O’ZBEKISTONRESPUBLIKASIALOQAAXBOROTLASHTIRISHVATEL EKOMMUNIKATSIYATEXNOLOGIYALARIDAVLATQO’MITASITOSH KENTAXBOROTTEXNOLOGIYALARIUNIVERSITETI

Dasturiy Injinering Fakulteti

ATDT kafedrasi

KURS ISHI

Mavzu: Temir yo’l vokzali ishini avtomatlashtirish

Bajardi: 215-12-guruh talabasi

Ergashev Dilshod

Tekshirdi:Babamuhammedova M.

Toshkent – 2015

Mundarija

I.Kirish

Insoniyat XXI asrga qadam qo`ydi. Respublikamizda olib borilayotgan islohatlarning zamirida Axborot tizimlarining roli benihoyat kattadir. Hammamizga ma’lumki, bugungi jadal sur’atlarda rivojlanayotgan hayotni texnika – texnologiya, internet, telefon, kompyuterlarsiz tasavvur qilib bo’lmaydi, albatta. Bu texnik qurilmalar bizning kundalik ishlarimizni shu qadar yengillashtirib yuborganki, endilikda mazkur mahsulotlardan voz kechish tugul, ulardan uzoqlashishning ham imkoni mavjud emas. Kashf qilinayotgan yangi, zamonaviy texnik vositalar yaxshi. Biroq ulardan meyor darajasida foydalanilsa, nur ustiga a’lo nur bo’lar edi.

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. Respublikamizda jamiyatimizni axborotlashtirish maqsadida bir qancha qaror va qonunlar qabul qilindi. Masalan, 1993- yil 7- may va 2003- yil 11- dekabrdagi «Axborotlashtirish to'g'risida»gi qonun, 2002- yil 30- maydagi «Kompyuterlashtirish va informatsion-kommunikatsion texnologiyalarni qo'llashni yanada rivojlantirish» haqidagi qaror, 2003- yil 11- dekabrdagi «Elektron raqamli imzo haqida»gi qonun va 2004- yil 29- apreldagi «Elektron hujjat yuritish» haqidagi qonun fikrimizning dalilidir. Informatika vositalari jamiyatimizning barcha jabhalariga tobora kirib borayotgani, axborotni tez va sifatli qayta ishlash malakasi o'sib kelayotgan har bir yoshning turmush talabiga aylanishini ko'rsatib bermoqda. Axborotning qimmatbaho tovarga aylanib borayotgani, informatika fanining nufuzi va ahamiyati o'sib borayotganidan dalolatdir.

Men OYDT fanidan kurs ishi bajarish uchun Java dasturlash tilini o’rganishni maqsad qilib oldim. Java dasturlash tilining imkoniyatlari bilan tanishishni, standard paketlar kutubxonalarining metodlari bilan tanishishni, ma’lumotlar bazasi bilan ishlashni, ma’lumotlar bazasida so’rovlar yozishni jadvallar yaratishni, yaratilgan ma’lumotlar bazasi ma’lumotlari bilan ishlashni, Java dasturlash tilida grafik user interfacelar bilan ishlashni, Java dasturlarini ma’lumotlar bazasi bilan bog’lashni o’rganishni **maqsad qilib oldim.**

Masalaning qo`yilishi:

II.Nazariy qism.

2.1 Java dasturlash tili haqida umumiy tushuncha

Hammamizga ma’lumki , dasturlash tillari 2 xil ko’rinishda bo’ladi:

1.Strukturaviy dasturlash tillari

- FORTRAN (IBM Mathematical FORmula TRANslation)
- BASIC (Beginner‘s All Purpose Symbolic Instruction Code)
- COBOL (Common Business Oriented Language)
- PASCAL
- C

2.Obyektga yo’naltirilgan dasturlash tillari

- C# (C Sharp)
- C++
- Object Pascal
- Java
- SmallTalk
- Ada

Quyidagilardankelibchiqqanholda, bizJavadasturlashtilihaqidato’xtalibo’tamiz.

Kompyuterdasturlariyokidasturiytaminotio'ziorqalio'lchanmalarmajmuiniko'rsatadiki, bularnikompyuteraniqmisoluchuniechishuchunqulaydir. Dasturchilar ularni maxsus tillar Bisic, C, C++ yoki Java orqali tuzadilar. Muammo shundaki, har xil tipdagi dasturlarni bir necha tilni qo'llashgan. Misol uchun, biznes uchun dasturchilar dastur tuzganda, ko'proq COBOL (the Comman Bisiness Oriented language) ni ishlatadi. Ilmiy dastur yaratuvchi dasturchilar FORTRAN (the Formula TRANslator) ni ishlatishadi. Sistemali dastur yaratuvchilar C yoki C++ foydalanishadi. Bugungi kunda kompyuter olamida “issiq mavzu” sifatida Internet va World Wide Web (“ Dunyo o'rgimchak to'ri ”) tashkil etadi. Internet bilan ishlashni yaxshilash uchun yangi programmalashtiruvchi til paydo bo'lishi lozim bo'ladi. Java shunday tillardan xisoblanadi. Hozirgi kompyuter industriyasidagi yangi dasturlash tiliga. Tarmoqdagi programmalashtirishning imkoniyatlarini va yana Java (JDK) ishlab chiqaruvchi komplektini nusxasini Sun Microsystems (Java ni ishlab chiqargan firma) firmasi tomonidan yuklab olishini o'rganish mumkin bo’ladi. Bu dasturni o'rgangandan keyin quyidagi asosiy kalit konseptsiyalarni bilish kerak bo'ladi:

* Java – bu dasturlash tildir, qaysini dasturchilarni avtanom dastur va brovzer bilan ishlovchi appletlarni tuzish mumkin.

