

O'ZBEKISTON RESPUBLIKASI

OLIY VA O'RTA MAXSUS TA'LIM VAZIRLIGI

QARSHI DAVLAT UNIVERSITETI

“AMALIY MATEMATIKA VA INFORMATIKA” KAFEDRASI

“5130200 – Amaliy matematika va informatika” ta'lim yo'nalishi bo'yicha

bakalavr darajasini olish uchun

PRIMOV BOTIR TOSHEMIROVICH ning

« Grafika va matn bilan ishlashda JAVA tilining AWT

kutubxonasidan foydalanish»

mavzusida yozgan

BITIRUV MALAKAVIY ISHI

Ilmiy rahbar: o'q. F.Qlicheva

“Himoyaga tavsiya etildi”

Fizika-matematika fakulteti dekani

_____ prof. A.Tashatov

“ ___ ” _____ 2018-y.

QARSHI – 2018-yil

MUNDARIJA

KIRISH.....	3
I BOB. JAVA DASTURLASH TILI VA UNDA OYNALAR BILAN ISHLASH	6
1.1. Java dasturlash haqida boshlang'ich tushunchalar	6
1.2. Applet oynasini yaratish	10
1.3. Appletda matnlarni tahrirlash	12
1.4. <i>Swing</i> oynasini hosil qilish	18
1.5. Tugmalar qo'shish	20
II BOB. JAVADA GRAFIKA	22
2.1. Grafikani qo'llab quvvatlash	22
2.2. Grafik primitivlarni hosil qilish	23
2.3. Grafiklarning o'lchamlarini o'zgartirish	26
2.4. Ranglar bilan ishlash.....	27
III BOB. Mukammal dasturlar yaratishda Java tilining grafik va matnlar bilan ishlash imkoniyatlaridan foydalanish.....	33
3.1. Shakl harakatini boshqarishda Java tilining grafik imkoniyatlaridan foydalanish.....	33
3.2. Javada animatsion dasturlar yaratish	36
3.3. Sodda matn muharriri dasturini yaratish	43
3.4. Sodda ko'rinishdagi Paint dasturini yaratish	51
XULOSA.....	55
FOYDALANILGAN ADABIYOTLAR VA INTERNET RESURSLARI	57

KIRISH

O'zbekiston Respublikasi mustaqillik odimlarini dadil qo'yayotgan hozirgi davrda axborotlashgan jamiyat qurish masalasi mamlakatimiz uchun naqadar katta ahamiyat kasb etayotgani hech kimga sir emas. Mamlakatimiz rivojlanishining muhim sharti zamonaviy iqtisodiyot, fan, madaniyat, texnika, texnologiya rivoji asosida kadrlar tayyorlashning takomillashgan tizimining amal qilishiga erishishdir. "Ta'lim to'g'risida"gi Qonun va "Kadrlar tayyorlash milliy dasturi" ning qabul qilinishi bilan uzluksiz ta'lim tizimi orqali zamonaviy kadrlar tayyorlashning asosi yaratildi. Bu esa mamlakatimiz kelajagi uchun qurilgan poydevor hisoblanadi.

Ta'limning barcha bosqichlariga oid umumiy pedagogik va didaktik talab talaba (yoki o'quvchi)ning dasturiy bilim, tasavvur va ko'nikmalari asosida mustaqil ishlash samaradorligini takomillashtirish, ilmiy fikrlashga, o'quv faniga qiziqishini kuchaytirish, kasbiy bilimlarini chuqurlashtirish, nazariy va amaliy mashg'ulot mobaynida ularning faolligini oshirishdan iboratdir.

Axborotlashgan jamiyatda yashar ekanmiz, dasturlash tillariga bo'lgan ehtiyoj kundan kunga o'sib bormoqda. Java dasturlash tili bugungi kunga kelib ko'plab sohalarda qo'llanilmoqda, shuningdek bu til web ilovalar yaratishga qulay bo'lgan til hisoblanadi. Chunki u bizga sinf yaratib beradi, bu sinfni applet orqali bema'lol web ilovalarda ishlatish mumkin.

Mavzuning dolzarbligi: Zamonaviy dasturlash tillarining asosini tashkil etuvchi obyektga yo'naltirilgan dasturlash (OYD)- hozirgi kundagi dasturlashga bo'lgan yangicha yondashuvdir. Hisoblash texnikasining rivojlanishi va yechilayotgan masalalarni tobora murakkablashuvi dasturlashning turli modellarini (paradigmalarini) yuzaga kelishiga sabab bo'lmoqda. Birinchi kompilyatorlar (masalan, FORTRAN tili) dasturlashning funksiyalardan foydalanishga asoslangan protsedura modelini qo'llab quvvatlagan. Bu model yordamida dastur tuzuvchi bir necha ming qatorli dasturlarni yozishi mumkin edi. Rivojlanishning keyingi bosqichida dasturlarning strukturali modeli paydo bo'ldi va ular ALGOL, Pascal va C tillar kompilyatorlarida o'z aksini topdi. Strukturali dasturlashning mohiyati –

dasturni o'zaro bog'langan protseduralar (blokklar) va ular qayta ishlaydigan berilganlarning majmuasi deb qarashidir. Ushbu model dastur blokklarini keng qo'llashga, GOTO operatoridan imkon qadar kam foydalanishga tayangan va unda dastur tuzuvchi o'n ming qatordan ortiq buyruqlardan iborat dasturni yarata oladi. Yaratilgan dasturni protsedurali modelga nisbatan sozlash va nazorat qilish oson kechadi. Bularga qaramasdan yangi murakkab masalalarning paydo bo'lishi dasturlashdagi yangi yondashuvning paydo bo'lishiga olib keldi. Ya'ni murakkab masalalarni yechish uchun dasturlashning yangi uslubiga zarurat paydo bo'lib, u OYD modelida amalga oshirildi. Ushbu ishda esa OYD ning imkoniyatlaridan foydalanib grafika va matn bilan ishlash jarayonini qarab chiqamiz. Bugungi kunga kelib grafika va matn bilan ishlovchi ko'plab dasturiy vositalar mavjud bo'lsada, kompyuter bilan bog'liqlikda ishlovchi qurilmalarning mukammallashuvi grafika va matn bilan ishlovchi yangi dasturiy vositalarga ehtiyoj sezmoqda. Bunga misol qilib tibbiyot, kuzatuv obyektlari kabilarni aytish mumkin. Shu boisdan ham ishda qo'yilgan masala ahamiyatlidir.

Java tilining grafika va matn bilan ishlash imkoniyatlari ko'plab jahon olimlari tomonidan o'rganilgan. Xususan, Gerbert Shildt, Mayk Mak Grat, Key Xorstmann, Gary Kornell, N.Vyazovik, V.Paramonov, R.Setterlar tomonidan java tilining imkoniyatlari keng o'rganilgan.

Ishning maqsadi: Java dasturlash tilining grafika va matn bilan ishlash imkoniyatlarini o'rganish hamda ularni amaliy dasturlar qurishga tatbiq etish.

Ishda qo'yilgan vazifalar:

- Tilning grafika va matn bilan ishlovchi komponentalarini o'rganish;
- Bitli tasvirlar va multiplikatsiyalar bilan ishlashni batafsil o'rganish;
- amaliy dasturlarda grafika, matn, animatsiyalarni boshqarish jarayonlarini tashkil etish;

Tadqiqotning ilmiy yangiliklari :

Mukammal dasturiy vositalar qurishda Java tilining grafik imkoniyatlaridan foydalanish.

Tadqiqotning predmeti va obykti: Uzluksiv ta'lim tizimining barcha bo'g'inlariga tegishli bo'lgan multimediali elektron o'quv adabiyotlar, matn, animatsiya, ovoz va video fayllar bilan ishlovchi mukammal dasturlar.

Tadqiqotning ilmiy ahamiyati:

- mavzu bo'yicha ilmiy-uslubiy, nazariy adabiyotlarni o'rganish;
- ta'lim to'g'risida davlat hujjatlari, DTS talablari, ilg'or mutaxassis olimlarning fikrlarini o'rganish;
- mavzuga aloqador mavjud internet resurslaridan foydalanish;
- Java tilining grafika va matn bilan ishlash imkoniyatlarini amaliy dasturlar qurishga tatbiq etish.

Ishning hajmi va strukturasi: Bitiruv malakaviy ishi, kirish, uchta bob, xulosa, foydalanilgan adabiyotlar hamda internet resurslari ro'yxatidan iborat. Ishning 1-bobida applet oynasi, applet oynasida matnlarni tahrirlash, swing oynasi bilan ishlash va unda tugmalar joylashtirish haqida umumiy tushunchalar keltiriladi. 2-bobda Java tilining grafik imkoniyatlari amaliy masalalar yordamida tushuntiriladi. 3-bobda esa grafika, animatsiya va matn bilan ishlovchi mukammal dasturlarning matni va ulardan foydalanib olingan natijalardan namunalar keltiriladi.

I BOB. JAVA DASTURLASH TILI VA UNDA OYNALAR BILAN ISHLASH

Java dasturlash tili bugungi kunga kelib ko‘plab sohalarda qo‘llanilmoqda, shuningdek bu til web ilovalar yaratishga qulay bo‘lgan til hisoblanadi. Chunki u bizga sinf yaratib beradi, bu sinfni applet orqali bemaol web ilovalarda ishlatish mumkin.

Javaning AWT kutubxonasi tarkibida grafikani qo‘llab quvvatlovchi bir qator sinflar mavjud. Bunday sinflar applet oynasida yoki *Swing* paketi yordamida hosil qilingan oynalarda turli grafiklar chizish imkonini beradi. Bunday oynalar applet oynasi yoki dastur ilovasi oynasi ko‘rinishida bo‘lishi mumkin. Har bir oynaning koordinatasi quyidagicha bo‘ladi: Oynaning yuqori chap burchagida (0;0) nuqta joylashadi. *X* o‘qi shu nuqtadan boshlab oynaning yuqori o‘ng burchagi tomon yo‘naltirilgan bo‘ladi. *Y* o‘qi esa oynaning quyi chap burchagi tomon yo‘naltiriladi. Oynaning o‘lchami piksellarda aniqlanadi.

1.1. Java dasturlash haqida boshlang‘ich tushunchalar

Java dasturlash tili – eng yaxshi dasturlash tillaridan biri bo‘lib unda korporativ darajadagi mahsulotlarni(dasturlarni) yaratish mumkin. Java dasturlash tili Sun Microsystems kompaniyasi xodimlari James Gosling, Patick Naughton, Christ Warth, Ed Frank va Mike Sheridanlar tomonidan 1991 yilda yaratilgan. Uning birinchi ishga yaroqli versiyasini tayyorlash uchun 18 oy vaqt sarflangan. Bu dasturlash tilining birinchi nomi oldin ta’kidlab o‘tganimizdek, “Oak” deb nomlangan, so‘ngra, 1995 yilda “Java” nomiga o‘zgartirilgan.

Sun firmasining ma’lumotlariga qaraganda 3 mlrd. atrofidagi qurilmada javadan foydalaniladi. Ulardan ba’zilarini misol sifatida sanash mumkin:

- Shaxsiy kompyuter dasturlari (Desktop Applications) – acrobat reader, media-player, antiviruslar va h.k.;
- Web-dasturlar;
- Korxonatashkilotlar dasturlari (Enterprise Applications);

- bank yoki ishlab chiqarishga oid dasturlar;
- Mobil dasturlar;
- Smart kartalar;
- Robotlar;
- O'yinlar.

Asosiy konsepsiyalari:

Java quyidagi 5 maqsad uchun qurilgan, u shunday til bo'lishi kerakki:

1. Oddiy, obyektga mo'ljallangan, taqsimlangan va o'rganishga oson bo'lsin.
2. Mustahkam va xavsiz bo'lsin.
3. Qaysidir qurilma platformasidan yoki uning arxitekturasidan mustaqil bo'lsin (ya'ni qaysidir platformaga tobe bo'lmasin).
4. Juda samarali bo'lsin.
5. Dasturlash tili uchun tarjimon(interpreter) yozish mumkin bo'lsin. Shuningdek dasturlash tili parallel ishlashni va dinamik tiplashda foydalanishni ta'minlay olsin.



