchilari va ularning a'zo ko'rinishiga ta'siri*

Object **klassi**

Java dasturlash tilida barcha classlar `Object` klassini avtomatik ravishda kengaytiradi. Klass `Object` klassini kengaytirishi uchun ushbu klass nomidan keyin `extends Object` ifodasini yozish shart emas. Java ushbu ifodani avtomatik ravishda qo'shib qo'yadi.

`Object` klassi bir necha muxim metodlarga ega. Kengaytirish qoidasiga asosan barcha klasslar `Object` klassi a'zolarini (jumladan metodlarini) meros qilib oladi. Quyida `Object` klassini asosiy metodlarini ko'rib chiqamiz.

`equals()` metodi

`Object` klassining `equals()` metodi ikki obyektning tengligini tekshiradi. Metod quyidagi ko'rinishda e'lon qilingan:

```
public boolean equals(Object obj)
{
...
}
```

`equals()` metodi boshqa obyektning qabul qiladi va mantiqiy qiymatni (obyektlar teng bo'lganda `true`, aks holda `false` qiymatini) qaytaradi. `Object` klassida e'lon qilinishi bo'yicha `equals()` metodi tenglikga tekshirilayotgan obyektlar bir xil obyekt o'zgaruvchilari tomonidan murojaat qilinayotganligi tekshiriladi. Meros qilib olingan metodlarni qayta e'lon qilish imkoniyati mavjud bo'lganligi sababli, `equals()` metodini xam qayta e'lon qilib olish mumkin.

Masalan, agar ikkita kitob bir xil nom va varoqlar soniga ega bo'lsa ular bir xil xisoblansin. Ushbu me'zonga asosan tenglikni tekshiradigan `equals()` metodini quyidagicha qayta e'lon qilish mumkin:

```
public class Kitob{
    int varoqlarSoni;
    String nomi;
    public boolean equals(Object obj)
    {
        return nomi.equals(obj.name) && varoqlarSoni ==
        obj.VaroqlarSoni;
    }
}
```

Yuqorida qayta e'lon qilingan `equals()` metodi `Kitob` obyektlarini tengligini tekshirish uchun qayta e'lon qilingan.

hashCode() metodi

hashCode() metodi joriy obyektни maxsus kodini qaytaradi. Ushbu maxsus kod obyekt mazmuni asosida xisoblanadi. Object klassida e'lon qilinishi bo'yicha hashCode() metodi xar bir obyekt namunasiga, ya'ni bitta klass andozasi asosida yaratilgan obyektga, qaytarilmas sonni maxsus kod sifatida belgilaydi. Maxsus kod obyektlarni to'plamlarda saqlash uchun ishlatiladi.

toString() metodi

toString() metodi obyektни matn qiymatini qaytaradi. Ushbu metod asosan obyekt va uning xolati xaqida ma'lumot berish uchun qo'llaniladi. Masalan:

```
Scooter s = new Scooter();  
System.out.println(s.toString());
```

Yuqoridagi dastur kodi quyidagi malumotni ekranga chiqarib beradi:

```
Scooter@42e816
```

Meros qilib olingan boshqa metodlar singari toString() metodini qayta e'lon qilib olish mumkin. Masalan, toString() metodi quyidagicha qayta e'lon qilingan bo'lsin:

```
public class Scooter  
{  
    public String toString()  
    {  
        return "Scooter bu kichik mototsikl";  
    }  
}
```

Bunda,

```
Scooter s = new Scooter();  
System.out.println(s.toString());
```

dastur kodi quyidagi ma'lumotni ekranga chiqarib beradi:

```
Scooter bu kichik mototsikl
```

### **Takrorlash uchun savol va topshiriqlar:**

1. Ostki klass deganda nima tushuniladi?
2. Irsiyat degani nima?
3. Biror bir klassning ostki klassi qanday xususiyatga ega bo'ladi?
4. Interfeys deganda nima tushuniladi?
5. Interfeysni bajarish uchun nima qilish kerak?
6. Ichki klass deganda nima tushuniladi?
7. Ichki klass qanday xususiyatga ega?
8. O'ram deganda nima tushuniladi?
9. Qanday ko'rinish o'zgartiruvchilari mavjud va ular qanday vazifani bajaradi?
10. Object klassi qanday klass?

## GLOSSARIY

***Avtomatlashtirilgan axborot texnologiyasi (AAT)*** – boshqaruv vazifalarini hal etish uchun tizimli tashkil etilgan axborot jarayonlarini amalga oshirish usul va vositalari majmui.

***Avtomatlashtirilgan boshqarish tizim*** – iqtisodiy-matematik usullar, texnik vositalar (kompyuter, aloqa vositalari, axborotni tasvirlash qurilmalari va hokazo) va murakkab obyektни qo'yilgan maqsadga muvofiq oqilona boshqarishni tashkil etish.

***Avtomatlashtirilgan ish o'rni*** – yakuniy foydalanuv-chiga ma'lumotlarni ishlab chiqish va aniq muammoli sohada boshqaruv vazifalarini avtomatlashtirishni ta'minlovchi axborot, dasturiy va texnik resurslar majmui.

***Avtomatlashtirish*** – bu inson ish faoliyatini mashina va mexanizmlar bilan almashtirish, ya'ni energiya, materi-allar yoki axborotni qabul qilish, uzatish va foydalanish jarayonlarida insonni bevosita ishtirok etishdan qisman yoki to'la ozod etuvchi texnik vositalar, iqtisodiy-mate-matik usullar va boshqarish tizimlarining qo'llanishi.

***Adres*** – xotira qurilmasidagi axborot yoziladigan katakcha (yacheyka) larning nomeri.

***Algoritm*** – ma'lum bir turkumdagi hamma masalalarni yechishda ishlatiladigan amallar tizimining muayyan tartibda bajarilishi haqidagi aniq qoida.

***Algoritmik til*** – belgilar to'plami va bu belgilardan algoritmlarni yozish uchun mo'ljallangan til konstruk-siyalarini tuzish va ifodalash qoidalari tizimi.

***Amaliy dasturiy ta'minoti*** – bu foydalanuvchining aniq vazifalarini hal etish va umuman axborot tizimi-ning hisoblash jarayonini tashkil etish uchun mo'ljallangan bo'lib, operasion tizimlar imkoniyatlarini ken-gaytiruvchi paketlar, umumiy belgilanishdagi paketlar, avtomatik boshqarish tizimida ishlashga mo'ljallangan paketlardan iborat.

***Analog (uzluksiz) signallar*** – biror ma'lum vaqt oraliq'ida cheksiz ko'p qiymatga ega bo'ladigan uzluksiz signallar.

***Animasiya*** –obyektlar, kameralar, yorug'lik manbala-rini o'zaro joyini almashtirish yoki ularning para-metrlarini vaqt bo'yicha o'zgarishiga ega vazifa, topshiriq.

***Antivirus dasturi*** –kompyuter viruslarini aniqlash, ularni o'chirish va ulardan muhofazalash uchun dastur: detektor dasturi; doktor (davolovchi) dastur; revizor (taftish) dasturi; filtrlı dastur; vaksina yoki immunitet hosil qiluvchi dastur.

**Approksimatsiya** – matematik miqdorlarni ulardan soddaroq bo'lgan miqdorlar orqali taqribiy ifodalash.

**Arifmetik ifoda** – ixcham arifmetik yozuv; u talab qilinayotgan qiymatni olish uchun berilganlar bilan qanday arifmetik amallar bajarish kerakligini ko'rsatadi.