* Avtanom dastur brovzer orqali ishlamaydi.

* Applet brauzer orqali ishlaydi.

* Appletlar kompyuter turiga qaramaydi, bu shuni ko'rsatadiki bir applet har xil turdagi Windows95, Macintosh yoki UNIX tizimlarida bir xilda ishlaydi.

* Java ning boshqa dasturlash tillaridan farqi shundaki, u aniq bir protsessor uchun kompilyatsiya qilmay, vertual mashina uchun kod xosil qiladi. Bu kodni brouzer aniq bir protsessor uchun binar kodga aylantiradi.

* Siz Java qayta ishlovchi komplekti Sun Microsystem firmasiningWeb tarmog’idan http://java.sun.com manzil orqali bepul yuklab olishingiz mumkin.

* Hot Java – bu brouzer bo'lib, Java dasturlash tilida yozilgan.

JAVA ning C & C++ dan ustunligi

- WORA - Write Once, Run Anywhere (portable).
- Xavfsizlik (ishonchyo’qkodnixavfsizishgatushirish).
- Xotirani xavfsiz boshqarish (avtomat ravishda musorlarni yig’adi)
- Tarmoqga dasturlash
- Ko’p oqimli (Multi-thread) dasturlash
- Dinamik & kengaytirish
    - Class lar alohida fayllarda saqlanadi
    - Kerak bo’lsa ishlatiladi
    - Dinamikravishdaimkoniyatinioshirishxammumkinkerakbo’lsa.

Quyidagi farqlar bilan Java C/C++ dan ajralib turadi

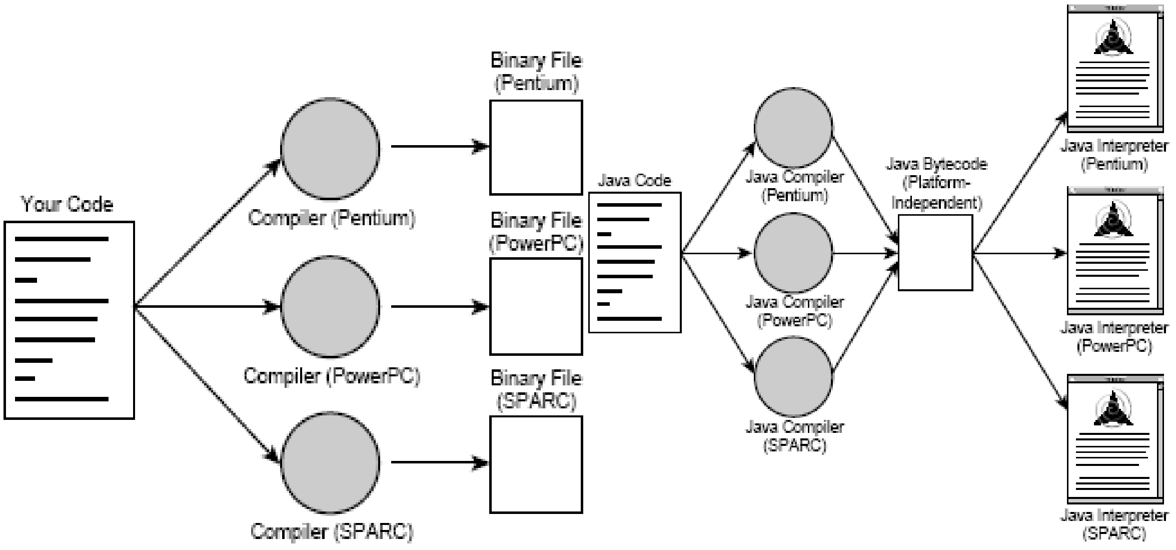
- header fayllar yo’q
- Preprocessor’lar yo’q

- Goto yo’q
- Unicode belgilar
- Avtomat musorlarni yig’adi
- Ko’rsatkich (pointer) yo’q
- Operatorlarni peregruzga qilish yo’q
- 100% OYDT – hamma narsa class
- Global o’zgaruvchi va funksiyalar yo’q

Java tili Sun Microsystems tomonidan 1991 yil bitovoy ustroystvaga dasturlar ishlab chiqishdan boshlangan. Java dasturlash tili James Gosling tomonidan yaratilgan. Java’ning birinchi nomi “Oak” bo’lgan.

Birinchi ofitsialniy versiya - Java 1.0, 1996 yil taqdim qilingan.