1.1-rasm. Java ko'rinishi

Java 1990-1991 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:

1. JDK Alpha va Beta (1995)
2. JDK 1.0 (23 Yan, 1996)
3. JDK 1.1 (19t Feb, 1997)

4. J2SE 1.2 (8 Dek, 1998)
5. J2SE 1.3 (8 May, 2000)
6. J2SE 1.4 (6 Feb, 2002)
7. J2SE 5.0 (30 Sen, 2004)
8. Java SE 6 (11 Dek, 2006)
9. Java SE 7 (28 Iyul, 2011)
10. Java SE 8 (18 Matr, 2014)

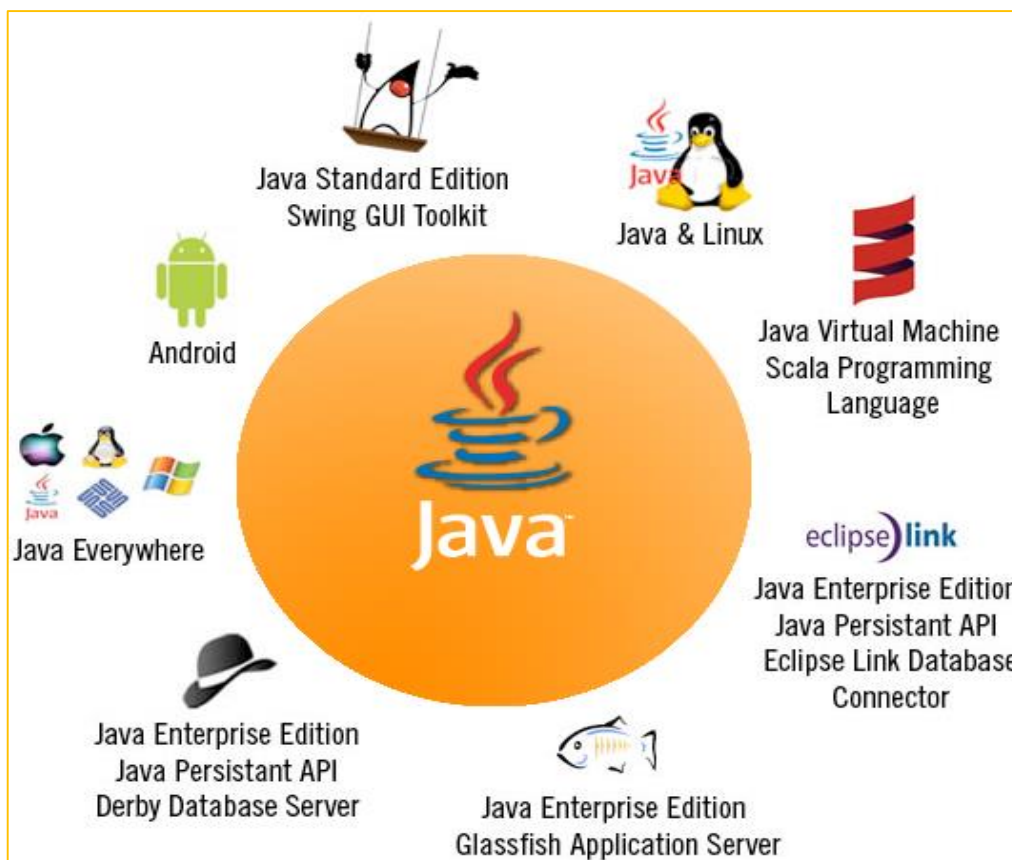
Java Obyektga Yo'naltirilgan Dasturlash (OOP-object oriented programming, OOP) tili va u C++ ga ancha o'xshash. Eng ko'p yo'l qo'yildigan xatolarga sabab bo'luvchi qismalari olib tashlanib, Java dasturlash tili ancha soddalashtirildi. Java texnologiyasi o'ta sodda, xavfsizlikni yuqori darajada ta'minlab 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 Javaning sintaksisi C++ ga asoslangan. Shuning uchun C++ tilini biladiganlar Javani oson o'rganishadi. Lekin undagi ko'pchilik xususiyatlar olib tashlangan. Masalan: Pointer(ko'rsatkich)lar bilan to'g'ridan to'g'ri ishlash, ya'ni Javada alohida ko'rsatkich tushunchasi yo'q. Operatorlarni qayta yuklash ham olib tashlangan. Yana eng muhimi, ishlatilmaydigan xotira (unreferenced objects) avtomatik tozalanadi. Buni Javadagi Garbage Collector (GC – chiqindi yig'ishtirgich) amalga oshiradi. C++ da bu destruktorlar orqali qo'lda (manual – ruchnoy) qilingan. Bundan tashqari Java har bir yangi versiyada bundan qulay imkoniyatlarni qo'shib kelmoqda.

Java 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. Shu bilan birga JavaSE barcha Java tillarining asosi hisoblanadi.

Java EE (Java Enterprise Edition) - Java texnologiyalari orasida eng ko'p tarqalgan tur hisoblanib, unda serverda ishlovchi dasturlar yaratiladi, masalan ko'p foydalanuvchili web-saytlar yaratishda keng qo'llaniladi, asosan internetda

ishlovchi dasturlar tuzishda foydalaniladi. 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 paketlarni, shuningdek *.jar ni ham o'z ichiga oladi. Bunday paketlar sirasiga Servlet, JavaMail, JSF(Java Server Face) va boshqa ko'pgina internetga asoslangan qo'shimcha paketlar kiradi.

Java ME (Java Micro Edition) - Java SE ning ba'zi qismlarini o'z ichiga oladi. Java ME yordamida kichik qurilmalar uchun dasturlar yozish mumkin, masalan, mobil telefon uchun o'yinlar kabi dasturlar yaratish mumkin. Java kompilyatori aytib o'tganimizdek, biz yozgan kodni bayt-kodga o'giradi. Odatda kompilyatsiyadan o'tgan dastur kodlari *.class kengaytmali faylda saqlanadi. Keyingi bosqichda kompilyatsiyadan o'tgan *.class kengaytmali fayl Java Virtual Machine(JVM) ga yuklanadi va JVM esa bu bayt-kodli faylni interpretatsiya qiladi, ya'ni mashina tiliga o'giradi, shu bilan birga undagi kodni imkoni boricha optimallashtiradi.



1.2-rasm. Java texnologiyasi.

Javada asosan 4 tipdagi dasturlar yoziladi:

1) **Standard applications** – Linux, Mac yoki Windowsga o‘rnatib, kundalik ishlatiladigan dasturlar: mp3 player, ofis, antivirus kabilar. Ular AWT, Swing yoki JavaFX texnologiyalari orqali tuziladi.

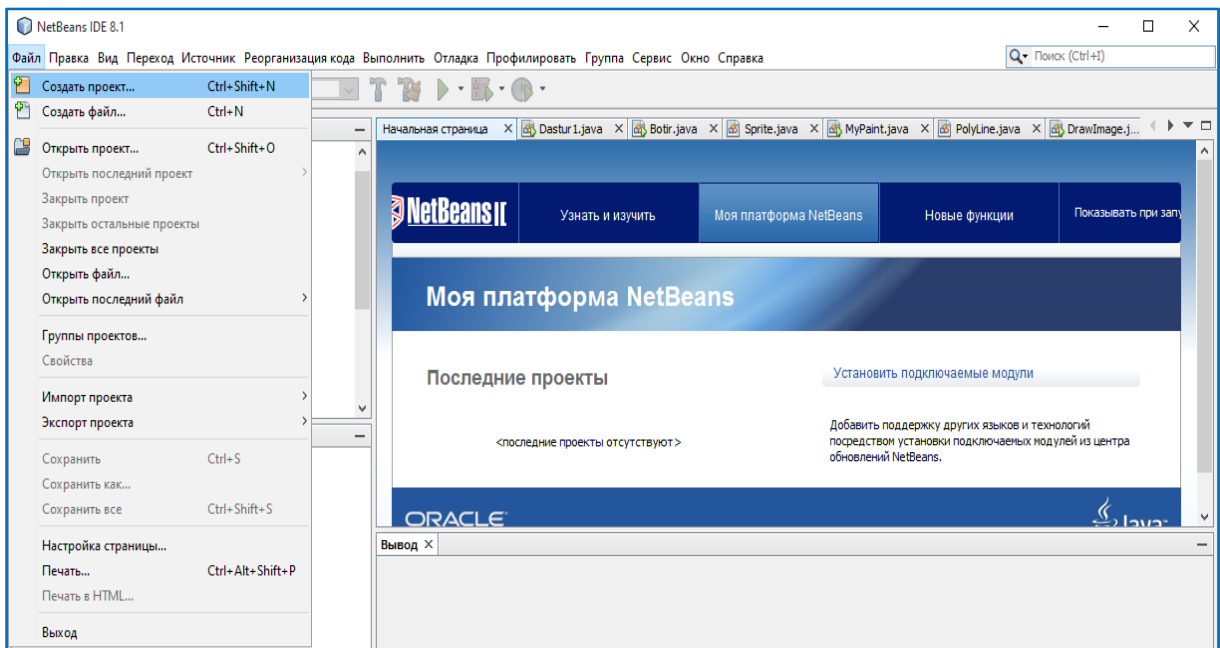
2) **Web Applications** – tarmoq orqali ishlovchi ixtiyoriy dasturlar. Web dasturlar ikki qismdan, server tomon hamda klient tomon (brauzer) dan iborat. Javada faqat server tomon uchun dasturlar yoziladi. Bunda servlet, jsp, jsf kabi fundamental texnologiyalardan boshlab *Swing*, *Play* kabi freymworklardan foydalaniladi. Umuman olganda brauzer uchun dasturlar HTML, CSS hamda Java Script kabi tillarda yoziladi.

3) **Enterprise Applications** – bu dasturlar yirik salmoqqa ega bo‘lib, odatda katta jabhalarda ishlatiladi. Banklar, tashkilotlar yoki astronomiya kabi sohalarda keng qo‘llaniladi. Ular yuqori xavfsizlik, yuklamani serverlarga teng taqsimlash (load balancing) yoki klasterlash (clustering – katta tizimdan xuddi yagona obyekt sifatida foydalanish) kabi sifatlarni talab qiladi.

4) **Mobile Applications** – Javada mobil qurilmalarga mos dasturlarni yozish imkoniyatini beradi. U Androiddan boshlab JME gacha qo‘llaniladi.

1.2. Applet oynasini yaratish

Javada applet yaratish uchun operatorlar ketma-ketligini “yig‘ish” lozim va bu chiquvchi fayl hisoblanadi. Chiqish faylini tuzish uchun bizga biror matn muhariri yoki *Eclipse* va *NetBeans* kabi IDE lardan biri kerak bo‘ladi. Applet yaratilgandan so‘ng u yechayotgan masalaga mos nom tanlanishi kerak. Misol uchun: appletingiz reklamani namoyish etish dasturi bo‘lsa, uning nomini “Shov_shuv” yoki “muhim_yangiliklar” deb nomlashingiz mumkin. Bunday nom bilan chiqish faylini ham nomlash mumkin. Misol uchun: Shov_shuv.java yoki muhim_yangiliklar.java. Endi biz “Salom. Bu applet oynasi!” xabarini chiqaradigan appletni tuzamiz. Buning uchun NetBeans dasturini ishga tushiramiz. Hosil bo‘lgan oynaning «Файл» menyusidan «Создать проект» bandini tanlaymiz:



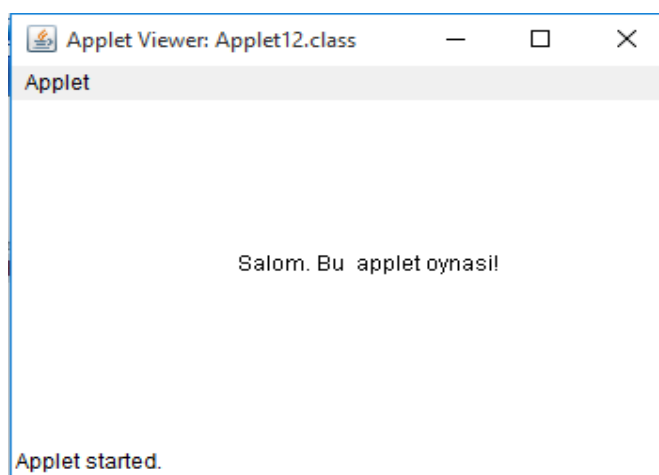
1.3-rasm NetBeans dasturida yangi loyiha ochish