**Arxiv fayl** – siqilgan holda yagona faylga birlashtirilgan bir yoki bir necha faylning yagona ko'rinishi.

**Arxivlash dasturlari** – diskda joyni tejash maqsadida fayllar hajmini kichraytirishga imkon beruvchi dasturlar.

**Arxivni ochish** – arxiv faylda joylashgan fayllarni asl holatiga qaytarish.

**Arxivni yangilash** – arxivdagi fayllarning eskiroq versiyasi ustiga yangiroq versiyasini yozish.

**Audiokonferensiya** – biror firma yoki tashkilotning hududiy jihatdan uzoqda joylashgan xodimlari yoki bo'limlari o'rtasida kommunikasiyani saqlab turuvchi ikki tomonlama audioaloqa.

**Axborot (lot. tushuntirish, bayon qilish)** – og'zaki, yozma yoki boshqa usulda (masalan, shartli signallar, texnika vositalar va boshqalar yordamida) berilgan, fan tilida esa, odam bilan avtomat, avtomat bilan avtomat o'rtasida almashinuvchi hamda manbalari va taqdim etilishi shaklidan qat'iy nazar, shaxslar, predmetlar, faktlar, voqealar, hodisalar va jarayonlar to'g'risidagi ma'lumot.

**Axborot jamiyati** – ko'pchilik ishlovchilarning axborot, ayniqsa uning oliy shakli bo'lmish bilimlarni ishlab chiqish, saqlash, qayta ishlash va amalga oshirish bilan band bo'lgan jamiyatdir.

**Axborot zahiralari (resurslari)** – bu alohida hujjat va alohida hujjatlar to'plami, axborot tizimlari (kutubxona, arxiv, fond, ma'lumotlar banklari, boshqa axborot tizimlari) dagi hujjatlar va hujjatlar to'plamidir.

**Axborot izlash tizimi** – kerakli axborotni saqlash, izlash va kerak bo'lganda berish (chiqarish) uchun ishlatiladigan vositalar to'plami.

**Axborot madaniyati** – axborot bilan ishlashni, uni olish, qayta ishlash va uzatish uchun axborot tizimlari va texnologiyalaridan foydalanishni bilish orqali inson faoliyatining barcha jabhalariga oid boy bilimlar, ishonchli axborotlardan to'liq va o'z vaqtida foydalanishni ta'minlashga qaratilgan kompleks chora-tadbirlarni joylarda tadbir etishni anglatadi.

**Axborot mulkdori** – o'z mablag'iga yoki boshqa qonuniy yo'l bilan olingan axborot egasi, undan foydalanuvchi va uni tasarruf etuvchi yuridik yoki jismoniy shaxs.

**Axborot nazariyasi** – kibernetika sohasi, matematikaning axborotni saqlash, qayta ishlash va uzatish jarayonlarini o'rganuvchi bo'limi.

**Axborot sohasi** – sub'yektlarning axborotni yaratish, qayta ishlash va undan foydalanish bilan bog'liq faoliyati sohasi.

**Axborot ta'minoti** – avtomatlashtirilgan axborot tizimining eng muhim elementi sifatida boshqarilayotgan obyektning holatini ta'riflovchi va boshqaruv qarorini qabul qilish uchun asos bo'luvchi axborotlarni aks ettirish uchun mo'ljallangan bo'lib, tashkilotda aylanib yuruvchi axborotlarni tashkil etish shakli, joylashtirilish hajmi (axborotni tavsiflash va kodlashtirish, hujjatlarni unifikatsiyalashgan tizimi, axborot oqimlarining yagona tizimi) bo'yicha loyiha qarorlarining shuningdek, ma'lumotlar bazasi tuzilishi uslubining majmuidir.

**Axborot tarmog'i** – aloqa tizimlarida kompyuterlarning bir biri bilan bog'lanishi.

**Axborot texnologiyasi** – ob'yekt (axborot mahsuloti)ning holati, jarayon yoki hodisaning yangi xususiyati to'g'risidagi axborot olish uchun ma'lumotlarni to'plash, ularga ishlov berish va uzatish vositalari hamda usullari majmuidan foydalaniladigan jarayon, boshqacha aytganda, axborotlarni yig'ish, uzatish, to'plash, qayta ishlash, saqlash, taqdim etish va foydalanish uslublari va usullari tizimi; uning maqsadi – axborot ishlab chiqarish bo'lib, uni tahlil etish va uning asosida biror bir harakatga qo'l urish uchun tegishli qaror qabul qilish.

**Axborot texnologiyasi fani** – axborotlarni jamlash, saqlash, uzatish va shu jarayonlarni amalga oshiruvchi texnik vositalarni ishlatishni o'rgatuvchi fan.

**Axborot tizimi** – belgilangan maqsadga erishish uchun axborotlarni shakl va mazmuniga ko'ra turlarga ajratish, ularni saqlash, izlash va qayta ishlash prinsiplari, qayta ishlashda qo'llaniladigan usullar, shaxslar hamda vositalar (odam-kompyuter)ning o'zaro bog'langan majmui.

**Axborot tili** – axborot izlash va mantiqiy axborot tizimlarida semantik axborotni, fakt va ma'lumotlarni tavsiflash maqsadida (masalan, ularni EHM yordamida qayta ishlash uchun) foydalaniladigan sun'iy til.

**Axborot xavfsizligi** – axborot sohasida ma'lumotlarni yo'qotish va o'zgartirishga yo'naltirilgan tabiiy yoki sun'iy xossalari tasodifiy va qasddan ta'sirlardan har qanday tashuvchilarda (shaxs, jamiyat va davlat manfaat-lari miqiyosida) axborotning himoyalanganlik holati.

**Axborotga ishlov berish** – texnologik jarayoni belgilangan ketma-ketlikda bajariladigan operatsiyalar majmui: dastlabki axborotni to'plash, birlamchi ishlov berish va ro'yxatga olish (birlamchi tashuvchilarda qayd qilish); ma'lumotlar va

abonentlarning so'rovlarini aloqa kanallari bo'yicha uzatish; ma'lumotlar va so'rovlarni qabul qilish va hisobga olish; qabul qilingan axborotning ishonchliligini nazorat qilish va xatolarini bartaraf etish; ma'lumotlarni tashuvchi mashinalarga ko'chirish va ko'chirishning to'g'riligini nazorat qilish; ma'lumotlarni EHMga kiritish kiritish to'g'riligini nazorat qilish; ma'lumotlarga mashina ichida ishlov berish va chiqish axborotini tuzish; chiqish hujjatlarini tayyorlash va ularni foydalanuvchiga uzatish.

***Axborotga mashinada ishlov berish*** – foydalanuvchilarning so'rovlarini amalga oshirish uchun bajariladigan (izlash, saralashning foydalaniladigan usullari bilan) ishlov berish jarayonlari xarakteri bilan va axborot massivlarini yuritish jarayonlarini tashkil etilishi bilan belgilanadi.

***Axborotlarning tarkibiy birligi*** – rekvizitlarning mantiqiy o'zaro bog'langan jamlanmasi (butunligi).

***Axborotlashgan jamiyat*** – bu tuzilishi, texnik bazasi va kishilik salohiyati bilimlarni axborot resurslarga muvofiq aylanishi va passiv shakllardagi (kitoblar, maqolalar, patentlar va h.k.) faollarga (modellar, algoritmlar, dasturlar, loyihalar) aylantirish maqsa-dida ishlab chiqish uchun moslashtirilgan jamiyat.

***Axborotni kodlash*** – axborotni ma'lumot sifatida yozish va mazmunini aynan saqlab qolgan holda axborotni boshqa shaklga o'tkazish.

***Axborotni soxtalashtirish*** — haqiqiy axborotni soxta (qalbaki) axborotga o'zgartirish.

***Axborotni tutish*** — uzatilayotgan axborotni qo'lga kiritish.

***Axborotni uzatish kabeli*** – tarmoqda axborotni bir kompyuterdan boshqasiga uzatish uchun xizmat qiladigan o'tkazgich.