Platforma



Ob’ekt va ob’ektga yo’naltirilgan dasturlash. Oddiy so'z bilan aytganda ob'ekt bu predmet xisoblanadi. Applet yaratayotganda, biz odatda biror ma'lumotni saqlash uchun o'zgaruvchilardan foydalanamiz. Ob'ektga yo'naltirilgan dasturlash tilini ishlatayotganimizda, biz sistemamizni to'ldiruvchi predmetlar va predmet ustida bajariluvchi operatsiyalarga e’tiborimizni qaratamiz.

Misol uchun, ob'ekga bog’liq fayllarda biz petsat qilish, ekranga chiqarish, o'chirish va faylga o'zgartirish kiritish operatsiyalarini aniqlashimiz kerak. Java da ob'ekni aniqlash uchun sinfdan foydalanamiz. Bu sinflarni bir dastur uchun yaratib, boshqa ob'ektlarda ham ishlatishimiz mumkin bo’ladi.

HTML bilan bog’liq holda ishlash. Appletni ishga tushirish uchun Java HTML faylga bir necha tushunchalarni kiritadi. Brouzerni yoki appletwiever bu tushununchalarni applet nomini, oyna o'lchamini va boshqalarni aniqlash uchun ishlatadi. Bunday vaziyatda HTML ning qo'shimcha tushunchalarini appletingizni brouzer oynasida chiqarish uchun ishlatish mumkin.

getParameter funktsiyasidan foydalanib, HTML fayli doirasida dasturning qiymat qabul qilishini o'rganish lozim bo’ladi. Bundan tashqari o’rganishni tamomlagach quyidagi kalit kontseptsiyalarni bilish kerak:

* HTML APPLET tushunchasi bir necha qo'shimcha atributlari bo'lib, brouzer oynasida appletni boshqarish uchun ishlatasiz.

* PARAM atributdan foydalanib, HTML fayli doirasida, va bu qiymatlarni

dastur getParameter funksiyasidan foydalanib qabul qilishi mumkin.

* HTML fayliga qiymatlarni berishni qo'llab appletimizning foydalanuvchilar

uchun qulaylik bo'ladi.

Massivlar bilan ishlash. Dasturimiz ish vaqtida o'zgaruvchilarda xabarni saqlash mumkin. Shu vaqtgacha siz ishlatgan hamma o'zgaruvchilar faqat bir qiymat qabul qilish mumkin edi. Shunday xolatlar bo'lishi mumkinki dasturimizga bir necha qiymatlarni saqlashi kerak bo’lishi mumkin.

Misol uchun 50 ta test natijalari, 100 ta kitob nomi yoki 1000 ta fayl nomi. Dasturimizga bunday ma'lumotlarni saqlash uchun massiv deb nomlanuvchi maxsus strukturadan foydalanamiz. Massivni e'lon qilish uchun dasturimizda uning nomini, elementlar sonini va ular tipini ko'rsatishimiz lozim bunda appletda massivni ishlatishni o'rganamiz.

Massiv– bu bir yoki bir nechta qiymatni saqlovchi o'zgaruvchidir. O'zgaruvchilar kabi massiv ham biror bir tipli bo'lishi (misol uchun int, float yoki char) va nomlanishi kerak. Bundan tashqari, massiv saqlovchi elementlar sonini ko’rsatishimiz kerak. Bu hamma qiymatlar bir xil tipda bo'lishi kerak. Boshqacha aytganda, applet har xil tipdagi qiymatlarni bir massiv doirasiga joylashtira olmaydi. Quyidagi operator 100 ta test natijalarni qabul qiluvchi TestScores nomli massiv yaratadi:

int TestScores [ ] = new int [100];

Kompilyator new opertorini uchratganda bu int tipli 100 ta qiymat uchun xotiradan joy ajratadi. Massiv saqlovchi qiymatlar massiv elementlari deyiladi. Bu xolda [ ] qavslar TestScores o'zgaruvchi massiv ekanligini bildiradi.

O'zgaruvchini aniqlashdan oldin Java initsializatsiya qilish uchun taminlash operatoridan foydalanishga ruxsat beradi. Misol uchun, quyidagi muloxaza int tipli index nomli o'zgaruvchini aniqlab va uning qiymatini 1 ga initsializatsiya qildi: int indexq1;

Annalogik xolda, massiv n aniqlaganimizda, Java massiv elemenlarini initsializatsiya qilishga imkon beradi. Bu tarzda massivni aniqlash, new operatori va massiv o'lchamini ko'rsatish kerak bo'lmaydi. Buning o'rniga kompilyator kirgizilgan qiymatlarning nomiga qarab massiv o'lchamini aniqlaydi. Quyidagi muloxaza, Values massivini yaratib intsializatsiya qiladi:

int Values []q{100 200 300 400 500};

Bu xolda kompilyator massiv uchun xotiradan5 elementga joy ajratadi.

Javada paketlar yaratish va ularni o’zaro bir – biri bilan bog’lanishini amalga oshirish.