So'ngra loyihaga nom beriladi va bu nom dasturdagi asosiy sinf nomi bilan bir xil bo'ladi. Applet loyiha tarkibidagi fayl hisoblanadi. Bitta loyiha tarkibida bir nechta fayllar bo'lishi mumkin. Shu tariqa applet yaratish uchun loyiha tarkibida fayl yaratamiz, buning uchun esa «Файл → Создать файл →Апплет» buyruqlar ketma-ketligini bajaramiz va appletga Applet12 deb nom beramiz, so'ngra esa hosil bo'lgan oynaga quyidagi dastur kodini kiritamiz:

```
import java.awt.*;
import java.applet.*;

public class Applet12 extends Applet
{
    public void paint(Graphics g)
    {
        g.drawString("Salom. Bu applet oynasi!",
            120, 100);
    }
}
```

Ushbu dastur kodini komplyatsiya qilganimizdan so'ng, applet oynasida quyidagi natija hosil bo'ladi:



1.4- rasm. Oddiy applet ko'rinishi

Appletni o'rganish jarayonida applet nomini ko'rsatgan operatorni ko'rasiz, quyida ko'rsatilgan kabi:

```
public class Applet12 extends Applet
```

Bilasizki, applet bajarilishi kerak bo'lgan masalani yechish uchun kerak bo'lgan operatorlardan foydalaniladi. Bundan tashqari Java obyektlar terminidan tuzilganini ham bilasiz. Siz applet yaratayotganingizda boshida Applet sinfini chaqiruvchi operator yozasiz. Keyin Applet sinfidan foydalanadigan appletni Applet12 ham ko'rsatiladi. Operatorning boshidagi `public` so'zi appletni brouzer ishga tushurishiga ruxsat beradi. Bu so'zni tashlab yozsangiz brouzer appletningi bajarishiga olmaydi. `class` so'zi esa kompilyatorga Applet12 nomni sinf tipidagi obyekt tuzilayotganligini anglatadi. Bu tarzda Applet sinfini kengaytirayotgan bo'lasiz. Java sinfini bu tarzda kattartirishingiz uchun yangi operatorlarni qo'llashingiz kerak.

1.3. Appletda matnlarni tahrirlash

Appletda shrifni boshqarish uchun *Font* obyektini ishlatasiz. *Font* sinfi *java.awt.Font* sinflar kutubxongasini aniqlaydi. Bu kutubxonadan foydalanish uchun, quyidagi import operatorni yozamiz:

```
import java.awt.*;
```

yoki

```
import java.awt.Font;
```

Appletingiz doirasida *Font* obyektini new operatorini ishlatib yaratash. *Font* obyekt yaratayotganingizda, *Font* funksiya-konstruktorlar shrift nomi ko‘rinishi va o‘lchamini uzatasiz:

Font font=new Font("TimesRoman", Font.BOLD, 18);

Bu holda Applet TimesNewRoman qalin, 18 punkt shrift yaratadi. Natijada siz ishlatadigan shriftlar soni oshadi. Ammo bugun brouzerlar HotJava yoki appletviewer shriftlari soni uncha katta emas. Shuning uchun 1.1-jadvalda ko‘rsatilgan shriftlar bilan cheklanash:

Shrift nomi	Misol	Java kodi
Arial	Sample text	font=new Font("Arial",Font.PLAIN, 18)
TimesRoman	Sample text	font=new Font ("TimesRoman", Font. PLAIN, 18)
Courier New	Sample text	font=new Font("Courier New", Font.PLAIN, 18)
Castellar	<i>Sample text</i>	font=new Font("Castellar", Font.PLAIN, 18)

1.1-jadval. Java ning asosiy shriftlari.

Bundan tashqari, *Font* obyeki yordamida shrift tipini tanlaysiz. Misol uchun, *TimesRoman* bundan tashqari u shrift atributlarini qaysini qalin, kursiv yoki ularning kombinatsiyasini ko‘rsatishga imkon beradi. Shrift ko‘rinishini ko‘rsatish uchun *Font.PLAIN*, *Font.BOLD* va *Font.ITALIC* dan atributlardan foydalanish. 1.2-jadvalda ko‘rsatilgan:

Shrift atributi	Misol	Java kodi
Font.PLAIN	Sample text	font=new Font ("Helvetica", Font.PLAIN, 18)

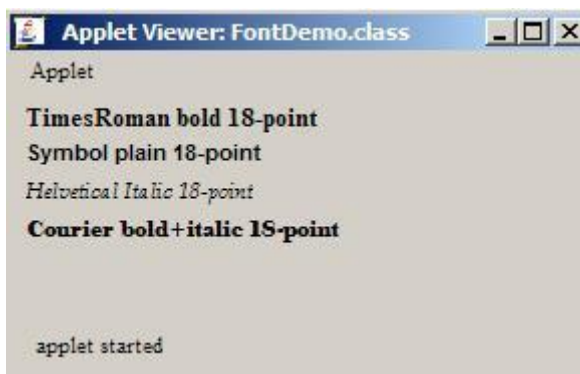
Font.ITALIC	<i>Sample text</i>	font=new Font("Helvetica", Font. ITALIC, 18)
Font.BOLD	Sample text	font=new Font("Helvetica", Font. BOLD, 18)

1.2-jadval. Java shriftlarining atributlari.

Yuqorida ko‘rsatilgani kabi shrift atributini ko‘rsatayotgarigizda siz belgilangan yuqori registirni ishlatishingiz lozim. Bu 1.2-jadvalda ko‘rsatilgan atributlardan tashqari Java appletlaringizga atributlar kombinatsiyasini “+” belgilarini ishlatib ishlatishga imkon beradi:

Font font=new Font ("Helveticaf", Font.BOLD+Font.ITALIC, 18);

Kompilyatsiya qilib, ishga tushirsangiz uning oynasida quyidagicha xabar chiqadi:



1.5-rasm. Appletda shriftlarni tanlash.

Metrik shriftlar haqida tushuncha. Dasturingizda tez-tez shriftlardan foydalaningiz shunday vaziyat bo‘lib qolishi mum- kinki, shrift haqida xabarlar, uning uzunligi, balandligi kerak bo‘lishi mumkin. Misol uchun, satrni applet oynasining o‘rtasiga chiqarish uchun uning enini piksellarda o‘lchamini bilishingiz kerak. Shrift haqida xabar chiqarish uchun, appletingizda quyidagi kabi *FontMetrics* obyektini yaratishingiz lozim:

Font font=new Font("Helvetica", Font.PLAIN, 18) ;

Font Metrics fontMetrics=get FontMetrics(font);

FontMetrics sinf metodni ishlatib, dasturingiz quyidagi kabi shriftlar xarakteristikasini aniqlash mumkin:

```

pixelWidth=fontMetrics.stringWidth("Java Now!");
charecterWidth=fontMetrics.charWidth('A');
Ascent=fontMetrics.getAscent();
Descent=fontMetrics.getDescent();
Loading=fontMetrics.getLoading();
Height=fontMetrics.getHeight();

```

Yuqorida ko'rib o'tilgan sinf obyektlaridan foydalanishni ko'rsatish maqsadida quyidagi dastur matnini keltiramiz:

```

import java.awt.*;
import java.applet.*;
    public class CenterString extends Applet
        {
            public void paint(Graphics g)
                {
int FontSize, y, window_width;
FontMetrics fontMetrics;
Font BigFont=new Font("TimesRoman", FontBOLD, 36);
Font LittleFont=new Font("TimesRoman", Font.PLAIN, 14);
window_width=size().width;
g.setFont(BigFont);
fontMetrics=g.getFontMetrics();
FontSize= fontMetrics.stringWidth("Java Now! Big");
g.drawString("Java Now! Big", (window_width-FontSize)/2,
150);
g.setFont(LittleFont);
fontMetrics=g.getFontMetrics();
FontSize= fontMetrics.stringWidthn("Java Now! Small");
g.drawString("Java Now! Small", (window_width-FontSize)/2,
150);
                }
        }

```

Ko'rib turganingizdek, applet *size().width* ni oynaning enini o'lchamini aniqlash uchun ishlatadi. Bundan keyin, bu funksiya *FontMetrics* sinfnining *StringWidth* funksiyasi pikselda satr uzunligini aniqlash uchun ishlatadi. Oyna va satr enining o'lchamini bilgan holda funksiya satrini oynaning o'rtasiga quyidagi chiqaradi:



1.6-rasm. Xabarni applet oynasining markaziga chiqarish.

Appletingizni qanday vazifa bajarishidan qat'iy nazar shunday hollar bo'lishi mumkinki, sizga shriftning xarakteristikalarini masalan: oddiy, qalin yoki kursivligini aniqlash kerak bo'lishi mumkin. Bundan tashqari xabarni chiqarish uchun funksiya shrift o'zgarishdan oldin, oldingi shriftni saqlashi lozim, u chunki keyinchalik tiklay olsin. Bunday axborotlarni aniqlash uchun Appletingiz Graphics sinf metodlari va Font dan quyidagi kabi ishlatishi lozim:

```
Font CurrentFont=g.getFont();  
String FontName=font.getName();  
int FontSize=font.getSize();  
int FontStyle=font.getStyle();  
boolean IsBold=font.isBold();  
boolean IsItalic=font.isItalic();
```

Matn rangi. Dasturingiz shrift usullari bilan ishlamasdan Graphics sinfining setColor funksiyasidan shrift rangini o'zgartirish uchun foydalanish mumkin. Rangni o'zgartirishning eng oson usuli ranglarning aniqlangan qiymatini ishlatish hisoblanadi:

<i>Color.black</i>	<i>Color.pink</i>	<i>Color.graf</i>	<i>Color.cyan</i>
<i>Color.darkGray</i>	<i>Color.yellow</i>	<i>Color.magenta</i>	<i>Color.gread</i>
<i>Color.lightGray</i>	<i>Color.blue</i>	<i>Color.red</i>	<i>Color.orange</i>

Misol uchun, quyidagi operator joriy rangni qizilga o'zgartiradi:


```
g.getColor(Color.red);
```

FontColor.java Appletini *SetColor* funksiyasidan foydalanib matnni bir necha rangda chiqarishi ko'rsatilgan:

```
import java.awt.*;  
import java.applet.*;  
public class FontColors extends Applet  
{  
    public void paint(Graphics g)  
    {  
        Font font=new Font("TimesRoman", Font.BOLD, 18);  
        g.setFont(font);  
        g.setColor(Color.red);  
        g.drawString("Red Red Red", 5, 30);  
        g.setColor(Color.blue);  
        g.drawString("Blue Blue Blue", 5, 30);  
        g.setColor(Color.green);  
        g.drawString("Green Green Green", 5, 30);  
        g.setColor(Color.yellow);  
        g.drawString("Yellow Yellow Yellow", 5, 30);  
    }  
}
```

Appletni kompilyatsiya qilib, ishga tushirsangiz uning oynasida quyidagi kabi xabar chiqadi:



1.7-rasm. Shriflarni turli ranglar bilan oynaga chiqarish.

1.4. Swing oynasini hosil qilish

Java tilidan foydalanib, foydalanuvchining grafik interfeysini hosil qilish mumkin. Bu ish java tilining *swing* paketidan foydalanib amalga oshiriladi. *javax.swing* paketida joylashgan sinflardan foydalanib, operatsion tizim oynalariga o'xshash bir qator oynalar hosil qilinadi. Bu paketni dasturga chaqirish uchun dastuning boshlang'ich qismida

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

satrini yozish kerak. Foydalanuvchi grafik interfeysini qurish uchun interfeysda foydalaniladigan komponentalarni o'zida saqlovchi sinflar qurish talab etiladi. Bu ishni bajarish *JFrame* sinfining sinf ostisi bo'lgan sinflarni *extends* kalit so'zidan foydalanib hosil qilish kerak. Bu foydalanuvchiga oynani siljitish, o'lchamini o'zgartirish kabi imkoniyatlarni beradi.

Oynalar bilan ishlashda hosil qilinayotgan sinf konstruktorida quyidagi keltirilgan minimal talablar uchun mo'ljallangan operatorlar bo'lishi talab etiladi:

- Oyna sarlavhasi – *JFrame* sinfidan meros bo'yicha olingan *super()* metodining argumenti sifatida yoziladi.
- Oyna o'lchami – bu xususiyat *setsize* metodining argumenti sifatida oynaning eni va balandligiga mos piksellar sonini bildiruvchi natural son ko'rinishidagi qiymatlar orqali beriladi.

- Foydalanuvchi tomonidan oynani yopish harakati – bu `setDefaultCloseOperation()` metodining argumenti sifatida beriladigan o‘zgarmas bilan aniqlanadi.
- Oynani aks ettirish parametri – bu grafik metod hisoblangan `setVisible()` metodining argumentida aniqlanadigan xususiyat.