***Axborotni o'zgartirish*** — axborotni qo'lga kiritib, uni o'zgartirish.

***Axborotni himoyalash*** – axborot borasidagi xavfsizlikka tahdidlarning oldini olish va ularning oqibatlarini bartaraf etish chora-tadbirlari: shaxs, jamiyat va davlat-ning axborot sohasidagi xavfsizligiga tahdidlarning; axborotning maxfiylikni ta'minlash, tarqalishi, o'g'irlanishi, yo'qotilishining; axborotning buzib talqin etilishi va soxtalashtirilishining oldini olish.

***Axborotni himoyalash tizimi*** – axborotning zaif tomonlarini kamaytiruvchi va axborotga ruxsat etilmagan kirishga, uning chiqib ketishiga va yo'qotilishiga to's-qinlik qiluvchi tashkiliy, texnik, dasturiy, texnologik va boshqa vosita, usul va choralarning kompleksi.

***Autentifikatsiya*** – foydalanuvchi yoki axborotning haqiqiy ekanligi kafolati.

**Bayonnoma (protokol)** – kompyuterlar orasida ma'lumot-larni uzatish tartibi va formatini belgilovchi qoidalar majmui.

**Bayt** – xotiraning eng kichik birligi, sakkizta ikkilik raqamlardan (8 bitdan) iborat.

**Beysik (BASIC, ing. boshlovchilar uchun universal programmalashtirish tili)** – hisoblash algoritmlarini yozish uchun qo'llaniladigan algoritmik til.

**Belgili ma'lumot** – alifbo raqamli belgilar majmuidan iborat ma'lumot turi.

**Bilimlar bazasi** – fan sohasida to'plangan bilimlarni kompyuterda tasvirlash, murakkab vazifalar yechimini topish uchun tahlil va xulosalarni yuzaga keltiruvchi, qayta ishlangan axborotlarni saqlashga mo'l-jallangan modellar, qoidalar va omillar (ma'lumotlar) ning majmui bo'lib, u asosiy ekspert tizimi sanaladi.

**Bilimlar bazasini boshqarish tizimi** – ma'lumotlar bazasini yaratish, yuritish va foydalanishga mo'l-jallangan dastur va til vositalari majmui.

**Billing tizimi** – telekommunikatsiya operatorlari tomonidan abonentlarga hisob-varaqlar taqdim etish va boshqa xizmat yetkazib beruvchilar bilan o'zaro hisob-kitoblar uchun ishlatiladigan har bir mijoz uchun aloqa xizmatlarining qiymatini hisoblaydigan va barcha tariflar va boshqa qiymat tavsiflari to'g'risidagi axborotni saqlaydigan tizim; ularning bajaradigan operatsiyalar sikli *billing* deb ataladi.

**Bit (ing. bit – binary digit – ikkili raqam)** – axborot miqdori birligi bo'lib, «0» va «1» dan iborat.

**Blok-sxema** – algoritmnin tasvirlashning grafik usuli.

**Bosh kalit** – ma'lumotlar bazasida saralash ishlarining tez va aniq bajarilishiga imkon beradigan jadvalning bir ustuni.

**Boshqarish tizimi** – boshqarish subyektlari – boshqaruvchi tizimlar va boshqarish obyektlari – turli tabiatli murakkab dinamik tizimlar majmui.

**Boshqaruv** – insonning turli ob'ektlarga va atrof muhitda kechayotgan jarayonlarga ongli ravishda ta'sir etishi; u jarayonlarga belgilangan yo'nalish berish va istalgan natijalarni olish maqsadida amalga oshiriladi; tashkiliy, reja, hisoblash, tahlil, nazorat, rag'batlantirish kabi funksiyalar amalga oshirilganda qo'yilgan maqsadni ta'minlaydi.

**Boshqaruv obyektlari** – jahon iqtisodiyoti, mamlakat, uning mintaqalari, hududlarining iqtisodiyoti, sohalar, korxonalar iqtisodiyoti, iqtisodiy resurslar va mehnat vositalari ko'rinishidagi ishlab chiqarish omillari, mehnatning o'zi, mehnat jamoalari, fan-texnika va axborot salohiyati moliyalarini o'z ichiga oladi.

**But virus** — disklarning yuklovchi sektorini ishdan chiqaruvchi virus.

**Bo'lingan (taqsimlangan) ma'lumotlar banki** – muayyan hudud bo'ylab tarqalgan va o'zaro bir-biri bilan bog'langan bir nechta alohida ma'lumotlar bazasidan iborat bo'lgan axborot tizimi.

**Videokonferensiya** – biror firma yoki tashkilotning hududiy jihatdan uzoqda joylashgan xodimlari yoki bo'limlari o'rtasida kommunikasiyani saqlab turuvchi ikki tomonlama videoaloqa.

**Virtual kanal** – qayd etilgan yo'nalish bo'lib, u ushbu aloqa seansida barcha uyalarni bir foydalanuvchidan ikkinchisiga o'tkazadigan kommutatorlar portlari nomerlarining ketma-ketligidan iborat.

**Virtual ulanish** – axborotni uzatish uchun ikki yoki undan ortiq oxirgi qurilmalar o'rtasida muayyan tarzda tuzilgan muhit.

**Gipermatn** – boshqa matnli hujjatlarga yo'l ko'rsatuvchi matn.

**Gipermedia** – matndan tashqari multimedia imkoniyat-larini ham o'zida mujassamlashtirgan ma'lumotlarga yo'l ko'rsatuvchi hujjatlar.

**Global tarmoq**– dunyoning ixtiyoriy davlatidagi kompyu-terlarni o'zida birlashtirish imkoniga ega bo'lgan tarmoq, bunda yo'ldosh orqali aloqa kanallaridan foydalaniladi (internet).

**Grafyasagich** – chizish qurilmasi.

**Dastur** – qandaydir ijrochi, odatda avtomatik qurilma, EHM bajarishi lozim bo'lgan amallar (harakatlar) rejasi.

**Dasturiy ta'minoti** – avtomatlashtirilgan axborot tizimining elementi bo'lib, hisoblash texnikasi vosita-lari bilan ma'lumotlarni qayta ishlash tizimini yaratish va undan foydalanish uchun dasturiy va hujjatli vositalarni jamlash.

**Dasturlash** – dastur, amal (harakat) rejasini tuzish jarayoni.

**Dasturlash fani** – dasturlarni tuzish usullari va yo'llarini o'rgatuvchi fan.

**Dasturlashtirilgan o'qitish, ta'lim** – o'qitishning bir turi bo'lib, oldindan tuzilgan o'qituvchi dasturi bo'yicha olib boriladi.

**Dasturlashtirish tili** – axborot va uni hisoblash mashinasida qayta ishlash algoritmlarini tavsiflash uchun mo'ljallangan, inson bilan hisoblash mashinasi muloqatining rasmiy tili (Fortran, Beysik, Paskal, C, C++ va b.).

**Dasturli virus** — avtonom ravishda ishlovchi, boshqa dastur qismiga o'z – o'zidan qo'shiluvchi, ishga qodir va kompyuter tarmoqlari hamda alohida kompyuterlarda tarqalish xususiyatiga ega bo'lgan dastur.

**Detektor** — virusning signaturasi bo'yicha operativ (tezkor) xotira va fayllarni ko'rish natijasida viruslarni topuvchi dastur.

**Disklarni optimallashtirish** – diskdagi bo'laklarga ajralgan fayllarni diskning boshiga ko'chirib, ularning joylashishini tartibga keltirish.

**Diskretlash** – analog signallarni raqamli ko'rinishga o'tkazish jarayoni.

**Display** – axborotni elektron nurli trubka ekranida vizual tasvirleydigan qurilma.

**Doktor revizorlar** – fayl va diskning tizimli sohasidagi o'zgarishlarni aniqlash, o'zgargan fayllarni dastlabki holatiga qaytara oladigan virusga qarshi dasturlar.