Java paket (package) – bu klasslarni ma’lum bir struktura asosida joylashtirish, yani klasslarni namespace larga bo’lib tashlash. Java packetlar **JAR** faylda ham bo’lishi mumkin. JAVA packetlar classlarning manosiga qarab bo’linadi. Paketlar quyidagi toifalarni o’z ichiga oladi:

Classlar

Interfacelar

Enum toifalar

Javada sinflar tushunchasi. Sinf Java ob'yektga - yo'naltirilgan dasturlash tilining asosiy qismi sanaladi. Bunda sinf mazkur ob'yektga mos keluvchi va funksiyalar foydalanuvchi (metodlar) a'zolarni gruppalashtiradi. Sinf appletingizga ob'ektning xamma atrebutlarini aniqlashga imkon beradi. Berilganlarni va ob'ekt kodini bir o'zgaruvchiga guruxlash orqali dasturlashni qisqartirish va chiquvchi kodni qayta ishlatish imkoniyatini oshirsa bo’ladi.

* Appletda sinfni aniqlash uchun uning nomini, sinf a'zolarini va sinf funksiyasini (metod) ko'rsatishi kerak.

* Ob'ektning nusxasini olish uchun new operatorini ko'rsatamiz.

* Applet sinf a'zolariga qiymat ta’minlash uchun “ . ” nuqta operatoridan foydalanadi.

* Dasturingiz sinf a'zolarini chaqirish uchun “ . ” nuqta operatoridan foydalanadi.

Class larni yaratish uchun class katlit so’zidan foydalanish kerak.

Misol:

```
class MyClass {
        //o’zgaruvchilar, constructor va methodlarni e’lon qilish
}
```

Class larni e’lon qilish quyidagi tartibda bo’ladi:

- *Access modifiers public, private, final va hk.*
- *Class nomi.*
- *Ota class (superclass) nomi, odatda extends katlir so’zidan keyin yoziladi. Class (subclass) faqat bitta classdan nasl olishi mumkin.*
- *Vergul (,) bilan ajratilgan interface’lar, odatda interfacelar implements katil so’zidan keyin yoziladi.*
- *Class tanasi, firurali qavus {}.*

Methodlarni yaratish 6 ta etapdan iborat:

- Access Modifiers - *public, private*
- Qaytarish qiymati (return type) – method qaytaradigan qiymat toifasi yoki void agar method hech narsa qaytarmasa.
- Method nomi
- Parametrlar – () qovusichigaparametrlarvergul (,) bilanajratilibyoziladi, parametrtoifasivanomiko’rsatilibo’tilishikerak, agarmethodhechqandayparametrolmasa () qovusichibo’shqoldiriladi.
- Exception lar – hatoliklarni ushlash

- Method tanasi – {} firurali qovus ochilib yopilgani, tanaga method qiladigan amallar yoziladi

Method overloading – bitta methodni bir necha xil usul bilan e’lon qilinishi, ya’ni method parametrlari va qaytarish toifalari har hil.

Contructor - bu class dan yangi obyekt yaratilganda obyektning boshlang’ich qiymatlarini berish.

- Constructor nomi class nomi bilan bir hil bo’ladi
- Contructorlarni overload qilsa ham bo’ladi
- Constructorlar qiymat qaytarmaydi

Class lardan nusxa olish uchun **extends** kalit so’zidan foydalaniladi. Nasl olingan ota (superclass) class ning public va protected e’lon qilingan o’zgaruvchilar va methodlarini ishlata oladi. Superclass ichidagi methodlarni subclass da qayta yozish mumkin.

- Interface –static va final o’zgaruvchilar va methodlar yig’indisi. Qisqa qilib aytganda prototiplar to’plami.
- Agar class interface dan nasl olsa, interface ichidagi hamma methodlarni class da yozib chiqishi kerak.

Java dasturlash tilidagi mavjud paketlarga quyidagilar kiradi:

java.lang	- tilning asosiy funksiyalari va asosiy tiplar
java.util	- collection data structure classes
java.io	- file operation
java.math	- matematik metodlar
java.nio	- New I/O framework
java.net	- tarmoq funksiyalari, sockets, DNS lookups, ...
java.security	- key generation, encryption and decryption
java.sql	- Java Database Connectivity (JDBC), bazalar bilan ishlash
java.awt	- Abstract Window Toolkit GUI
javax.swing	- GUI component

java.applet - applet'larni yaratish uchun klass

java.lang paketi **import** siz ham ishlatiladi.

Paketni ishlatish uchun **import** kalit so'zidan foydalanish kerak.

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

java.awt.event paketidagi hamma klaslarni import qiladi. Agar:

```
import java.awt.event.ActionEvent;
```

ActionEvent klasi import qilinadi.