Bularga qo‘shimcha qilib sinf konstruktoriga oyna komponentasini ham qo‘shish mumkin. Bu ish `JPanel` deb nomlangan konstruktordan foydalanib amalga oshiriladi, boshqa kichik hisoblangan komponentalarni qo‘shish uchun `JFrame` sinfining `add()` metodidan foydalanish kerak.

Quyida keltiriladigan dasturda `JFrame` konteyneri (paneli) `FlowLayout` qolip menejeri bilan `JPanel` konteyneridan foydalanib bazaviy oynani hosil qilish amalga oshirilgan. Bu dastur yordamida tuzilgan oynadan, keyingi oyna bilan bog‘liq ravishda ishlaydigan dasturlarda ham foydalanamiz:

```
import javax.swing.*;

public class Oyna extends JFrame {

JPanel pnl = new JPanel();

public Oyna() {

    super("Swing oynasi!");

    setSize(500, 200);
    setDefaultCloseOperation(EXIT_ON_CLOSE);

add(pnl);

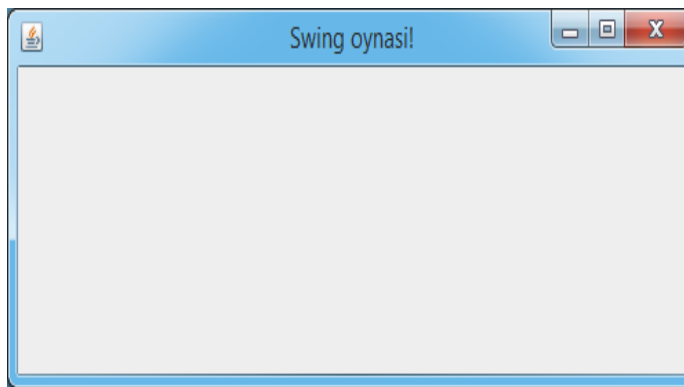
setVisible(true); }

    public static void main(String[] args){

Oyna gui = new Oyna();}

}
```

Dastur natijasi:



1.8.- rasm. Swing paketidan foydalanib yaratilgan oyna ko'rinishi

1.5. Tugmalar qo'shish

Grafik interfeysda tugmalar o'rnatish uchun *JButton* sinfidan foydalaniladi. Bu tugmalarning vazifasi dastur ishlashi davomida o'ziga biriktirilgan operatorlarni ishga tushirishdan iborat.

JButton turdagi obyekt *new* kalit so'zi bilan hosil qilinadi. Uning argumentida keltirilgan satr turidagi qiymat esa tugma ustiga chiqadigan yozuv bo'lib xizmat qiladi.

Tugma ustiga rasmlarni ham joylashtirish mumkin. Buning uchun avval *ImageIcon* turidagi obyektни hosil qilish kerak. Bu obyektning konstruktori argument sifatida rasm faylining nomini qabul qiladi. Odatda foydalanilayotgan rasm dastur matni bilan bitta katalogda bo'lishi kerak. Agar rasm faylida boshqa katalogda joylashgan bo'lsa, shu katalogga yo'l, rasm fayli bilan birga argumentda ko'rsatilishi kerak. Agar tugma ustida yozuv va rasmni birgalikda joylashtirish talab etilsa, *JButton* obyektining argumentida yozuv va *ImageIcon* obyektini ko'rsatiladi.

Quyidagi dasturda oynaga tugmalar joylashtirish amalga oshirilgan:

```
import javax.swing.*;  
public class Tugma extends JFrame {  
    JPanel pnl = new JPanel();  
    ImageIcon rasml = new ImageIcon("rasml.png");
```

```

ImageIcon rasm2 = new ImageIcon("rasm2.png");
JButton btn = new JButton("Oddiy tugma!");
JButton tickBtn = new JButton(rasm1);
JButton crossBtn = new JButton(
    "Keyingi oynaga", rasm2);
public Tugma() {
    super("Tugmalar joylashtirilgan oyna!");
    setSize(500, 200);
    setDefaultCloseOperation(EXIT_ON_CLOSE);
    pnl.add(btn);
    pnl.add(tickBtn);
    pnl.add(crossBtn);
    add(pnl);
    setVisible(true);
}
public static void main(String[] args) {
    Tugma gui = new Tugma();
}
}

```

Dastur natijasi:



1.9.- rasm. Tugmalar joylashtirilgan oyna

Hosil bo'lgan oynada ko'rib turibsizki, oddiy tugma va ikonkali tugmalar oynada o'z aksini topgan.

II BOB. JAVADA GRAFIKA

2.1. Grafikani qo'llab quvvatlash

Javaning AWT paketi tarkibida grafikani qo'llab quvvatlovchi bir qator sinflar mavjud. Bunday sinflar *Swing* paketi yordamida hosil qilingan oynalarda turli grafiklar chizish imkonini beradi. Bunday oynalar applet oynasi yoki dastur ilovasi oynasi ko'rinishida bo'lishi mumkin. Har bir oynaning koordinatasi quyidagicha bo'ladi: Oynaning yuqori chap burchagida (0;0) nuqta joylashadi. *X* o'qi shu nuqtadan boshlab oynaning yuqori o'ng burchagi tomon yo'naltirilgan bo'ladi. *Y* o'qi esa oynaning quyi chap burchagi tomon yo'naltiriladi. Oynaning o'lchami piksellarda aniqlanadi.

Bundan tashqari `java.awt` paketi interfeys oynasining ko'rinishini o'zgartirishga mo'ljallangan bir qator sinflarni o'zida saqlaydi. Bu paketni dasturga chaqirish uchun dasturning boshlang'ich qismida

```
import.java.awt.*;
```

satrini yozish kerak bo'ladi.

Oynada joylashtiriladigan har bir element yuqorida keltirilgan parametrlarga asoslanadi.

Grafik kontekstlar *Graphics* sinfida jamlangan. Ular quyida keltirilgan 2 ta usul yordamida olinadi:

1. Metod argumenti sifatida uzatish. Masalan *paint()* yoki *update()* metodlarning argumenti sifatida;
2. *Component* sinfidan *GetGraphics()* metodi yordamida olish.

Bulardan tashqari *Graphics* sinfida bir qator grafiklarni chizishga mo'ljallangan metodlar ham joylashgan. Ular yordamida chiziq, to'rtburchak, ellips kabi shakllarni oynada hosil qilish mumkin. Grafiklar chizishda ba'zi metodlar shakl chegarasini tasvirlashni amalga oshirsa, ayrimlari hosil qilingan shakllarning o'zi yoki ichini bo'yashni bajaradi.

Hosil qilingan grafiklar odatda qora rangda bo‘ladi. Agar chizilayotgan grafik obyekt mavjud oyna chegarasidan chiqib ketsa, u avtomatik ravishda to‘g‘rilanadi.

2.2. Grafik primitivlarni hosil qilish

Bu bo‘limda *Graphics* sinfining bir qator metodlari va ularning tavsifini keltiramiz.

Chiziq chizish. Ekranda chiziq chizish uchun *DrawLine()* metodidan foydalaniladi, uning umumiy ko‘rinishi quyida keltirilgan:

```
void DrawLine (int x1, int y1, int x2, int y2)
```

Bu metod *x1* va *y1* nuqtadan boshlanib, *x2* va *y2* nuqtada tugaydigan kesmani chizadi.

To‘g‘ri to‘rtburchak chizish. *drawRect()* va *fillRect()* metodlari mos ravishda to‘rtburchak va ichi bo‘yalgan to‘rtburchaklar chizishni bajaradi. Ularning umumiy ko‘rinishi quyida keltirilgan

```
void drawRect(int x1, int y1, int x2, int y2)
```

```
void fillRect(int x1, int y1, int x2, int y2)
```

To‘rtburchakning yuqori chap burchak *x1*, *y1* koordinatada joylashadi. Uning eni va balandligi *x2* va *y2* o‘zgaruvchilardagi berilgan qiymatlar bilan aniqlanadi.

Uchlari yoysimon bo‘lgan to‘g‘ri to‘rtburchak chizish uchun *RoundRect()* va *fillRoundRect()* metodlaridan foydalaniladi. Bu metodlarning umumiy ko‘rinishi quyidagicha:

```
void drawRoundRect (int x1, int y1, int x2, int y2, int dx, int dy)
```

```
void fillRoundRect (int x1, int y1, int x2, int y2, int dx, int dy)
```

Bu metodlar yordamida to‘g‘ri to‘rtburchaklarning uchlari yoysimon ko‘rinishga ega bo‘ladi. Uchlaridagi yoylarning *x* va *y* o‘qlari bo‘yicha diametrlari mos ravishda *dx* va *dy* o‘zgaruvchilarning qiymatlari bilan aniqlanadi.

Ellips va aylana chizish. *drawOval()* va *fillOval()* metodlari mos ravishda ichi bo‘yalmagan va bo‘yalgan ellips hamda aylanalar chizishni bajaradi. Ularning umumiy ko‘rinishi quyidagicha:

```
void drawOval (int x1, int y1, int x2, int y2)
```

void fillOval (int x1, int y1, int x2, int y2)

Ellips o'zini chegaralovchi (unga tashqi chizilgan) to'g'ri to'rtburchakli sohada chiziladi. Shuning uchun ham uning argumentidagi parametrlar quyidagicha: $(x1,y1)$ ellipsga tashqi chizilgan to'g'ri to'rtburchakning yuqori chap uchining koordinatasi, $(x2,y2)$ lar esa shu to'rtburchakning eni hamda balandligining (piksellarda) son qiymati. Aylana chizish uchun esa chegaralovchi to'g'ri to'rtburchakning eni va bo'yining qiymatlarini o'zaro teng ko'rinishda berish, ya'ni to'rtburchakni kvadrat shakliga keltirish kerak bo'ladi.

Yoy chizish. *drawArc()* va *fillArc()* metodlari yordamida aylana yoyini chizish mumkin. Ularning umumiy ko'rinishi quyidagicha:

void drawArc(int x1, int y1, int x2, int y2, int boshlang'ich_burchak, int burilish_burchagi)

void fillArc(int x1, int y1, int x2, int y2, int boshlang'ich_burchak, int burilish_burchagi)

Aylana yoyi to'g'ri to'rtburchakli soha bilan chegaralanadi. Shuning uchun ham $(x1,y1)$ parametrlar chegaralovchi to'g'ri to'rtburchakning yuqori chap uchining koordinatasi, $x2$ va $y2$ lar esa chegaralovchi to'g'ri to'rtburchakning eni va balandligining (piksellardagi) o'lchami hisoblanadi.

Yoy <boshlang'ich_burchak> va <burilish_burchagi> ning qiymatlarga (burchaklar gradus o'lchovida ko'rsatiladi) asosan yuqorida ta'kidlangan to'g'ri to'rtburchakli sohada chiziladi. 0^0 soat strelkasining 3 ni ko'rsatib turgan holatiga mos ko'rinishda olinadi va burchak yoyining oshishi soat strelkasi yo'nalishiga teskari yo'nalishda bo'ladi. Agar burilish_burchagi qiymati manfiy bo'lsa burilish_burchagi soat strelkasi yo'nalishi bo'yicha bo'ladi.

Masalan soat strelkasining 12 sonidan 6 sonigacha bo'lgan oraliqda soat strelkasi yo'nalishidagi yoyni chizish uchun <boshlang'ich_burchak> parametriga 90^0 ni , <burilish_burchagi>ga esa -180^0 ni qiymatlash kerak bo'ladi.

Ko'pburchak chizish. *drawPolygon()* va *fillPolygon()* metodlari yordamida ichi bo'yalmagan va bo'yalgan ko'p burchaklar chizish mumkin. Ularning umumiy ko'rinishi quyidagicha:


```
void drawPolygon(int x[], int y[], int)
```

```
void fillPolygon(int x[], int y[], int n)
```

Ko'pburchakning oxirgi uchining koordinatasi x va y massivlarining juftligi bilan aniqlanadi. x va y massivi bilan berilgan uchlarining koordinatasi nuqtalar sonini aniqlaydi.

Ko'pburchakni chizishning muqobil usuli *Polygon* sinfi turidagi obyektidan foydalanish hisoblanadi.