**Yo'llovchi** – kompyuter tarmog'ida ma'lumotlarni manzilga yetkazuvchi dasturlar paketi.

**Yopiq axborot** – tarqatilishi faqatgina kafolati mavjud bo'lgan tashkilotlar tomonidan berilgan ruxsat asosida amalga oshiriladigan maxfiy axborot.

**Yordamchi diskni tayyorlab qo'yish** – kompyuterni ishga tushirishga mo'ljallangan tizimli diskni tayyorlash.

**Jarayon** – qo'yilgan maqsadga erishish uchun yo'naltirilgan hatti-harakatlarning muayyan majmui.

**Zamonaviy axborot texnologiyasi (kompyuter axborot texnologiyasi)** – bu kompyuterlarni qo'llash, foydalanuv-chilarning (dasturlash sohasida kasb egasi bo'lmagan-larning) axborot jarayonida faol ishtirok etishiga, «do'stona» foydalanuvchi interfeysning yuqori daraja-siga, umumiy va muammoviy belgilanishdagi amaliy dasturlar paketini keng qo'llashga, EHMning hisoblash tarmoqlari tufayli foydalanuvchini uzoqlashtirilgan ma'lumotlar bazalari va dasturlariga kiritishga asoslangan texnologiyadir.

**Ikkilik sanoq sistemasi** – asosi 0 va 1 bo'lgan pozitsion sanoq sistemasi.

**Integrallashgan dasturlar** – bir necha soha masalalarini yechishga mo'ljallangan amaliy dasturlar majmui.

**Intellekt** – insonning tafakkur yuritish qobiliyati (aql, ong).

**Intellektual axborotli izlash tizimlari** – ish joyidan turib bilimlar bazasidagi kerakli axborotni izlashga imkon beradigan tizimlar.

**Intellektual tizimlar** – insonning mantiqiy fikr-lash usulini qo'llagan holda masalani yechadigan tizimlar.

**Internet (Internet)** – xuddi shu nomdagi jamiyat tomonidan tuzilgan xalqaro axborot tarmoqlarining global tarmog’i bo’lib, u minglab lokal va mintaqaviy kompyuter tarmoqlarini birlashtiruvchi dunyo kompyuter tarmog’idir.

**Internetning axborotli qismi** – internet tarmog’ida mavjud bo’lgan turli elektron hujjat, grafik rasm, audioyozuv, videotasvir va hokazo ko’rinishidagi axborot-lar majmui.

**Internetning dasturiy ta’minoti** – tarmoqqa ulangan kompyuterlar va tarmoq, vositalarini yagona standart asosida muloqot qilish, ma’lumotlarni ixtiyoriy aloqa kanali yordamida uzatish darajasida qayta ishlash, axborotlarni qidirib topish va saqlash hamda tarmoqda axborot xavfsizligini ta’minlash kabi muhim vazifalarni amalga oshiruvchi dasturlar majmui.

**Internetning texnik tarkibiy qismi** – turli rusumdagi kompyuterlar, aloqa kanallari, tarmoq texnik vositalari majmui.

**Interfeys** – foydalanuvchining tili; xotiradagi axborotlar va foydalanuvchining qulay muloqotini ta’minlaydigan dasturiy imkoniyat; foydalanuvchining bilimni belgilaydi.

**Intranet** – internet texnologiyasi, dastur ta’minoti va bayonnomalari asosida tashkil etilgan, ma’lumotlar bazasi va elektron jadvallar bilan jamoa bo’lib ishlash imkonini beruvchi korxonaga yoki tashkilot miqyosidagi kompyuter tarmog’i.

**Informatika** – ilmiy axborotning tuzilishi va umumiy xossalarini, shuningdek axborotni yig’ish, saqlash, izlash, qayta ishlash, almashtirish, tarqatish va inson faoliyati-ning turli sohalarida foydalanish bilan bog’liq masalalarni o’rgatuvchi fan sohasi bo’lib, uning asosiy vazifasi – davlat boshqaruv organlarining, sanoat va tadbirkorlik hamda boshqa sohalardagi axborot ehtiyojlarini qondirish uchun moddiy-texnik bazani yaratishdir.

**Itologiya** – axborot texnologiyalari to’g’risidagi fan bo’lib, uning predmeti - axborot texnologiyalari, shuningdek, ularni yaratish va qo’llash bilan bog’liq bo’lgan jarayonlardir.

**Ishchi stansiya** – tarmoqqa ulangan shaxsiy kompyuter, foydalanuvchi shu orqali axborot resurslariga kirib boradi.

**Ishchi tizimlar** – katta miqdordagi ma’lumotlarni saqlash, izlash, murakkab hisoblashlar, modellashtirish, dasturiy ta’minotni rivojlantirishga xizmat qiladigan vositalar.

**Iqtisodiy kibernetika** – mashina texnologiyalariga asoslangan holda ijtimoiy-iqtisodiy jarayonlarni o’rganadigan fan.

***Iqtisodiy zahiralar*** – sanoat jamiyatida moddiy ishlab chiqarishga yo'naltirilgan zaxiralar; uning turlari: moddiy, tabiiy, mehnat, moliyaviy va energetik zaxiralar.

***Iqtisodiyot*** – o'zining namoyon bo'lishining xilma-xil ko'lamlari, shakllari, turlarida belgilangan tuzilishga ega, mo'ljallangan maqsadlarga erishishni ta'minlovchi vazifalar doirasida amalga oshirishga qaratilgan ko'plab kishilar ishtirokida harakat qiluvchi tizim.

***Kalit (klyuch)*** — kriptografik o'zgartirishlar algoritmi parametrlarining maxfiy holati

***Katalog*** — fayl nomlari, uning hajmi, tashkil etilgan sanasi va boshqa xossalari haqida ma'lumotlarni saqlaydi (papka).

***Kattalik*** – biror obyektning xarakterlovchi va uning muayyan bir nusxasi uchun berilgan sonli yoki matnli qiymatni belgilovchi ko'rsatkich.

***Kibernetik tizimlar*** – kibernetika fani shug'ullana-digan murakkab boshqarish tizimlari.

***Kibernetika*** – boshqarish tizimlari, shakllari va vositalari haqidagi hamda boshqarish tizimlariga oid umumiy qonunlarni o'rganadigan fan.

***Koaksial kabel*** – ma'lumotlar uzatishning ikkita turli tizimi: signalni modulyasiyalab (raqamli signal qanday shaklda uzatilsa, undan shunday shaklda foydalaniladi) va modulyasiyalamasdan (raqamli signalni analogli signalga aylantirib jo'natiladi va qabul qilishda aksincha; buni modem bajaradi) uzatishdan foydalanishga imkon beradi.

***Kod (fran. qonunlar majmui)*** – turli axborotni uzatish, qayta ishlash va saqlash uchun mo'ljallangan shartli belgilar tizimi, boshqacha aytganda, berilgan so'zlar to'plami bilan o'zaro bir qiymatli moslikka ega bo'lgan va biror-bir alifbo bo'yicha tuzilgan so'zlar to'plami.

***Kodlashtirish*** – axborotni shartli belgilar va qoidalar (*kodlashtirish tizimi*) yordamida belgilab, tartib bo'yicha uni bir tizimdan boshqa tizimga o'tkazish jarayoni.

***Kommunikatsion tizimlar*** – tarmoqdagi kompyuterlar orasida axborotlarni uzatish uchun marshrutlash va bog'lanishlarni kommutasiya qilish vazifasini bajaradigan tizimlar.