JAR fayl – bu class larni bitta joyga yig'ish, qisqa qilib aytganda arxivlab qo'yadi. JAR faylini yaratish oddiy buyrug'i: **jar** cf*jar-filenameinput-files*

Misol:

jar *cfmy_classes.jar bir.class ikki.class*

jar cvf *project.jar MyClass.class images*

v opsiyasi arxivlash jarayoni haqida to'liq ma'lumot beradi.

```
added manifest
adding: MyClass.class(in = 462) (out= 291)(deflated 37%)
adding: images/(in = 0) (out= 0)(stored 0%)
adding: images/logo.gif(in = 4178) (out= 4001)(deflated 4%)
```

JAR fayl ichini ko'rish:

jar tf jar-filename

Yuqoridagi proyektning klaslarini jar faylga joylashtiramiz:

```
dedexes@localhost:~/programs/java$ jar cvf classes.jar One
added manifest
adding: One/(in = 0) (out= 0)(stored 0%)
adding: One/Monkey.class(in = 390) (out= 275)(deflated 29%)
```

```
adding: One/Two/(in = 0) (out= 0)(stored 0%)

adding: One/Two/Cat.java(in = 92) (out= 85)(deflated 7%)

adding: One/Two/Dog.java(in = 92) (out= 85)(deflated 7%)

adding: One/Two/Cat.class(in = 385) (out= 274)(deflated 28%)

adding: One/Two/Dog.class(in = 385) (out= 275)(deflated 28%)

adding: One/Monkey.java(in = 94) (out= 84)(deflated 10%)
```

JAR faylni ishlatish uchun **JAR** faylni **CLASSPATH** ga qo’shish kerak:

```
dedexes@localhost:~/programs/java/main$export
CLASSPATH=/home/sher/programs/java/main/classes.jar:.

dedexes@localhost:~/programs/java/main$ javac TestPackage.java
```

2.2 Ma’lumotlar bazasi. Relyatsiyon ma’lumotlar bazasi.

Relyatsion ma'lumotlar bazasini boshqarish tizimi - o`zining ma'lumotlar strukturasi oddiyligi bilan, foydalanuvchi uchun  jadval ko`rinishida joylashtirilishi bilan va ma'lumotlar ustidan oson hisob-kitob amallarini bajarish imkoniyati mavjudligi bilan ajralib turadi.

Hozirda relyatsion ma’lumot bazalari o’zining qulayligi tufayli keng miqyosda ishlatilmoqda. Relyatsion ma’lumot bazalari ma’lumotlarni jadvallarga joylashgan va jadvallar orasida mos bog’liqliklarni, ya’ni munosabatni (relyatsiyani) o’rnatishga asoslangan. Ular jadvallar orasidagi turli bog’liqliklarni o’rnatish, ma’lumot kiritish shakllarini yaratish, hisobot shakllarini chiqarish, turli so’rovlar (Zaprosi) tuzish imkonini beradi.

Relatsion model ma'lumotlarni ikki o`lchamli jadvalda tartiblashga asoslangan.  Har bir relatsion jadval ikki o`lchamli massivdan iborat bo`ladi va quyidagi husuisiyatlarga ega:

-Jadvalning bir ustunidagi barcha kataklar bir hil tipga mansub(misol uchun bir ustun barcha elementlari tipi simvolli yoki raqamli);

-Jadvalda bir xil qatorlar mavjud emas;

-Ustun va qatorlar ketmaketligi turlicha bo`lishi mumkin;

Relatsion ma'lumotlar bazasi boshqarish tizimining asosiy tushunchalari bu:

* Atribut;

* Relation;

* Kortej;

2.3 Java Swing xaqida.

Java/SWING nima – JAVA uchun GUI Framework

• JAVA dasturlarini “look and feel” holatiga keltirish

• Java Foundation Classes (Sun Microsystems) bir qismi

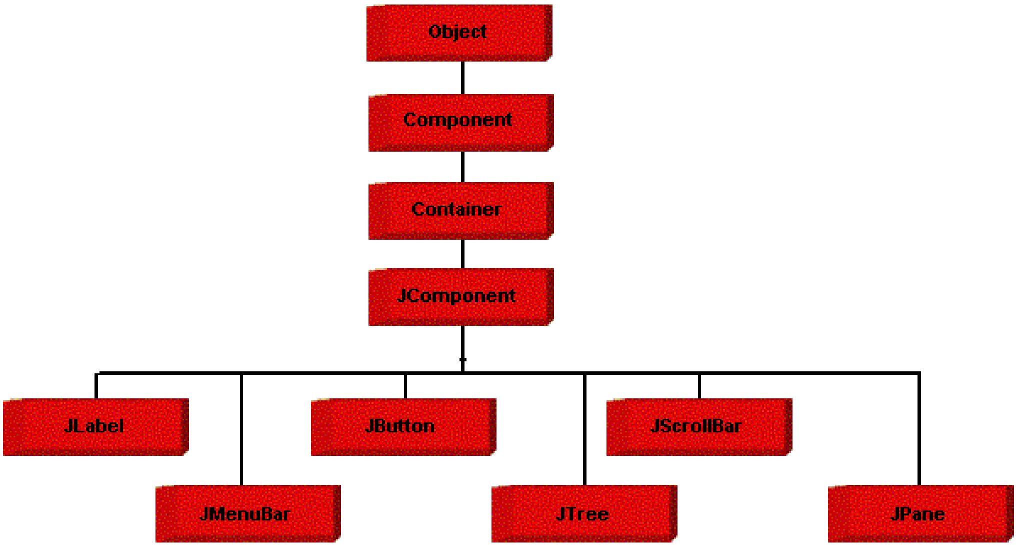
• IFC (Netscape) va JFC (Sun Microsystems) qo’shilgani