Quyida keltirilgan dasturda yuqorida ko'rib o'tilgan metodlardan foydalanib, turli shakllar chizish namoyish etilgan:

```
//Grafik primitivlarni chizish
import java.applet.Applet;
import java.awt.*;
import java.applet.*;
public class Grafik_primitivlar
            extends Applet {
    public void paint(Graphics g) {
        //Kesmalar chizish
        g.drawLine(0, 0, 100, 90);
        g.drawLine(0, 90, 100, 10);
        g.drawLine(40, 25, 250, 80);
        //To'g'ri to'rtburchaklar chizish
        g.drawRect(10, 150, 60, 50);
        g.fillRect(100, 150, 60, 50);
        g.drawRoundRect(190,150,60,50,15,15);
        g.fillRoundRect(280,150,60,50,30,40);
        //Ellips va aylanalarni chizish
        g.drawOval(10, 250, 50, 50);
```

```

        g.fillOval(90, 250, 75, 50);
        g.drawOval(190, 260, 100, 40);
        //Yoylar chizish
        g.drawArc(10, 350, 70, 70, 0, 180);
        g.fillArc(60, 350, 70, 70, 0, 75);
        //Ko'pburchak chizish
        int xpoints[]={10,200,10,200,10};
        int ypoints[]={450,450,650,650,450};
        int num=5;
        g.drawPolygon(xpoints,ypoints,num);
    }
}

```

2.3. Grafiklarning o'lchamlarini o'zgartirish

Ko'p hollarda chizilayotgan chizmani oyna o'lchamlari bilan moslashtirishga to'g'ri keladi. Masalan, *getSize()* metodi yordamida joriy oyna o'lchamlarini aniqlash mumkin. Bu metod *Dimension* sinfi turidagi obyekt bo'lib, oyna o'lchamlarini olish va bu o'lchamlardan foydalanib chiziladigan grafiklarni masshtablashtirish jarayonini boshqarishda yordam beradi.

Ekkranda chizilgan shaklning o'lchamini sichqoncha tugmasini bosish yordamida o'zgartirishni namoyish etuvchi dastur matnini keltiramiz:

```

// Chizma o'lchamini o'zgartirish
import java.applet.Applet;
import java.awt.*;
import java.awt.event.*;
public class Ulchamni_uzgartirish extends Applet
{ final int inc=25;
  int max=500;
  int min=200;
  Dimension d;
  public Ulchamni_uzgartirish(){

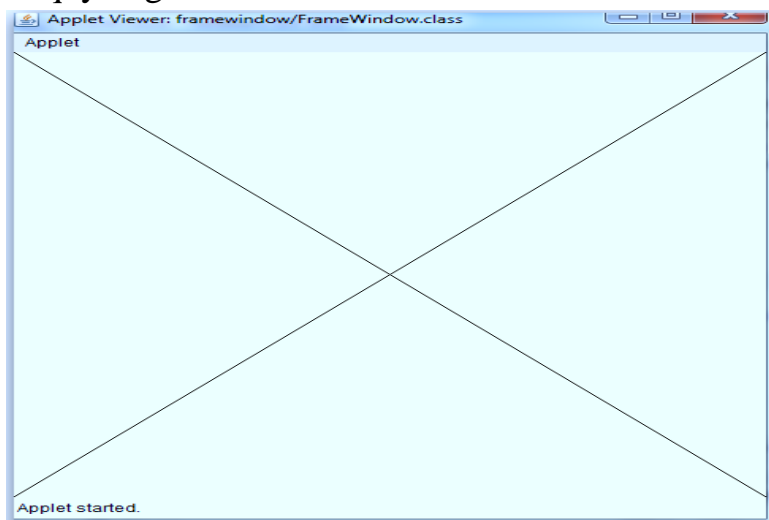
```

```

addMouseListener(new MouseAdapter() {
    public void mouseReleased(MouseEvent me) {
        int w =
            (d.width+inc)>max?min:(d.width+inc);
        int h =
            (d.height+inc) > max?min:(d.height+inc);
        setSize(new Dimension(w,h));    }
    }; }
public void paint(Graphics g) {
    d=getSize();
    g.drawLine(0,0,d.width-1,d.height-1);
    g.drawLine(0,d.height-1,d.width-1, 0);
    g.drawLine(0,0,d.width-1, d.height-1);
}
}

```

Dastur natijasi quyidagicha:



2.1-rasm.

Applet oynasi ustida sichqoncha tugmasi bilan ta'sir qilinganda oyna o'lchami kattalashadi.

2.4. Ranglar bilan ishlash

Java tilining AWT paketi dastlab ma'lum bir rangni taklif etadi. So'ngra bu rang joriy apparat parametrlariga moslab o'zgartirishi mumkin. Ranglar *Color* sinfida joylashgan o'zgarmaslar yordamida aniqlanadi. Kerakli rangdan foydalanish uchun *Color* sinfi turidagi obyekt konstruktoriga tegishli rang o'zgarasini kiritish kerak.

Quyida *Color* sinfi konstruktorining ko‘p foydalaniladigan 3 xil ko‘rinishiga misol keltirilgan:

```
Color(int qizil, int yashil, int ko‘k)
```

```
Color(int RGB_dagi_rang_o‘zgarishining_qiymati)
```

```
Color(float qizil, float yashil, float ko‘k)
```

Sinfning birinchi o‘rinda keltirilgan konstruktori 3 ta butun turdagi argument qabul qiladi. Beriladigan ranglar esa qizil, yashil, ko‘k ranglari yordamida aniqlanadi. Bu konstruktor argumentida ko‘rsatilgan qiymatlar 0-255 oraliqda bo‘lishi mumkin, ya’ni quyidagicha:

```
newColor(255,100,100) // yorqin qizil rang
```

Sinfning 2-o‘rinda yozilgan konstruktori argumentida butun son ko‘rinishidagi bitta qiymat yoziladi. Bu qiymat RGB uyg‘unlashtirilgan rang qiymati hisoblanadi.

Bu butun sonli qiymat quyidagicha tashkil etilgan:

- Qizil rang tashkil etuvchisi uchun 16 bitdan 23 bitgacha ;
- Yashil rang tashkil etuvchisi uchun 8 bitdan 15 bitgacha ;
- Sariq rang tashkil etuvchisi uchun 0 bitdan 7 bitgacha;

Quyida bu konstruktorning qabul qiladigan qiymatiga misol keltirilgan:

```
int newRed = (0xFF000000) | (0xC0<<16) | (0X00<<8) | (0X00);
```

```
Color darkRed = new Color(newRed);
```

Color sinfining oxirgi o‘rinda joylashgan konstruktori 3 ta haqiqiy turga tegishli argumentga ega bo‘lib, ular 0,0 dan 1,0 gacha oraliqdagi haqiqiy son qiymatlarini qabul qiladi. Bu qiymatlar qizil, yashil va ko‘k ranglarni tasvirlaydi. *setForeground()* va *setBackground()* metodlaridan foydalanib, grafiklarning old va orqa fonlarni rangini berish mumkin.

Chizmaga ko‘rsatilgan rangni berish. Odatda grafik obyektning old foni ayni vaqtda joriy bo‘lgan rangda chiziladi. Chizmalarning rangini o‘zgartirish uchun *Graphics* sinfining *setColor()* metodidan foydalaniladi. Uning umumiy ko‘rinishi quyidagicha:

```
void setColor(Color yangi_rang)
```

Bu yerda yangi_rang parametri chizilayotgan obyektga beriladigan rangni bildiradi.

getColor() metodini chaqirib oldin o'rnatilgan rangni qayta o'rnatish mumkin.

Bu ish quyidagicha:

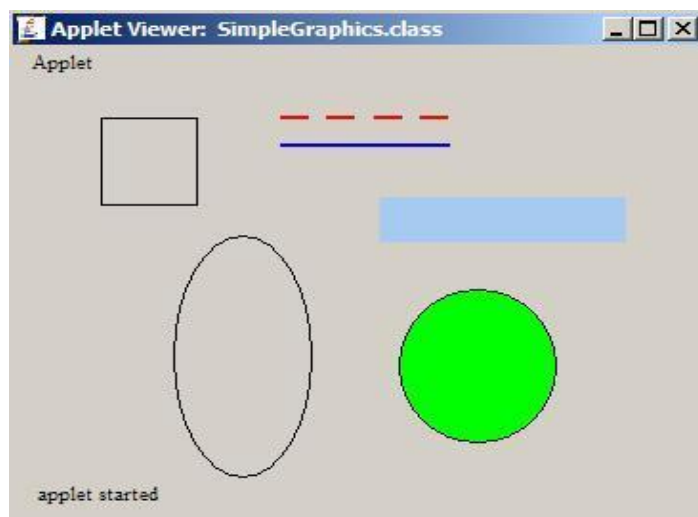
Color getColor()

Quyidagi applet (*SimpleGraphics.java*) oddiy tasvirlarni xosil qilish uchun grafikli funksiyalardan foydalanadi:

```
import java.awt.*;
import java.applet.*;

public class SimpleGraphics extends Applet
{
    public void paint (Graphics g) {
        g.drawRect (25,25,50,50);
        g.drawOval (75,100,100,150);
        g.fillRect (250,25,100,25);
        g.fillOval (205,125,100,100);
    }
}
```

Kompilyatsiya qilib, appletni ishga tushirsangiz, uning oynasida quyidagi rasm hosil bo'ladi:



2.2- rasm. Graphics sinfidan foydalanib bir nechta oddiy tasvirlarni hosil qilish.

Tasvirlarga rang berishda yuqoridagi usullardan tashqari ranglar palitrasini to'g'ridan to'g'ri ekranga chiqarib, shu yerdan ham tanlash imkoniyati bor. Rang tanlashga mo'ljallangan quyidagi dastur kodini keltiramiz:

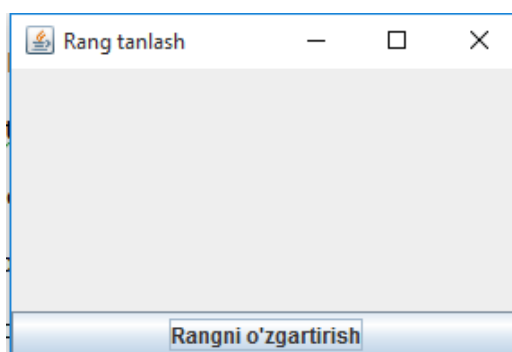
```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
@SuppressWarnings("serial")
public class Rang extends JFrame {
    JPanel panel;
    Color bgColor = Color.LIGHT_GRAY;
    public Rang() {
        panel = new JPanel(new BorderLayout());
        JButton btnColor = new JButton("Rangni
o'zgartirish");
        panel.add(btnColor, BorderLayout.SOUTH);
        btnColor.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent evt) {
                Color color =
JColorChooser.showDialog(Rang.this,
```

```

        "Rang tanlash oynasi", bgColor);
    if (color != null) {
        bgColor = color;
    }
    panel.setBackground(bgColor);
}
});
setContentPane(panel);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setTitle("Rang tanlash");
setSize(300, 200);
setLocationRelativeTo(null);
setVisible(true);
}
public static void main(String[] args) {
    SwingUtilities.invokeLater(new Runnable() {
        @Override
        public void run() {
            new Rang();
        }
    });
}
}
}

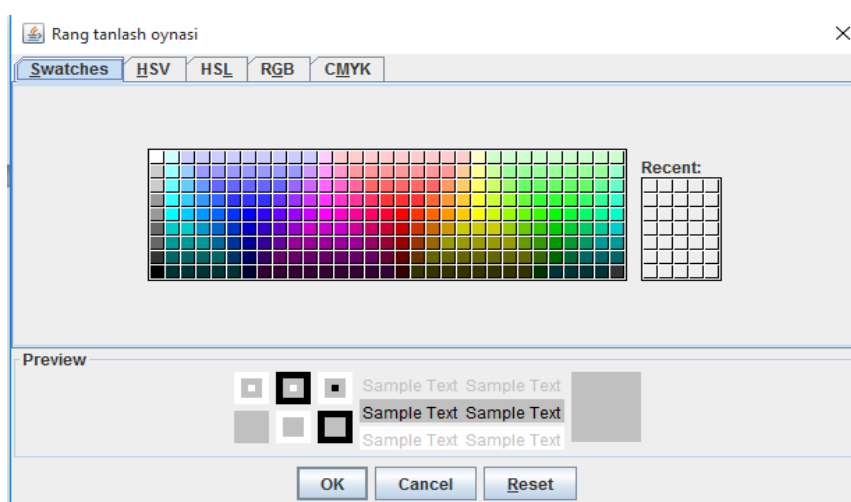
```

Dasturni ishlatish natijasida quyidagi oyna hosil bo'ladi:



2.3-rasm. Rangni o'zgartirishga mo'ljallangan dastur oynasi

Bu yerda “Rangni o'zgartirish” tugmasini bosish yordamida rang tanlash oynasini hosil qilamiz:



2.4-rasm. Rang tanlash oynasi

Bu oynadan istalgan rangni tanlab OK tugmasi bosilsa, dastlabki oyna belgilangan rangga bo'yaladi. Masalan qizil rang tanlanib OK tugmasi bosilsa, oyna quyidagi ko'rinishga ega bo'ladi:



2.5-rasm. Rang o'zgartirilgandan keying holat

III BOB. Mukammal dasturlar yaratishda Java tilining grafik va matnlar bilan ishlash imkoniyatlaridan foydalanish

3.1. Shakl harakatini boshqarishda Java tilining grafik imkoniyatlaridan foydalanish

Java ning asosiy qulayliklardan biri grafikani va animatsiyalarni qo'llashidir. Grafiklar chizishda ba'zi metodlar shakl chegarasini tasvirlashni amalga oshirsa, ayrimlari hosil qilingan shakllarning o'zi yoki ichini bo'yashni bajaradi. Bunday metodlar sirasiga *drawLine()*, *drawRect()*, *fillRect()*, *drawOval()*, *fillOval()*, *drawPolygon()*, *fillPolygon()* kabilarni keltirish mumkin.