***Kompleks himoya tizimi*** — huquqiy, tashkiliy, muhandis – texnik va dasturiy – matematik elementlari mavjud bo'lgan himoya tizimi

***Kompyuter (ingl. Computer)*** – hisoblovchi, uning tur-lari: raqamli, analogli (uzluksiz), raqamli-analogli, maxsuslashtirilgan; uning guruhlarini: super kompyuter-lar (Super Computer); katta kompyuterlar (Manframe Computer); mini kompyuterlar

(Minicomputer); shaxsiy kompyuterlar (PC - Personal Computer); bloknot kompyuterlar (Notebook).

**Kompyuter virusi** – bu o'lchami katta bo'lmagan dastur yoki dastur kodidan iborat bo'lib, kompyuterda turli noxush amallarni bajarishga, masalan, boshqa dastur, hujjat yoki axborot tashish vositasining ma'lum bir qismiga o'rna-shib olib, ruxsat berilmagan o'zgartirishni amalga oshi-rishga mo'ljallab yozilgan dastur; uning asosiy turlari: dastur viruslari; yuklanuvchi viruslar; makroviruslar.

**Kompyuter dasturi** – kompyuter instruksiyalarining ketma-ketligi.

**Kompyuter grafikasi** – bu EHM yordamida obyektlar modellari va ularning tasvirlarini yaratish, saqlash va qayta ishlash demakdir.

**Kompyuter tarmoqlari** – kompyuterlarda o'zaro axborot almashish imkoniyatini beruvchi qurilmalar majmui.

**Kompyuterli modellashtirish** – hodisa va jarayonlarning modelini kompyuterda ko'rish va o'rganish.

**Konfidensial axborot** – maxfiy axborot darajasiga ega bo'lgan yopiq axborot.

**Konsentrator (HUB)** – tarmoqda kompyuterlarni o'zaro axborot almashinuvini ta'minlovchi maxsus qurilma.

**Kreker** – kompyuter tizimlariga mualliflashtirilmagan holda kiruvchi foydalanuvchi.

**Kriptografiya** – maxfiy xabar mazmunini shifrlash, ya'ni ma'lumotlarni maxsus algoritm bo'yicha o'zgartirib, shifrlangan matnni yaratish yo'li bilan axborotga ruxsat etilmagan kirishga to'siq qo'yish usuli bo'lib, ma'lumot-larni himoyalashning maxfiylik va yaxlitlilik muammo-larini hal qilishga yo'naltirilgan.

**Kriptologiya** — maqsadli xabarlarining maxfiyligini va haqiqiyiligini ta'minlanishi lozim bo'lgan sirli aloqalar sohasi.

**Kriptomustahkamlik** — yomon niyatli shaxslarning o'z maqsadiga erisha olmasligi va kriptotahlilchilarning shifrlangan axborotni o'qiy olmasligi kafolati.

**Kriptotahlil** — maqsadli kriptograflar tomonidan ishlab chiqilgan himoya tizimini ochish.

**Ko'rsatkich** – axborotlilikni saqlovchi minimal axborotlarning tarkibiy birligi.

**Lokal tarmoq** – bir binoda yoki bir-biriga yaqin binolarda joylashgan kompyuterlarda o'zaro axborot almashish imkonini beradi va kompyuterlarni mahalliy tarmoqqa birlashtiradi.

**Ma'lumotlarning tarmoqli modeli** – ma'lumotning boshqa pog'onadagi ma'lumot bilan ikki yoki undan ortiq marta bog'lanadigan turi.

**Mantiqiy ma'lumot** – bir-birini inkor qiluvchi TRUE (rost) - «1» yoki FALSE (yolg'on) - «0» qiymatlarini qabul qiluvchi ma'lumot turi.

**Marshrutlash** – kerakli manzilga axborotlar blokini uzatish yo'lini aniqlash jarayoni.

**Matematik model** – o'rganilayotgan obyektning matematik formula yoki algoritmi ko'rinishida ifodalangan xarakteristikalarini orasidagi funksional bog'lanish.

**Matematik ta'minot** – avtomatlashtirilgan axborot tizimining elementi bo'lib, axborot tizimida vazifalar-ni hal etishda foydalaniladigan axborotlarni qayta ish-lash algoritmi, modellari, matematik uslublari majmui.

**Material** – bu ko'rinadigan xususiyatlar (rang, faktura, aks ettirish, keskinlik yoki noaniqlik) ning kombinatsiyasi, ular yordamida bitta sirt boshqasidan farqlanadi.

**Matnli prosessor** – matnli hujjatlarni yaratish va ishlov berish uchun mo'ljallangan amaliy dasturiy mahsulot turi.

**Maqsad** – intilish predmeti, oldindan mo'ljallangan yakuniy g'oya, tizim harakatining kutilgan natijasi, tizim nima uchun faoliyat yuritayotganini belgilaydi.

**Mahalliy tarmoq** – bir tashkilot yoki ishlab chiqarish birlashmasi doirasida kompyuterlarni o'zida mujassam-lashtirgan tarmoq.

**Ma'lumotlar bazasi (MB)** – obyektlarning muayyan sinflarini tavsiflovchi axborot massivlarining majmui; kompyuterning uzoq, muddatli xotirasida saqlanayotgan berilganlar va ular ustida aniq, amallarni bajarishga imkon beradigan ma'lumotlar yig'indisi.

**Ma'lumotlar bazasini boshqarish tizimi (MBBT)** – ma'lumotlar bazasini yaratish va uni dolzarb holatda saqlab turishni amalga oshiradigan, shuningdek, turli foydalanuvchilarning ma'lumotlar bazasida saqlanayotgan axborotlardan o'z maqsadlari uchun foydalanish imkoniyat-larini ta'minlaydigan dasturlar majmui; ma'lumotlar bazasidan foydalanish uchun maxsus yaratilgan dastur.

**Ma'lumotlar bazasini boshqarishning relyasion tizimi** – ma'lumotlarning relyasion tizimini yaratish va undan foydalanish uchun mo'ljallangan ma'lumotlar bazasini yaratish tizimi.

**Malumotlar bazasini boshqarishning iyerarxik tizimi** – ma'lumotlarning iyerarxik tizimini yaratish va undan foydalanish uchun mo'ljallangan ma'lumotlar bazasini yaratish tizimi.

***Malumotlar bazasini boshqarishning tarmoqli tizimi*** – ma'lumotlarning tarmoqli tizimini yaratish va undan foydalanish uchun mo'ljallangan ma'lumotlar bazasini yaratish tizimi.

***Ma'lumotlar banki*** – ma'lumotlar bazasi, MBBT, shuningdek, ma'lumotlar bazasida amalga oshirilgan texnik vositalar majmui.

***Malumotlar modeli*** – axborotlarni ifodalovchi vositalar majmui.

***Ma'lumotlarni kiritish*** – ma'lumotni qayta ishlash yoki masalani yechish uchun zarur bo'lgan ma'lumotlarni va dasturni kompyuter xotirasiga kiritish.

***Ma'lumotlarni kodlash*** – cheklangan to'plam elementlaridan belgilangan tartib bo'yicha ma'lumotlarni tuzish jarayoni.

***Ma'lumotlarni tartiblash*** – ma'lumotlar qiymati va formatini foydalanish uchun qulay holatga keltirish jarayoni.

***Ma'lumotlarni uzatish*** – formal ko'rinishda ifodalangan (belgilar yoki uzluksiz funksiyalar ko'rinishida) va EHM yordamida qayta ishlashga mo'ljallangan axborotni uzatish masalalarini o'z ichiga oluvchi elektr aloqa sohasi.

***Ma'lumotlarni chegirish*** – axborotlar tizimida ko'rsa-tilgan shartni qanoatlantirmagan elementlarning ma'lumotlar bazasiga kiritmay qoldirish holati.