• Java Standard Edition 1.2 versiyasidan boshlab paket shaklida qo’shilgan

• GUI controllar java dastur orqali generatsiya qilinadi

• Ko’rinishi va funksionallkin jihatidan Windows, Mac va Linux OS’laridaka

Java Swing class iearxiyasi



Swing API 18 ta public paketlardan iborat

javax.accessibility

javax.swing

javax.swing.border

javax.swing.plaf

javax.swing.plaf.basic

javax.swing.plaf.metal

javax.swing.text

javax.swing.text.html

javax.swing.text.html.parser

javax.swing.colorchooser

javax.swing.event

javax.swing.filechooser

javax.swing.plaf.multi

javax.swing.plaf.synth

javax.swing.table

javax.swing.text.rtf

javax.swing.tree

javax.swing.undo

SWING Layots

AWT va Swing klaslarining layout manager’lari:

BorderLayout
BoxLayout
CardLayout
FlowLayout
GridBagLayout
GridLayout
GroupLayout
SpringLayout

Java/SWING BorderLayot

BorderLayout

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

III. Amaliy qism.

3.1 Ma’lumotlar bazasi tuzilmasi.

MySQL haqida ma`lumot

MySQL - bu eng mashhur va juda ko‘p foydalaniladigan ma'lumotlar bazasini boshqarish tizimi(MBBT) hisoblanadi. Bu tizim juda katta ma'lumotlar bilan ishlash uchun yaratilmagan, aksincha biroz kichik hajmdagi bazalar bilan katta tezlikda ishlash uchun yaratilgan. Uning asosiy ishlash doirasi, saytlar hisoblanadi. Hozirgi kunda juda ko‘p sayt va bloklarning ma'lumotlari aynan shu MBBT saqlanadi.

Ho‘sh, saytlarning nimalari bazada saqlanishi mumkin? Dastlab, ma'lumotlar saqlash uchun dasturchilar fayllardan foydalanishgan, ya'ni fayl ochilib kerakli ma'lumotlar u yerga saqlanib, kerakli paytda chaqirib ishlatilgan. Keyinchalik fayllardan voz kechilib(noqulayliklar yuzaga kelgan, ya'ni faylni ochish, o‘qish, yopish,.. ko‘p vaqtni olib qo‘ygan, u yerdan qidirish, xullas juda ko‘p), ma'lumotlar bazasiga o‘tilgan. Bazada saytdagi maqolalar, sayt foydalanuvchilari haqidagi ma'lumotlar, sayt kontentlari, qoldirilgan kommentariyalar, savol-javoblar, hisoblagich natijalari va shunga o‘xshash juda ko‘p ma'lumotlar saqlanadi. MySQL shunday ma'lumotlarni o‘zida saqlaydi.

MySQL - juda katta tezlikda ishlovchi va qulay hisoblanadi. Bu tizimda ishlash juda sodda va uni o‘rganish qiyinchilik tug‘dirmaydi.

MySQL tizimi tcx kompaniyasi tomonidan, ma'lumotlarni tez qayta ishlash uchun korxona miqyosida ishlatishga yaratilgan. Keyinchalik ommalashib, saytlarning asosiy bazasi sifatida yoyildi.

So‘rovlar SQL tili orqali amalga oshiriladi. Bu MBBT relyatsion ma'lumotlar baza hisoblanadi. Bu degani baza jadvallar, jadvallar esa ustunlardan tashkil topgandir.

MySQL MBBTi 2 xil turdagi litsenziyaga ega. Birinchisi tekin, ya'ni MYSQLni ko‘chirib olish va ishlatish hech qanday harajat talab qilmaydi va GPL(GNU Public Licenseb, GNU) litsenziyasiga asoslanadi. Ikkinchi turi, GPL shartiga ko‘ra, agar siz MySQL kodlarini biror dasturingizda ishlatsangiz, bu dasturingiz ham GPL(tekin) bo‘lishi kerak. Bu esa dasturchiga to‘g‘ri kelmaydi. Shuning uchun, bu dasturingizni pullik qilishingiz uchun MySQL pullik litsenziyasini sotib olishingiz kerak.

MySQL logotipi delfin hisoblanadi. Bu delfinni ismi "Sakila"dir. Bu logotipni OpenSource tuzuvchilaridan biri Ambrose Twebaze ga tegishlidir.

MySQL juda ko‘p operatsion tizimlar bilan ishlay oladi. Bularni yozadigan bo‘lsam:AIX, BSDi, FreeBSD, HP-UX, Linux, Mac OS X, NetBSD, OpenBSD, OS/2 Warp, SGI IRIX, Solaris, SunOS, UnixWare, Windows 95, Windows 98, Windows NT, Windows 2000, Windows XP, Windows Server 2000, Windows Vista, Windows 7,...

MySQL shved korxonasi MySQl AB ga tegishli bo‘lgan, keyinchalik SUN firmasi MySQL ni o‘ziniki qilib olgan. Bir necha yil oldin esa, gigant korxonalardan biriOracle, SUN firmasini sotib oldi va hozirda MySQL Oracle firmasi mahsuloti sifatida chiqib kelmoqda. Bu bir hisobda Microsoft SQL Server ga katta raqobatdosh ekanligini anglatadi.