Java tilining AWT paketi dastlab ma'lum bir rangni taklif etadi. So'ngra bu rang joriy apparat parametrlariga moslab o'zgartirishi mumkin. Ranglar *Color* sinfida joylashgan o'zgarmlar yordamida aniqlanadi. Kerakli rangdan foydalanish uchun *Color* sinfi turidagi obyekt konstruktoriga tegishli rang o'zgarmlarini kiritish kerak.

Yuqoridagi bobda keltirilgan metodlardan foydalangan holda applet oynasida qizil rangdagi ustunni chizib, uni mos ravishda o'ngga va chapga siljitishni amalga oshiruvchi dastur kodini keltiramiz:

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
public class Grafik extends JFrame {
    public static final int CANVAS_WIDTH = 400;
    public static final int CANVAS_HEIGHT = 140;
    public static final Color CANVAS_BG_COLOR = Color.CYAN;
    private DrawCanvas canvas;
    private Sprite sprite;
    public Grafik() {
        sprite = new Sprite(CANVAS_WIDTH / 2 - 5, CANVAS_HEIGHT
        / 2 - 40, 10, 80, Color.RED);
        JPanel btnPanel = new JPanel(new FlowLayout());
        JButton btnLeft = new JButton("Chapga surish ");
        btnPanel.add(btnLeft);
```

```

btnLeft.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent evt) {
        moveLeft();
        requestFocus(); }
});
JButton btnRight = new JButton("O'ngga surish");
btnPanel.add(btnRight);
btnRight.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent evt) {
        moveRight();
        requestFocus(); }
});
canvas = new DrawCanvas();
canvas.setPreferredSize(new Dimension(CANVAS_WIDTH,
CANVAS_HEIGHT));
Container cp = getContentPane();
cp.setLayout(new BorderLayout());
cp.add(canvas, BorderLayout.CENTER);
cp.add(btnPanel, BorderLayout.SOUTH);
addKeyListener(new KeyAdapter() {
    @Override
    public void keyPressed(KeyEvent evt) {
        switch(evt.getKeyCode()) {
            case KeyEvent.VK_LEFT: moveLeft(); break;
            case KeyEvent.VK_RIGHT: moveRight(); break;
        } }
});
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setTitle("Ustunni harakatlantirish");
pack();
setVisible(true);
requestFocus(); }
private void moveLeft() {
    int savedX = sprite.x;
    sprite.x -= 10;
    canvas.repaint(savedX, sprite.y, sprite.width,
sprite.height);
    canvas.repaint(sprite.x, sprite.y, sprite.width,
sprite.height);
}

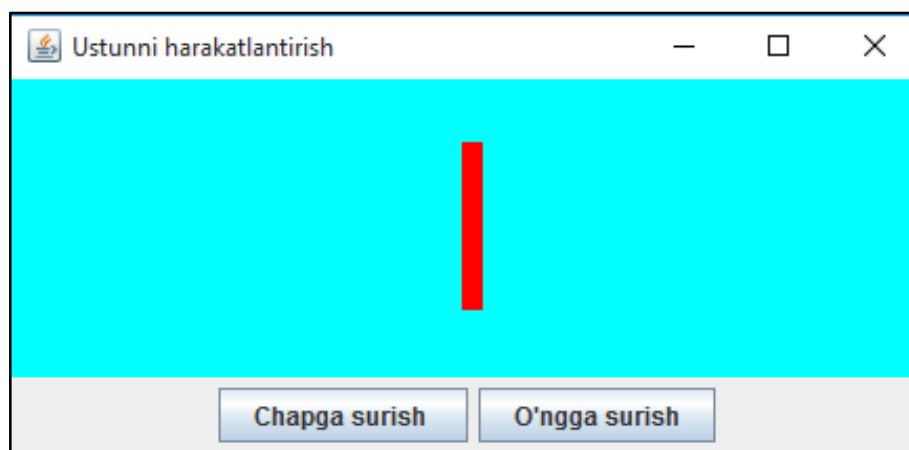
```

```

private void moveRight() {
    int savedX = sprite.x;
    sprite.x += 10;
    canvas.repaint(savedX,      sprite.y,      sprite.width,
sprite.height);
    canvas.repaint(sprite.x,    sprite.y,    sprite.width,
sprite.height);  }
class DrawCanvas extends JPanel {
    @Override
    public void paintComponent(Graphics g) {
        super.paintComponent(g);
        setBackground(CANVAS_BG_COLOR);
        sprite.paint(g); // the sprite paints itself
    }
public static void main(String[] args) {
    SwingUtilities.invokeLater(new Runnable() {
        @Override
        public void run() {
            new Grafik();  } });
}
public class Sprite {
    int x, y, width, height;
    Color color = Color.RED;
    public Sprite(int x, int y, int width, int height, Color
color) {
        this.x = x;
        this.y = y;
        this.width = width;
        this.height = height;
        this.color = color;  }
    public void paint(Graphics g) {
        g.setColor(color);
        g.fillRect(x, y, width, height);
    }
}
}
}

```

Ushbu dastur ishlashi natijasida quyidagi natijaga erishildi:



3.1-rasm. Shakl harakatini boshqaruvchi dastur oynasi

Yuqorida keltirilgan dasturdan foydalanib, Java tilining grafik imkoniyatlaridan foydalangan holda jism harakatini boshqarishga mo'ljallangan animatsion holatlarni bajarish mumkin. Bunday dasturiy vositalardan turli fanlarni o'qitishda ko'rgazmali materiallar tayyorlash uchun foydalanish mumkin bo'ladi.

3.2. Javada animatsion dasturlar yaratish

Animatsiya bu – biror obyektning ekrandagi harakati hisoblanadi.

Quyida keltirilgan kodlarda “to'p” (koptok) ning ko'rsatilgan sohadagi tartibsiz harakatini boshqarishni aks ettiradigan dastur keltirilgan.

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
@SuppressWarnings("serial")
public class Animatsiya extends JFrame {
    private static final int CANVAS_WIDTH = 640;
    private static final int CANVAS_HEIGHT = 480;
    private static final int UPDATE_PERIOD = 50;
    private DrawCanvas canvas;
    private int x = 100, y = 100;
    private int size = 250;
    private int xSpeed = 3, ySpeed = 5;
```

```

public Animatsiya() {
    canvas = new DrawCanvas();

    canvas.setPreferredSize(new Dimension(CANVAS_WIDTH,
CANVAS_HEIGHT));

    this.setContentPane(canvas);

    this.setDefaultCloseOperation(EXIT_ON_CLOSE);

    this.pack();

    this.setTitle("Bouncing Ball");

    this.setVisible(true);

    ActionListener updateTask = new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent evt) {
            update();
            repaint();
        }
    };

    new Timer(UPDATE_PERIOD, updateTask).start();
}

public void update() {
    x += xSpeed;
    y += ySpeed;

    if (x > CANVAS_WIDTH - size || x < 0) {
        xSpeed = -xSpeed;
    }

    if (y > CANVAS_HEIGHT - size || y < 0) {
        ySpeed = -ySpeed;
    }
}

private class DrawCanvas extends JPanel {
    @Override

```

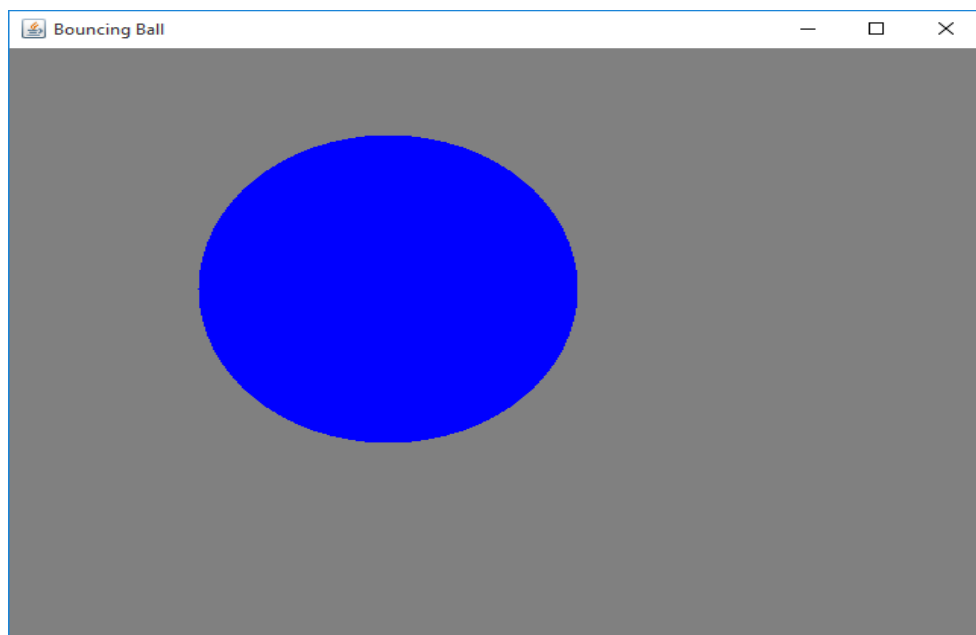
```

public void paintComponent(Graphics g) {
    super.paintComponent(g);
    setBackground(Color.GRAY);
    g.setColor(Color.BLUE);
    g.fillOval(x, y, size, size);
}
}

public static void main(String[] args) {
    SwingUtilities.invokeLater(new Runnable() {
        @Override
        public void run() {
            new Animatsiya();
        }
    });
}
}

```

Ushbu dastur ishlashi natijasida quyidagi oyna hosil bo'ladi. Undagi harakatlayotgan to'p chegaralangan soha ichidan chiqib ketmaydi.



3.2-rasm. Harakatlanuvchi koptok

To'p obyektini boshqarish soha chegaralarida shartni tekshirish yo'li bilan amalga oshirilgan. Yuqoridagi dasturda dastlab harakatlanuvchi obyekt hosil qilingan, so'ngra esa uning harakatlantirish amalga oshirilgan.