***Ma'lumotlarni chiqarish*** – axborotni qayd qilish yoki ko'z bilan kuzatish uchun uni hisoblash mashinasidan chetki qurilmalar (masalan, alfavit-raqamli yozish qurilmasi, printer, displey va boshqa)ga uzatish.

***Ma'lumotlarning iyerarxik modeli*** – ma'lumotlarning daraxtsimon joylashish modeli.

***Ma'lumotlarning relyatsion modeli*** – ma'lumotlarning jadval ko'rinishida saqlanishi.

***Maxfiy axborot*** – foydalanilishi qonun hujjatlariga muvofiq cheklab qo'yiladigan hujjatlashtirilgan axborot.

***Mehnat zaxiralari*** – jamiyatda ishlash uchun umumiy ma'lumot va kasbiy bilimlarga ega kishilar.

***Mijoz*** – server resurslaridan va xizmatidan foydalanuvchi ish stansiyasi, kompyuter yoki dastur.

***«Mizoj-server» modellari*** – tarmoqdagi kompyuterlar-ning o'zaro aloqa qilish texnologiyasi.

***Mintaqaviy tarmoq*** – biror tuman, viloyat yoki respublika miqyosidagi kompyuterlarni o'zida mujassamlash-tirgan tarmoq.

**Model** (lot. *o'lchov, namuna, me'yor*) – biror obyekt yoki obyektlar tizimining obrazi, namunasi yoki xususiyat-larining belgilangan guruhi.

**Modellashtirish** – bilish obyektlarini ularning modellari yordamida tadqiq qilish, mavjud predmet va hodisalarning modellarini yasash va o'rganish.

**Modem** – bu modulyasiya va demodulyasiya so'zlaridan olingan bo'lib, uzluksiz signallarni raqamli (modulyasiya) va raqamli ma'lumotlarni uzluksiz (demodulyasiya) signalga almashtirib beruvchi maxsus elektron qurilma.

**Moddiy zahiralalar** – ijtimoiy mahsulot ishlab chiqarish jarayonida foydalanish uchun mo'ljallangan mehnat resurslari yig'indisidir, masalan, xom-ashyo materiallar, yoqilg'i energiya, yarim tayyor mahsulotlar, detallar, pul mablag'lari va h.k.

**Monitor** (lot. *eslatuvchi, tasvirlovchi*) – axborotni kineskop ekranida sifatli aks ettirish uchun qo'llani-ladigan qurilma.

**Multimedia** (*multimedia - ko'pvositalilik*) – kompyu-terning axborotlarni rangli grafika, matn va grafikda dinamik effektlar, ovozlarning chiqishi va sintezlangan musiqalar, animasiya, shuningdek to'laqonli videokliplar, hatto videofilmlar kabi turli xil ko'rinishlari bilan ishlash imkoniyati.

**Multimedia vositalari** – bu insonga o'zi uchun tabiiy muhit: tovush, video, grafika, matnlar, animasiya va boshqalardan foydalanib, kompyuter bilan muloqotda bo'lishga imkon beruvchi texnik va dasturiy vositalar majmuidir.

**Multimedia-texnologiya** – kompyuter bilan matnli, grafik, audio va video axborot almashinuvini dasturiy-texnik tashkil etish.

**Norezident virus** — to'raligicha bajarilayotgan fayl ichida saqlanadigan, maqsadga erishgandan so'ng operativ xotirada saqlanmaydiga virus

**Normallashtirish** – iyerarxik yoki tarmoqli tuzilishga ega bo'lgan ma'lumotlarni relyasion ko'rinishga o'tkazish jarayoni.

**Ob'yekt** – o'ziga o'xshash narsalardan ajralib turadigan alohida olingan predmet.

**Ob'yekt nusxasi** – obyektlar to'plamidan olingan muayyan obyekt.

**Ommaviy axborot** – cheklanmagan doiradagi shaxslar uchun mo'ljallangan hujjatlashtirilgan axborot, bosma, audio, audiovizual hamda boshqa xabarlar va materiallar.

**Operatsion tizim** – kompyuter qurilmalarini boshqaradigan dasturlari majmui; bu EHM zahiralarni boshqarish, amaliy dasturlarni chiqarish va ularning tashqi qurilmalar, boshqa dasturlar bilan o'zaro aloqasini amalga oshiruvchi, shuningdek,

foydalanuvchining kompyuter bilan muloqotini ta'minlovchi dasturiy vositalar yig'indisidir.

**Operatsiya (amal)** – hisoblash mashinasi berilgan kattaliklar ustida dasturning biror komandasi bo'yicha bajariladigan qandaydir amal; bir ish joyida bajariladigan muayyan harakatlar majmui.

**Optik tolali kabel** – mahalliy hisoblash tarmog'ida foydalaniladigan eng yangi texnologiya bo'lib, bunda axborot eltuvchi yorug'lik nuri bo'ladi, u tarmoq tomonidan o'zgartiriladi va signal shaklini oladi.

**Optimal boshqarish** – eng kam vaqt va mehnat yoki eng kam miqdordagi narsa va energiya sarflagan holda tizimni bir holatdan boshqa yangi holatga o'tkazish jarayoni.

**Paketlar** – ma'lumotlarning qismlarga bo'linishi.

**Parol bilan arxivlash** – begona foydalanuvchilar ochmasliklari uchun faylga parol qo'yib arxivlash.

**Plastik karta** – shaxsiy to'lov vositasi bo'lib, u mazkur vositadan foydalanadigan shaxsga tovar va xizmatlarni naqdsiz pulini to'lash, bundan tashqari bank muassasalari va bankomatlardan naqd pulni olishga imkon beradi.

**Port** – xabarlarni jo'natish yoki qabul qilish maqsadida mijoz va server tomonidan ishlatiladigan va, odatda, 16 bitli son bilan nomlanadigan dastur.

**Provayder** – kompyuterlarning tarmoqqa ulanish va axborot almashishini tashkil qiladigan tashkilot.

**Revizor dasturlar** – dastlab dastur va diskning tizimli sohasi haqidagi ma'lumotlarni xotiraga olib, so'ngra ularni dastlabkisi bilan solishtiradigan va mos kelmagan hollarda foydalanuvchiga ma'lum qiladigan virusga qarshi dasturlar.

**Rezident virus** — operativ xotirada saqlanadigan virus.

**Rekvizit** – axborotning moddiy tarkibiy birligi bo'lib, ma'no darajasida bo'linmaydi, muammo sohasining miqdoriy yoki mazmuniy ta'rifini aks ettiradi.

**Robot** – antronomorf (insonga o'xshash) harakatlar qiluvchi mashina.

**Sahifa (sayt)** – grafika va multimedia elementlari joylashtirilgan gipermedia hujjatlari ko'rinishidagi mantiqan butun axborot.

**Seleksiya** – tegishli manzillardagi axborotlarni saralash.

**Server** – tarmoq ishini ta'minlovchi va so'rovga ishlov beruvchi maxsus qurilma yoki kompyuter.

**Servis dasturiy ta'minoti** – bu foydalanuvchiga kompyuter bilan ishlashda qo'shimcha xizmatlar taqdim etuvchi va operasion tizimlar imkoniyatlarini oshiruvchi dasturiy mahsulotlar turkumidan iborat bo'lib, qobiqlar, utilitalar, antivirus dasturlari kabi guruhlarga bo'linadi.

**Signal kodi** – xabar mazmunini ifodalovchi sonlar majmui.

**Sonli ma'lumot** – ixtiyoriy sondan iborat ma'lumot turi.

**Standart dasturlar kutubxonasi** – oldindan tayyorlangan va kompyuter xotirasida saqlanadigan maxsus qism-dasturlar to'plami.

**Stenografiya** – axborotni shifrlash yo'li bilan unga ruxsatsiz kirishlarga to'siq qo'yish hamda maxfiy axborotning mavjudligini yashirish usuli.