MySQL bir necha serverlarning bir qismi hisoblanadi. Misol uchun, WAMP, AppServ, LAMP, Denwer,... Kliyentlar MySQL serveriga ma'lum bir kutubxonalar orqali ulanadi. MySQL ga quyidagi dasturlash tillari ulanib ishlashi mumkin: Delphi, C, C++, Java, Perl, Php, Python, Ruby va boshqalar.

3.2 Java dasturlarini ma’lumotlar bazasi bilan bog’lash

Java dasturlarini ma’lumotlar bazasi bilan bog’lash uchun bizga JDBC drayveri kerak bo’ladi.

JDBC Driver(Java Database Connectivity – Javada ma’lumotlar bazasi bilan bog’lanish) – platformaga bog’liq bo’lmagan, turli ma’lumotlar bazalari bilan bog’lanishga imkon beradigan, JavaSE da java.sql paketi tarkibiga kiritilgan standard drayver xisoblanadi.

JDBC – MBga bog’lanish, unda SQL so’rovlarini bajarishni ta’minlaydi.

Buning uchun quyidagi manzillardan MBga tegishli drayverlarning .jar paketini ko’chirib olamiz:

MySQL — http://dev.mysql.com/downloads/connector/j/

PostqreSLQ — http://jdbc.postgresql.org/download.html

Oracle — http://www.oracle.com/technetwork/database/features/jdbc/index-091264.html

MySQLga ulanish:

```
Class.forName("com.mysql.jdbc.Driver");

Connection conn = DriverManager.getConnection("jdbc:mysql://hostname:port/dbname","username", "password");

conn.close();
```

PostgreSQLga ulanish:

```
Class.forName("org.postgresql.Driver");

Connection connection = DriverManager.getConnection("jdbc:postgresql://hostname:port/dbname","username", "password");

connection.close();
```

ORACLEga ulanish:

```
Class.forName("oracle.jdbc.driver.OracleDriver");
```

```
Connection connection =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:mkyong","username","password");

connection.close();
```

1-qatorlarda biz MBga ulanish uchun drayverni ko’rsatyapmiz.

2-qatorlardaJDBC Manager bazaga bog’lanishni bajaradi va undan keyinchalik foydalanishimizga imkon yaratadi.

3-qatorlar bog’lanishni yopadi.

Imkoni bo’lsa JDBC drayverni try{} catch{} ichiga joylash kerak. Bu ish drayverning kompyuterimizda bor ekanligini va ishlashini nazorat qiladi.

```
try {
   Class.forName("com.mysql.jdbc.Driver");
} catch (ClassNotFoundException e) {
   System.out.println("Where is your MySQL JDBC Driver?");
   e.printStackTrace();
   return;
}
```

Javada bazaga bog’lanish uchun klass xosil qilib olamiz:

```
private static Connection getDBConnection() {
   Connection dbConnection = null;
   try {
      Class.forName(DB_DRIVER);
   } catch (ClassNotFoundException e) {
      System.out.println(e.getMessage());
   }
   try {
```

```
            dbConnection = DriverManager.getConnection(DB_CONNECTION,
    DB_USER,DB_PASSWORD);
            return dbConnection;
        } catch (SQLException e) {
            System.out.println(e.getMessage());
        }
        return dbConnection;
    }
```

Quyida MB ichida jadval xosil qilish metodini yaratamiz:

```
private static void createDbUserTable() throws SQLException {
    Connection dbConnection = null;
    Statement statement = null;

    String createTableSQL = "CREATE TABLE DBUSER("
            + "USER_ID NUMBER(5) NOT NULL, "
            + "USERNAME VARCHAR(20) NOT NULL, "
            + "CREATED_BY VARCHAR(20) NOT NULL, "
            + "CREATED_DATE DATE NOT NULL, " + "PRIMARY KEY
(USER_ID) "
            + ")";

    try {
        dbConnection = getDBConnection();
        statement = dbConnection.createStatement();

                // SQL so’rovni bajarish
        statement.execute(createTableSQL);

        System.out.println("Table \"dbuser\" is created!");

    } catch (SQLException e) {

        System.out.println(e.getMessage());

    } finally {

        if (statement != null) {

            statement.close();

        }

        if (dbConnection != null) {

            dbConnection.close();

        }

    }

}
```

main funksiyasi ichida createDbTable() metodini chaqirish:

```
    public static void main(String[] argv) {

        try {

            createDbUserTable();

        } catch (SQLException e) {

            System.out.println(e.getMessage());

        }

    }
```