Dasturlashda faqat shakllarni emas, balki tayyor rasmlarni ham harakatga keltirish mumkin. Buning uchun harakatlanishi lozim bo'lgan rasmlar dastur saqlanayotgan katalogga olib o'tiladi. Quyida keltirilgan dastur kodlarida oldindan mavjud bo'lgan tasvirlarni harakatlantirish amalga oshiriladi:

```
import java.awt.geom.AffineTransform;
import javax.imageio.ImageIO;
import java.net.URL;
import java.awt.*;
import javax.swing.*;
import java.io.*;
@SuppressWarnings("serial")
public class Animatsiyal extends JPanel {
    static final int CANVAS_WIDTH = 640;
    static final int CANVAS_HEIGHT = 480;
    public static final String TITLE = "Harakatlanuvchi
packman";

    private String[] imgFileNames = {
        "images/pacman_1.png", "images/pacman_2.png",
"images/pacman_3.png"};

    private Image[] imgFrames;
    private int currentFrame = 0;
    private int frameRate = 5;
    private int imgWidth, imgHeight;
    private double x = 100.0, y = 80.0;
    private double speed = 8;
    private double direction = 0;
```

```

private double rotationSpeed = 1;

private AffineTransform transform = new
AffineTransform();

public Animatsiyal() {
    loadImages(imgFileNames);
    Thread animationThread = new Thread () {
        @Override
        public void run() {
            while (true) {
                update();
                repaint();
                try {
                    Thread.sleep(1000 / frameRate);
                } catch (InterruptedException ex) { }
            }
        }
    };
    animationThread.start();

    setPreferredSize(new Dimension(CANVAS_WIDTH,
CANVAS_HEIGHT));
}

private void loadImages(String[] imgFileNames) {
    int numFrames = imgFileNames.length;
    imgFrames = new Image[numFrames];
    URL imgUrl = null;
    for (int i = 0; i < numFrames; i++) {
        imgUrl =
getClass().getClassLoader().getResource(imgFileNames[i]);
        if (imgUrl == null) {
            System.err.println("Couldn't find file: " +
imgFileNames[i]);

```



```

    } else {
        try {
            imgFrames[i] = ImageIO.read(imgUrl);
        } catch (IOException ex) {
            ex.printStackTrace();
        }
    }
}

imgWidth = imgFrames[0].getWidth(null);
imgHeight = imgFrames[0].getHeight(null);
}

public void update() {
    x += speed * Math.cos(Math.toRadians(direction));
    if (x >= CANVAS_WIDTH) {
        x -= CANVAS_WIDTH;
    } else if (x < 0) {
        x += CANVAS_WIDTH;
    }

    y += speed * Math.sin(Math.toRadians(direction));
    if (y >= CANVAS_HEIGHT) {
        y -= CANVAS_HEIGHT;
    } else if (y < 0) {
        y += CANVAS_HEIGHT;
    }

    direction += rotationSpeed;
    if (direction >= 360) {
        direction -= 360;
    } else if (direction < 0) {
        direction += 360;
    }
}

```

```

    }
    ++currentFrame;
    if (currentFrame >= imgFrames.length) {
        currentFrame = 0;
    }
}

@Override
public void paintComponent(Graphics g) {
    super.paintComponent(g);
    setBackground(Color.WHITE);
    Graphics2D g2d = (Graphics2D) g;
    transform.setToIdentity();
    transform.translate(x - imgWidth / 2, y - imgHeight /
2);
    transform.rotate(Math.toRadians(direction),
        imgWidth / 2, imgHeight / 2);
    g2d.drawImage(imgFrames[currentFrame], transform,
null);
}

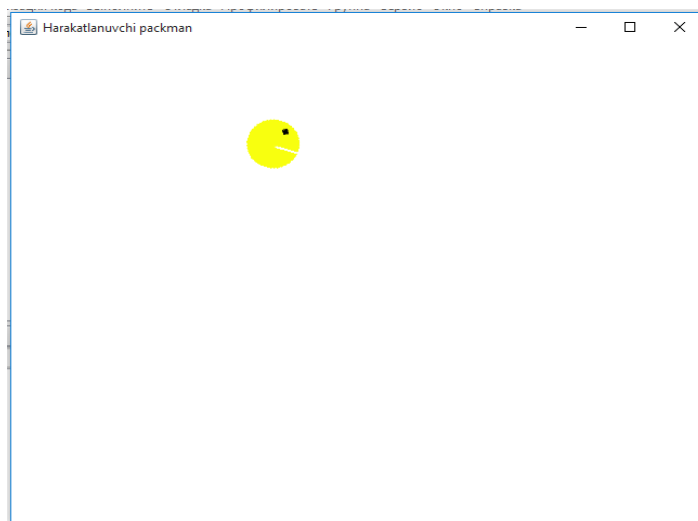
public static void main(String[] args) {
    SwingUtilities.invokeLater(new Runnable() {
        @Override
        public void run() {
            JFrame frame = new JFrame(TITLE);
            frame.setContentPane(new Animatsiyal());
            frame.pack();

frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
            frame.setLocationRelativeTo(null);
            frame.setVisible(true);
        }
    }
}

```

```
    });  
  }  
}
```

Bu dastur bajarilishi natijasida quyidagicha harakatlanuvchi oyna hosil bo'ladi:



3.3-rasm. Harakatlanuvchi “packman”

Yuqoridagi dastur ishlashi jarayonida uchta obyektning belgilangan sohadagi harakati amalga oshadi.

3.3. Sodda matn muharriri dasturini yaratish

Java tilining imkoniyatlaridan foydalanib turli qurilmalarda ishlay oladigan matn muharririni yaratish mumkin. Matn muharririni yaratishdan maqsad ularni turli qurilmalar uchun moslashtirilgan ko'rinishda tashkil etishdan iboratdir. Quyida keltirilgan dastur kodlarida sodda korinishdagi matn muharriri tasvirlangan:

```
import java.io.*;  
import java.awt.*;  
import java.awt.event.*;  
import java.util.*;  
class FrameWindow extends Frame implements ActionListener,  
WindowListener
```

```

{
    TextArea ta; // matn oynasi
    MenuBar mb; // menyu paneli

    Menu mFile; //File menyusi

    MenuItem miOpen;
    MenuItem miSave;
    MenuItem miSaveAs;
    MenuItem miExit;

    String PolniiPut = "";
    byte buf[];

    // konstruktorni ifodalaymiz
    public FrameWindow(String szTitle)
    {
        super(szTitle);
        setSize(800, 600);

        mb = new MenuBar();
        mFile = new Menu("File");

        miOpen = new MenuItem("Open...");
        mFile.add(miOpen);

        miSave = new MenuItem("Save");
        mFile.add(miSave);

        miSaveAs = new MenuItem("Save As...");
        mFile.add(miSaveAs);

        mFile.add("-");
    }
}

```

```

miExit = new MenuItem("Exit");
mFile.add(miExit);

mb.add(mFile);

// menu punktlari
miOpen.addActionListener(this);
miSave.addActionListener(this);
miSaveAs.addActionListener(this);
miExit.addActionListener(this);

setMenuBar(mb); //menu yaratish
this.addWindowListener(this);

ta = new TextArea(10, 30);
setLayout(new BorderLayout());
add("Center", ta);
} //FrameWindow

public void actionPerformed(ActionEvent e)
{
    if(e.getSource().equals(miOpen))
    {
        FileOpen();
    }

    else if(e.getSource().equals(miSave))
    {
        FileSave();
    }

    else if(e.getSource().equals(miSaveAs))

```

```

    {
        FileSaveAs ();
    }

    else if (e.getSource().equals(miExit))
    {
        setVisible(false);
        System.exit(0);
    }
} //actionPerformed

public void windowClosing(WindowEvent e)
{
    setVisible(false);
    System.exit(0);
}

public void windowOpened(WindowEvent e) {}
public void windowClosed(WindowEvent e) {}
public void windowIconified(WindowEvent e) {}
public void windowDeiconified(WindowEvent e) {}
public void windowActivated(WindowEvent e) {}
public void windowDeactivated(WindowEvent e) {}

void FileOpen()
{
    FileDialog      FD;
    FD = new FileDialog(this, "Open file",FileDialog.LOAD);
    FD.show();
    PolniiPut = FD.getDirectory()+FD.getFile();
    setTitle("Simplest Java Notepad" + " - " +PolniiPut);

    FileInputStream FIS = null

```

```

try
{
    FIS = new FileInputStream(PolniiPut);
    buf = new byte[FIS.available()];
    FIS.read(buf);
}
catch (IOException ex)
{
    System.out.println(ex.toString());
}
ta.selectAll();
ta.replaceRange("", 0, ta.getSelectionEnd());

String szStr = new String(buf);

StringTokenizer st;
st = new StringTokenizer(szStr, "\r\n");

while(st.hasMoreElements())
{
    szStr = new String((String)st.nextElement());
    ta.append(szStr + "\r\n");
} //end_while

try
{
    FIS.close(); // FileInputStream yopildi
}
catch (IOException ex)
{
    System.out.println(ex.toString());
}

```

```

} //FileOpen

void FileSaveAs ()
{
    FileDialog FD;
    FD = new FileDialog(this, "Save file as...",
FileDialog.SAVE);
    FD.show();

    if(FD.getDirectory() == null || FD.getFile() == null)
    {
        System.out.println("No File Selected!!!");
    }
    else
    {
        PolniiPut = FD.getDirectory()+FD.getFile();
        setTitle("Simplest Java Notepad" + " - "
+PolniiPut);
        FileSave();
    }
} //FileSaveAs

void FileSave ()
{
    FileOutputStream OS = null; //FileOutputStream - faylga
//yozish

    String sz = ta.getText();
    buf = sz.getBytes();

    if(PolniiPut==" " || PolniiPut==null)
    {
        System.out.println("No File Selected!!!");
    }
}

```



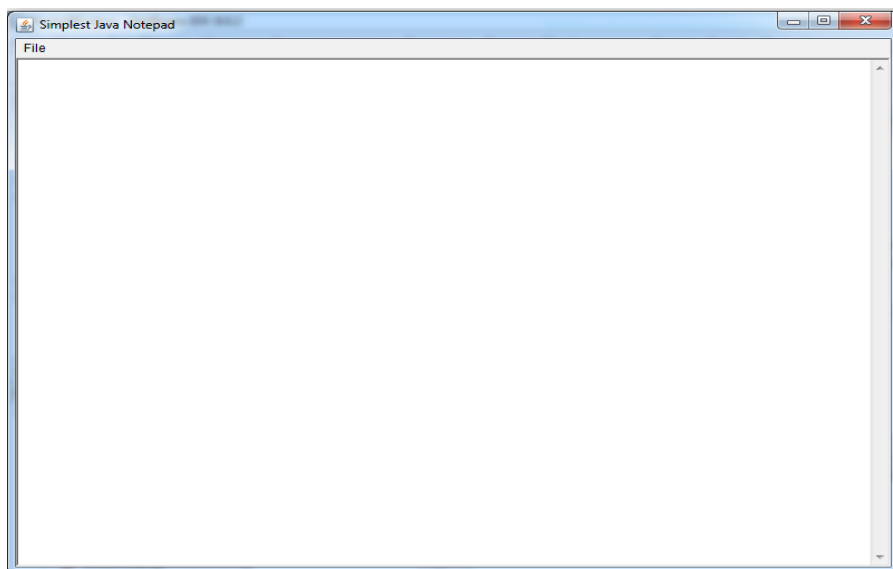
```

    }
else // fayl tanlangan
{
    try
    {
        OS = new FileOutputStream(PolniiPut);
        OS.write(buf);
        OS.close();
    } //try
catch (IOException ex)
{
    System.out.println(ex.toString());
}
} //end_else
} //FileSave

public static void main(String args[])
{
    FrameWindow frame;
    frame = new FrameWindow("Simplest Java Notepad");
//yangi oyna yaratish
    frame.setVisible(true); // ko'rinarli qilish
} //main funksiyasi yakunlandi
} //FrameWindow yakunlandi

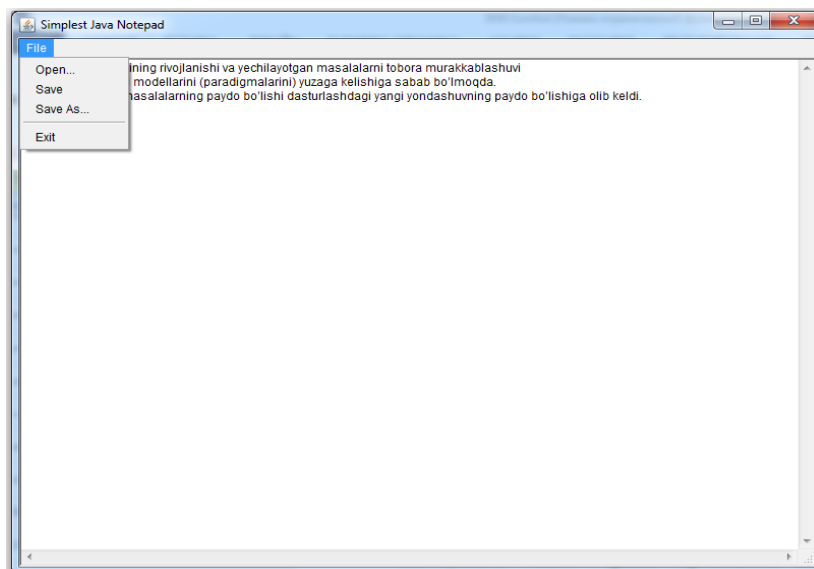
```