**Sun'iy tillar** – tabiiy tillardan farq qilib, xalqaro munosabatlar uchun maqsadga yo'naltirilgan holda konst-ruksiyalanuvchi, axborotni EHM yordamida avtomatik qayta ishlash – dasturlashtirish tillari, fan va texnikaning ma'lum sohasidagi axborotni yozish (axborot tili) va boshqa tillar (esperanto, interling kabi).

**Sun'iy intellekt** – insonning ba'zi intellektual faoliyatini, mantiqiy, analitik tafakkurini modellashtiruvchi kibernetik tizimlarning shartli belgisi bo'lib, aqliy hatti-harakatlarga nisbatan kompyuter tizimining qobiliyati tushuniladi.

**Tabiiy zahiralari** – kishilarning moddiy va ma'naviy ehtiyojlarini qondirish uchun jamiyat tomonidan foydalaniladigan tabiat ob'yektlari, sharoitlari.

**Tarmoqlararo ekran** — umumiy tarmoqni ikki qismga ajratib, tarmoqlararo himoya vazifasini o'tovchi va ma'lumotlarni chegaradan o'tish shartlarini amalga oshiradigan qoidalar to'plamini o'zida saqlovchi tizim.

**Tarmoqli virus** — kompyuter tarmoqlarida tarqaluvchi virus.

**Tarmoqning axborot ashyolari** – arxiv, kutubxona, fondlar, ma'lumotlar bazasi va boshqa axborot tizimlaridagi hujjatlar yig'indisi.

**Tashkilot** – umumiy maqsadlar yo'lida birlashgan va umumiy moddiy hamda moliyaviy vositalardan moddiy va axborot mahsulotlarini hamda xizmatlarini ishlab chiqarish uchun foydalanadigan odamlar hamjamiyati yoki rasmiy ijtimoiy tuzilma.

**Taqribiy hisoblash** – ma'lumotlar va natija (yoki huch bo'lmaganda natijaning o'zi) mos kattaliklarning haqiqiy qiymatlarini faqat taqribiy ifodalangan sonlardan iborat bo'ladigan qilib hisoblash.

**Tashqi xotira** – bitta yoki bir nechta dasturni bajarishda hisoblash tizimi tarkibidagi juda katta hajmli axborotni uzoq muddat saqlash va undan bir necha marta foydalanish uchun mo'ljallangan xotira qurilmalarining kompleksi.

**Tashqi xotira qurilmasi (masalan, vinchester, fleshka, CD va DVD disklar va boshqa)** – u juda katta sig’imli (odatda u megabayt, gegabayt yoki terabaytlar bilan o’lchanadi) qurilma bo’lib, dastlabki ma’lumotlarni, hisoblash natijalarini, spravochnik ma’lumotlarini va boshqa ma’lumotlarni uzoq vaqt saqlash, ulardan takror foydalanish uchun mo’ljallangan.

**Telekommunikatsiya** – bu kompyuter tarmoqlari va zamo-naviy texnik aloqa vositalari negizida ma’lumotlarni masofadan uzatishdir.

**Terminal** – hisoblash tizimiga axborotni kiritish va undan axborotni chiqarish uchun qo’llaniladigan chetki qurilma.

**Texnik ta’minot** – avtomatlashtirilgan axborot tizimlarining elementi bo’lib, axborot tizimi ishi uchun mo’ljallangan texnik vositalar kompleksi, shuningdek, ushbu vositalar va texnik jarayonlarga tegishli hujjatlardir.

**Texnologiya** – san’at, mahorat, bilish degan ma’nolarni bildiradi (bu esa jarayondan boshqa narsa emas), ya’ni mahsulotni ishlab chiqarish jarayonida amalga oshiriladigan xom-ashyo, material yoki yarim tayyor mahsulot shakli, xususiyati, holatining o’zgarishi, uni qayta ishlash, tayyorlash usullarining majmui bo’lib, bu biror bir ishni yuqori darajada uddalash degani.

**Tijorat axboroti** – egasining ruxsati va shartlari bilan tarqatilishi va savdo–sotiq jarayonlarida ishtirok etilishi mumkin bo’lgan axborot.

**Tizim (sistema)** – yagona maqsad yo’lida bir vaqtning o’zi-da ham yaxlit, ham o’zaro bog’langan tarzda faoliyat ko’rsa-tadigan bir necha turdagi elementlar, qismlar majmui.

**Tizimli dasturiy ta’minot** – bu operasion tizimlar, servis dasturlar, dasturlashtirish tillari translyatorlari, texnik xizmat dasturlari bo’lib, kompyuterda axborotni qayta ishlash jarayonini tashkil etadi va amaliy dasturlar uchun me’yordagi ish muhitini ta’minlaydi.

**Tizimni boshqarish** – tizimni tashkil etuvchi element-larga nisbatan amalga oshiriladigan maqsadga yo’nalti-rilgan ta’sir.

**Tizimni himoyalash** – tizim dasturlari va ma’lumotlari-ga ruxsatsiz kirishdan yoki uning ishiga tasodifan ara-lashishdan saqlovchi chora–tadbirlarni o’tkazish jarayoni.

**Trafik** – tarmoqda uzatiladigan xabarlar oqimi.

**Uzoqlashgan terminal** – markaziy kompyuterdan uni bevosita ulash imkoniyatini istisno etadigan masofada uzoqda joylashgan kiritish-chiqarish qurilmasi.

**Uskunaviy dastur vositalari** – ma’lum bir yo’nalish-dagi masalalarni yechishga mo’ljallangan dastur qobiqlari.

**Fag** — zararlangan fayldan viruslarni chiqarib tashlab, faylna oldingi holatiga qataruvchi dastur.

**Fayl** – xotira qurilmalarida saqlanuvchi ma'lumotlar-ning bir qismi, boshqacha aytganda, ma'lum bir ma'lumot saqlanuvchi diskning nomlangan sohasi.

**Fayl viruslar** — \*.com, \*.exe kengaytmali fayllarni zararlovchi virus.

**Faksimil aloqa** – muayyan muammoni hal qilayotgan guruh ishtirokchilarining, ularning geografik joylashuvidan qat'iy nazar, faks apparatlaridan foydalangan holda hujjatlarni o'qish va uning tasvirini kommunikasiya kanalining ikkinchi uchida qayta tiklab olish hisobiga qarorlar qabul qilishi.

**Faktografik tizim** – sodda va qo'yilgan masalalarga yagona hamda aniq yechimni ko'rsata oladigan tizim.

**Fizik model** – tabiati va geometrik tuzilishi asl nusxa-dagidek bo'lib, miqdor jihatdan undan farq qiladigan model.

**Filtr** — viruslarnikiga xos harakatlarni topuvchi dastur.

**Filtr dasturlar yoki rezident dasturlar** – viruslar tomonidan zararni ko'paytirish va ziyon yetkazish mahsadida operasion tizimga qilinayotgan murojaatlarni ushlab qolish va ular haqida foydalanuvchiga ma'lum qilish vazifasini bajaruvchi virusga qarshi dasturlar.

**Filtrlash** – ma'lumotlarning aniq to'plamga taalluqliligini tekshirish.

**Filtrlovchi** — kompyuter tarmog'ida ma'lumotlarni manzilga yetkazuvchi, kirish – chiqish paketlarini filtrlovchi dastur.

**Foydalanuvchi interfeysi** – berilgan masalaga mos interfeysni tanlash.

**Foydalanuvchi muhiti** – interfeys tushunchasining boshqacha nomlanishi.

**Foydalanuvchini identifikasiyalash** – foydalanuvchi-ning ismi, sharifi, paroli yoki boshqa parametrlar bo'yicha tanib olish va ma'lumotlardan foydalanish bo'yicha unga berilgan huquqlarni aniqlash.