Ushbu dastur kompilyatsiya qilinishi natijasida quyidagi oyna hosil bo'ladi:



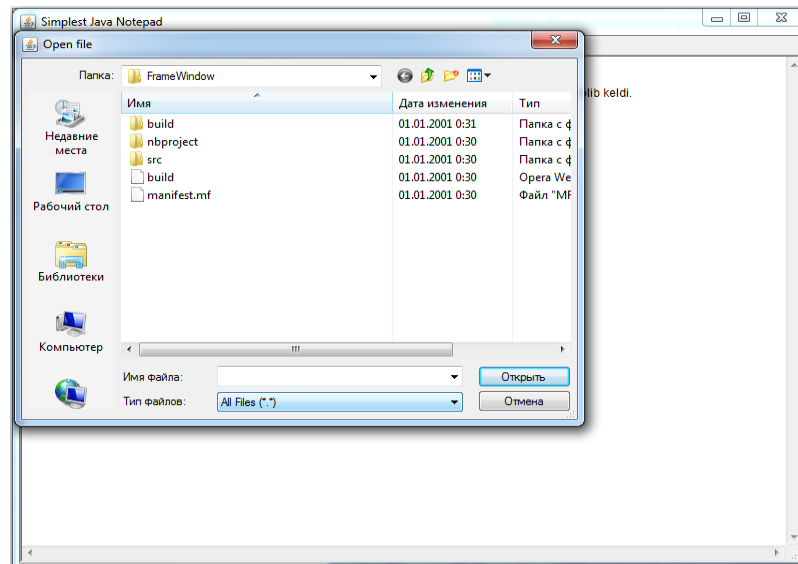
3.4-rasm. Sodda ko'inishdagi matn muharriri

Hosil bo'lgan dastur oynasida matn yozish uchun maydon va bitta menyu mavjud. Bu menyu to'rtta buyruqdan iborat va ularning har birining vazifasi dasturlashtirilgan. Quyida menyu buyruqlari bilan tanishamiz:



3.5-rasm. File menyusi buyruqlari

Ko'rib turganingizdek, yaratilgan matnni saqlash uchun **Save** buyrug'i, qayta saqlash uchun **SaveAs** buyrug'i, boshqa faylni ochish uchun **Open** buyrug'i va dasturdan chiqib ketish uchun **Exit** buyruqlari mavjud. **Open** buyrug'ini bajartirganimizda quyidagi muloqot oynasi hosil bo'ladi:



3.6-rasm. Faylni ochish muloqot oynasi

Bu sodda ko'rinishdagi matn muharriri dasturi Java tilining AWT kutubxonasi imkoniyatlaridan foydalanib yaratildi. Java dasturlash tilida mavjud boshqa kutubxonalardan foydalanib dasturni yanada kengaytirish mumkin.

3.4. Sodda ko'rinishdagi Paint dasturini yaratish

Bugungi kunda rasmlar va tasvirlar bilan ishlovchi ko'plab grafik dasturlar mavjud. Bular sirasiga Paint, PhotoPaint, Adobe Photoshop, CorelDraw kabi ko'p foydalanuvchiga ega dasturlarni keltirish mumkin. Java tilining grafik imkoniyatlarini ko'rsatib berish maqsadida bu bo'limda sodda ko'rinishdagi, rasm chizishga mo'ljallangan Paint dasturini keltiramiz. Dastur kodi quyidagicha:

```
import java.util.List;
import java.util.ArrayList;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class Paint extends JFrame {
    public static final int CANVAS_WIDTH = 500;
    public static final int CANVAS_HEIGHT = 300;
```

```

    public static final Color LINE_COLOR = Color.RED;

    private List<PolyLine> lines = new
ArrayList<PolyLine>();

    private PolyLine currentLine;

    public Paint() {

        DrawCanvas canvas = new DrawCanvas();

        canvas.setPreferredSize(new Dimension(CANVAS_WIDTH,
CANVAS_HEIGHT));

        canvas.addMouseListener(new MouseAdapter() {

            @Override

            public void mousePressed(MouseEvent evt) {

                currentLine = new PolyLine();

                lines.add(currentLine);

                currentLine.addPoint(evt.getX(), evt.getY());

            }

        });

        canvas.addMouseMotionListener(new
MouseMotionAdapter() {

            @Override

            public void mouseDragged(MouseEvent evt) {

                currentLine.addPoint(evt.getX(), evt.getY());

                repaint();

            }

        });

        getContentPane(canvas);

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        setTitle("Paint");

        pack();

        setVisible(true);

```

```

    }

    private class DrawCanvas extends JPanel {

        @Override

        protected void paintComponent(Graphics g) {
super.paintComponent(g);

            g.setColor(LINE_COLOR);
            for (PolyLine line: lines) {
                line.draw(g);
            }
        }
    }

    public class PolyLine {
private List<Integer> xList;
private List<Integer> yList;

        public PolyLine() {
            xList = new ArrayList<Integer>();
            yList = new ArrayList<Integer>();
        }

        public void addPoint(int x, int y) {
            xList.add(x);
            yList.add(y);
        }

        public void draw(Graphics g) {
            for (int i = 0; i < xList.size() - 1; ++i) {
                g.drawLine((int)xList.get(i), (int)yList.get(i),
(int)xList.get(i + 1),
                    (int)yList.get(i + 1));
            }
        }
    }
}

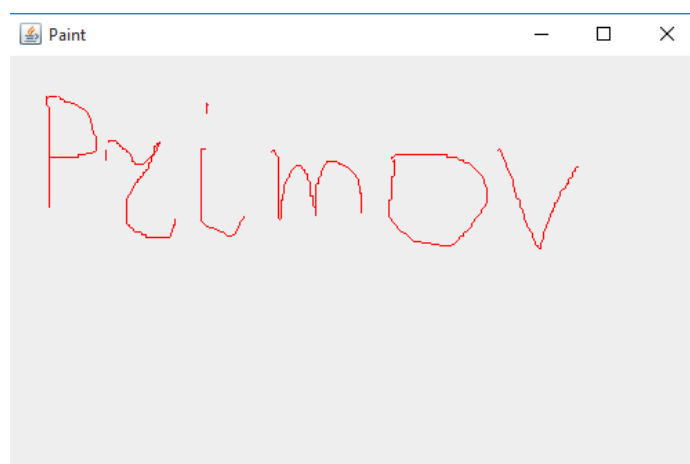
```

```

public static void main(String[] args) {
    SwingUtilities.invokeLater(new Runnable() {
        // Run the GUI codes on the Event-Dispatching
        thread for thread safety
        @Override
        public void run() {
            new Paint();
        }
    });
}
}

```

Yuqorida keltirilgan Paint dasturini komilyatsiya qilganimizda, quyidagicha oyna hosil bo'ladi:



3.7-rasm. Paint dasturi interfeysi

Hosil bo'lgan oynada sichqoncha chap tugmasi yordamida istalgancha chizish ishlarini olib borish mumkin. Bu dasturdan mukammal ko'rinishdagi grafik muharrir darajasidagi amaliy dasturlar yaratishda foydalanish mumkin.

XULOSA

Vatanimiz mustaqillikka erishgach, hayotimizning barcha jabhalarida bo'lganidek, xalq ta'limi tizimida ham ulkan islohotlar yo'lga qo'yildiki, bunda ta'lim tarbiya jarayoniga zamonaviy axborot texnologiyalarini joriy etish, ta'limni kompyuterlashtirish muammolarini qal qilish muhim ahamiyat kasb etadi.

Axborotlashgan jamiyatda yashar ekanmiz, dasturlash tillariga bo'lgan ehtiyoj kundan kunga o'sib bormoqda. Java dasturlash tili bugungi kunga kelib ko'plab sohalarda qo'llanilmoqda, shuningdek bu til web ilovalar yaratishga qulay bo'lgan til hisoblanadi.

Ushbu bitiruv malakaviy ishida qo'yilgan "Grafika va matn bilan ishlashda JAVA tilining AWT kutubxonasidan foydalanish" mavzusi ostida qo'yilgan maqsad va unga mos vazifalarni bajarish davomida quyidagi natijalarga erishildi:

- 1) Java tilining grafika va matn bilan ishlovchi komponentalari batafsil o'rganib chiqildi va ish tarkibiga kiritildi;
- 2) Java tilida applet hosil qilish jarayonlari o'rganib chiqildi va unda matnlarni tahrirlash usullari o'rganildi va amaliy masalalar yordamida ko'rsatildi;
- 3) Swing oynasi, uni yaratish haqida va bu oynaga biror vazifani bajarishga mo'ljallangan tugmalar joylashtirish masalalari ko'rib o'tildi;
- 4) Ishda asosiy grafik primitivlarni hosil qilish haqida umumiy tushunchalar keltirildi;
- 5) Grafik primitivlarni hosil qiluvchi metodlardan foydalangan holda applet oynasida turli xil shakllar hosil qilish dasturi yaratildi;
- 6) Tasvirlarning o'lchamlarini o'zgartirish, rang tanlash va o'rnatish ishlari o'rganib chiqildi va ish tarkibiga kiritildi;
- 7) Tasvirlarni tashqi fayldan o'qish jarayoni o'rganildi va amaliy masalalar yordamida yoritildi;
- 8) Harakatlanuvchi shakllar hosil qilish jarayoni o'rganib chiqildi va amaliy dasturiylar yordamida yoritildi;

- 9) Grafik tasvirni yaratish va uni tugmalar yordamida boshqaruvchi dasturiy vosita yaratildi;
- 10) Java tilida sodda matn muharriri dasturi yaratildi va ish tarkibiga kiritildi;
- 11) Sichqoncha yordamida chizmalar chizishga mo'ljallangan sodda ko'rinishdagi Paint dasturi yaratildi va ish tarkibiga kiritildi.

Ushbu bitiruv ishida tuzilgan animatsiyani, matnni, tasvirni boshqarishga mo'ljallangan dasturiy vositalardan foydalanib, multimedialli elektron o'quv qo'llanmalar yaratish va undan ta'lim berish jarayonida foydalanish samarali natijalarga erishishga imkoniyat yaratadi.

FOYDALANILGAN ADABIYOTLAR VA INTERNET RESURSLARI

1. Герберт Шилдт. «Java. Полное руководство» 8-ое издание. Язык русский. М: Издательский дом «Вильямс», 2012, -1104.
2. Майк МакГрат. «JAVA: Программирование для начинающих» Пятое издание. Москва 2016. 193 стр.
3. Bjarne Stroustrup. The C++ Programming Language (3th Edition). Addison-Wesley, 1997.
4. D.S. Malik. C++ Programming: From Problem Analysis to Program Design. Fifth Edition. Course Technology, 2011.
5. Хорстманн К. С., Корнелл Г. – Java 2. Том 1. Основы Java. М: Издательский дом «Вильямс», 2007 – 896 с. 7-ое издание.
6. Хорстманн К. С., Корнелл Г. – Java 2. Том 2. Тонкости программирования. М: Издательский дом «Вильямс», 2014 – 1874 с. 9-ое издание.
7. P.J.Deitel, H.M.Deitel. Java:how to program. NJ:Prentice Hall, 9 th edition. 2012.
8. Н.А.Вязовик. Программирование на Java. ИНТУИТ. 2-е издание. 2016 г. 604 стр.
9. И.В.Парамонов. «Разработка приложений баз данных с использованием средств объектно-реляционного отображения». Ярославль 2008 г. 44с.
- 10.Е.А.Eshboyev, F.G.Qlicheva “Java tilida dasturlash asoslari”. O‘quv uslubiy qo‘llanma. Q-2018 y. 205 s.
- 11.О.М.Shukurov, E.A.Eshboyev, B.H.Shovaliyev – “Delphi va C++ algoritmik tillarida dasturlash” – Qarshi-2012 y., 228 s.
- 12.Е.А.Eshboyev, F.Yu.Shodiyev, F.G.Qlicheva “C++ tilida dasturlash”. O‘quv uslubiy qo‘llanma. T-2015 y. 205 s.

Elektron manbalar:

13. www.eclipse.org
14. www.oracle.com
15. www.acm.timus.ru
16. www.acm.tuit.uz
17. www.algo.ubtuit.uz
18. www.codeforces.ru
19. www.cyberforum.ru
20. www.cplusplus.com
21. www.acm.dvipion.ru
22. www.acm.sgu.ru
23. www.acm.msu.ru
24. www.topcoder.com
25. www.olympiads.ru
26. www.ziyonet.uz