**Xabar** – axborotning biror moddiy ko'rinishda mujas-samlangan shakli.

**Xaker** – kompyuter tizimlarining mavjud barcha imkoniyatlarini aniqlashga qodir bo'lgan yuqori malakali dasturlovchi.

**Xost-kompyuter** – kompyuter tarmog'idagi fayllarga va kompyuter zaxiralariga kirish funksiyalarini bajaruvchi kompyuter.

**Xotira qurilmasi** – EHMning bir qismi yoki avtomatik rostdash va boshqarish tizimlarida, telemexanikada, dastur bilan boshqariluvchi agregatlarda asosan kod

shaklida berilgan axborotni yozish, saqlash va eshittirish uchun qo'llaniladigan mustaqil qurilma.

**Shaxsiy kompyuter** – bu hammaboplik va qo'llashda universallik talablarini qoniqtiruvchi, bir kishi foydalanadigan mikro-EHMdir.

**Shifr** – elementlari va kodlash tartibi cheklangan shaxslargagina ma'lum bo'lgan kod.

**Shifrlangan ma'lumot** – kriptografik himoya usuli qo'llanilgan ma'lumot.

**Shifrlash** – ma'lumotlarni maxsus algoritm bo'yicha o'zgartirib, shifrlangan matnni yaratish jarayoni.

**Shlyuz** – bayonnomani bir turdagi muhitdan ikkinchi turdagi muhitga o'tkazuvchi tarmoq qurilmasi.

**Ekspert** – bu, aniq muammo sohada samarali yechim va qarorlarni topa oluvchi mutaxassis.

**Ekspert tizimlar** – bu ayrim mavzu sohalarida bilimlarni to'plash va qo'llash, uyushtirish usullari hamda vositalari majmui bo'lib, xulosa chiqarish qoida va mexanizmlari yig'indisiga ega bo'lgan bilimlar bazasini o'z ichiga olgan suniy intellekt tizimidir.

**Elektron aloqa** – axborot tarmoqlari orqali foydalanuvchilarga xatlarni yetkazishni ta'minlashning muhim tarmoqli ko'rinishi.

**Elektron ofis** – muammo soha vazifalarini kompleks amalga oshirishni ta'minlovchi, ixtisoslashtirilgan dasturlar va axborot texnologiyalarini o'zi ishlab chiqara oluvchi amaliy dasturlarning integrasiyalangan loyiha-lari mavjudligini ko'zda tutadi.

**Elektron pochta (E-mail)** – kompyuterlardan tarmoqda foydalanishga asoslangan bo'lib, foydalanuvchiga tarmoq bo'ylab o'zining sheriklariga xabar yuborish, olish va saqlash imkoniyatini beradi.

**Elektron savdo** – bu global axborot tarmoqlari orqali mahsulotlarni sotish va pulli xizmatlar ko'rsatish bo'lib, uning asosiy turlari: axborotlar sotuvi; elektron do'konlar; elektron banklar.

**Elektron tijorat** – bu tadbirkorlik faoliyatining bir turi bo'lib, sotuvchi va xaridor o'rtasidagi shartnomalar hamda xizmatlar axborot texnologiyalari yordamida amalga oshiriladi; uning turlari: biznes-biznesga (B2B), V iste'molchiga (B2C), V ma'muriyatga (V2A), S2A, S2S, S2V.

**Elektron to'lovlar tizimi** – bu bank plastik kartalari-ni to'lov vositasi sifatida qo'llanilishidagi usullar va ularni amalga oshiruvchi subyektlar majmuasi.

**Elektron hisoblash mashinasi (EHM)** – bu hisoblash va axborot vazifalarini hal etish jarayonida axborotni avtomatik qayta ishlash uchun mo'ljallangan texnik vositalar kompleksidir.

**Energetik zahiralari** – energiya quvvati mavjud mahsulotlar, masalan, ko'mir, neft mahsulotlari, gaz, gidroenergiya, elektroenergiya va h.k.

**Yaxlitlik** — boshlang'ich ko'rinishda bo'lishligi va ruxsatsiz o'zgartirishlar qilinmaganligi kafolati.

**O'qitishni takomillashtiruvchi dastur vositalari** – mutaxassislar guruhi tomonidan ishlab chiqilgan va predmetdagi tegishli mavzular bo'yicha dars jarayonini tashkil qilishga imkon beradigan dasturlar.

**Qaror qabul qilish** – vaziyatni tahlil qilish, maqsad-larni aniqlash, bu maqsadga erishish dasturini ishlab chiqishga asoslangan boshqaruv obyektiga aniq maqsadga qaratilgan ta'sir etish hatti-harakati.

**Hisoblash mashinasi** - axborotlarni qayta ishlash (hisob) jaryonlarini mexanizasiyalashtirish va avtomatlashtirishga mo'ljallangan qurilma yoki qurilmalar majmui.

**Hisoblash texnikasi** – hisoblash jarayonlarini hamda axborot-mantiqiy masalalarni yechishni mexanizasiya-lashtirish va avtomatlashtirish vositalari majmui.

**Hisoblash tarmog'i** – uzatish kanallari orqali o'zaro bog'langan kompyuterlar majmui.

**Hisoblash tizimi** – hisoblash texnikasi majmui bo'lib, unga kamida ikkita asosiy prosessor yoki hisoblash mashinasi va tashqi qurilmalar (ma'lumotlarni kiritish-chiqarish, tashqi xotira qurilmalari, ichki va tashqi aloqa qurilmalari va boshqalar)ning rivojlangan sistemasi kiradi.

**Hujjat** – bu ma'lum bir qoidaga ko'ra rasmiylashtirilgan, belgilangan tartibda tasdiqlangan qog'oz, ovoz yoki elektron shaklida aks ettirilgan axborot yoki mustaqil ahamiyatga ega axborotlarning tarkibiy birligi.

**Hujjat aylanishi** – hujjatlarni yaratish, izohlash, uzatish, qabul qilish va arxivlashtirish, shuningdek ularning ijrosini nazorat qilish hamda ularni ruxsatsiz foydalanishdan himoyalash tizimidir.

**Hujjatlashtirilgan axborot** – identifikasiya qilish imkonini beruvchi rekvizitlari qo'yilgan holda moddiy jismda qayd etilgan axborot.

### Adabiyotlar

- [1]. David J.Eck. Introduction to Java. Geneva NY, 14456, (2016)
- [2]. Daniel Y.Liang. Introduction to Java programming, Comprehensive version. Princeton Hall 07458, New Jersey, (2011).
- [3]. Bruce Eckel. Thinking in Java. MidView Inc, New Jersey (2006).
- [4]. Шилдт Герберт. Java 8. Полное руководство; 9-е изд.: Пер. с англ. - М. : ООО "И.Д. Вильямс", (2015).
- [5]. Rashid Turg'unboyev. Java2 SE, Zamona
- [6]. B. Cole, R. Eckstein, J. Elliott, M. Loy, D. Wood. *Java Swing*, 2002.
- [7]. J. Zukowski. *The Definitive Guide to Java Swing*, 2005, ISBN (pbk): 1-59059-447-9
- [8]. C. Hortsman, G. Cornell, *Core Java 2*, Prentice Hall PTR, 2000.
- [9]. B.Cole, R. Eckstain, J. Elliot, M. Loy and D. Wood, *Java Swing*, O'Reilly &Associates, 2002.
- [10]. C. Walnut, *Java by Example*, Que Corporation, USA, 1996

### Elektron resurslar

1. <http://www.java.sun.com/javase/downloads/index.jsp>;
2. <http://www.eclipse.org/downloads/>;
3. <http://math.hws.edu/javanotes>
4. [www.oracle.com](http://www.oracle.com)
5. [www.gluon.com](http://www.gluon.com)