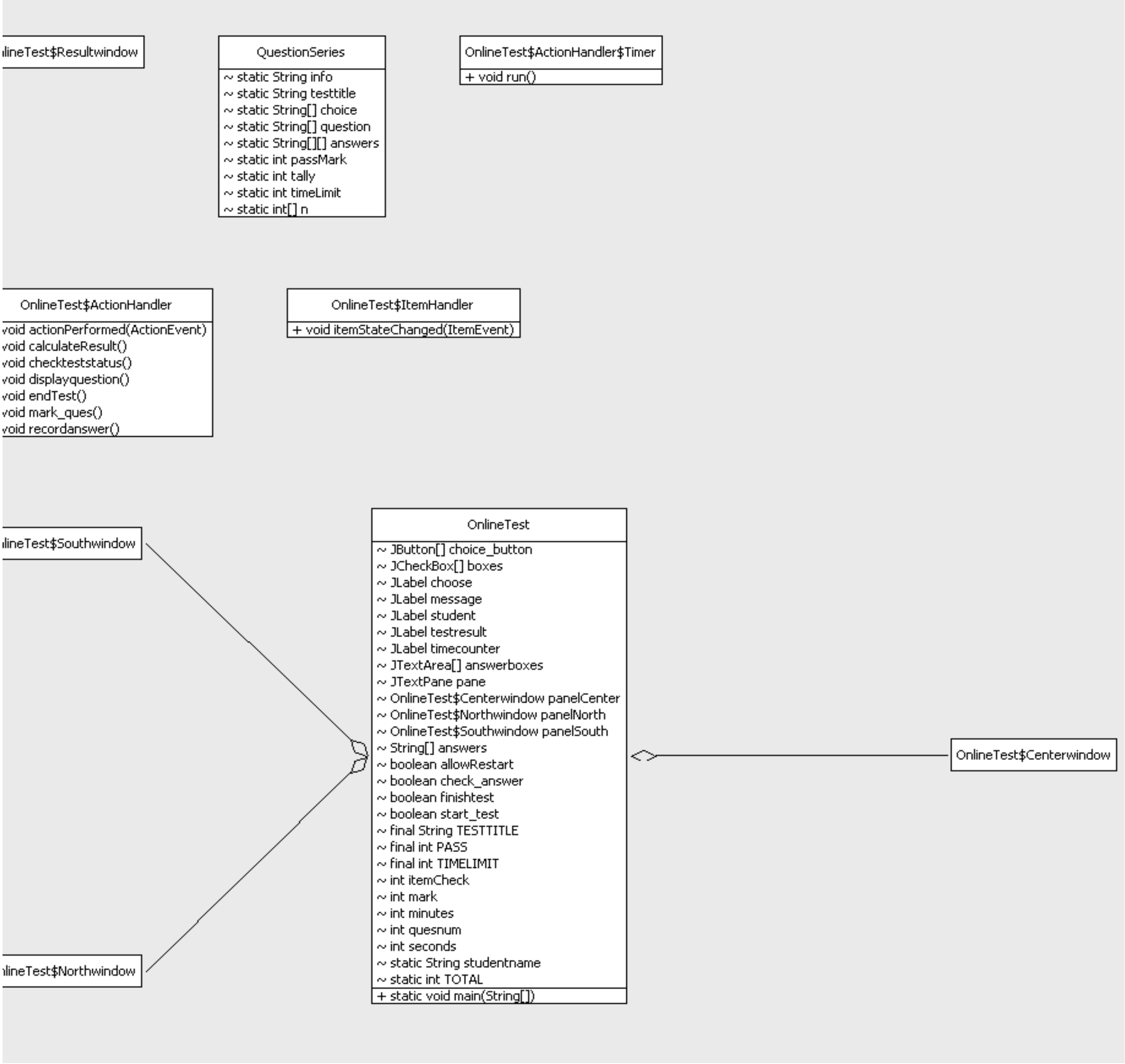
MBga ma’lumot kiritish:

```
String insertTableSQL = "INSERT INTO DBUSER"

            + "(USER_ID, USERNAME, CREATED_BY, CREATED_DATE) " + "VALUES"

            + "(1,'mkyong','system', " + "to_date('"
            + getCurrentTimeStamp() + "', 'yyyy/mm/dd hh24:mi:ss'))";

private static String getCurrentTimeStamp() { Date today = new Date(); return dateFormat.format(today.getTime()); }

statement.executeUpdate(insertTableSQL);
```

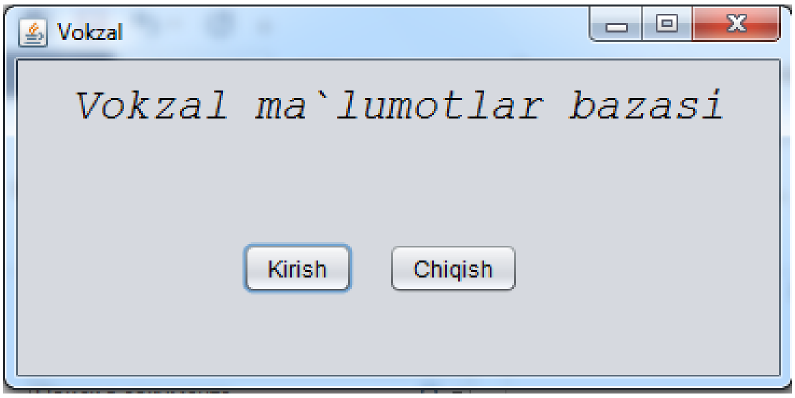
MB dan ma’lumotlarni o’qib olish:

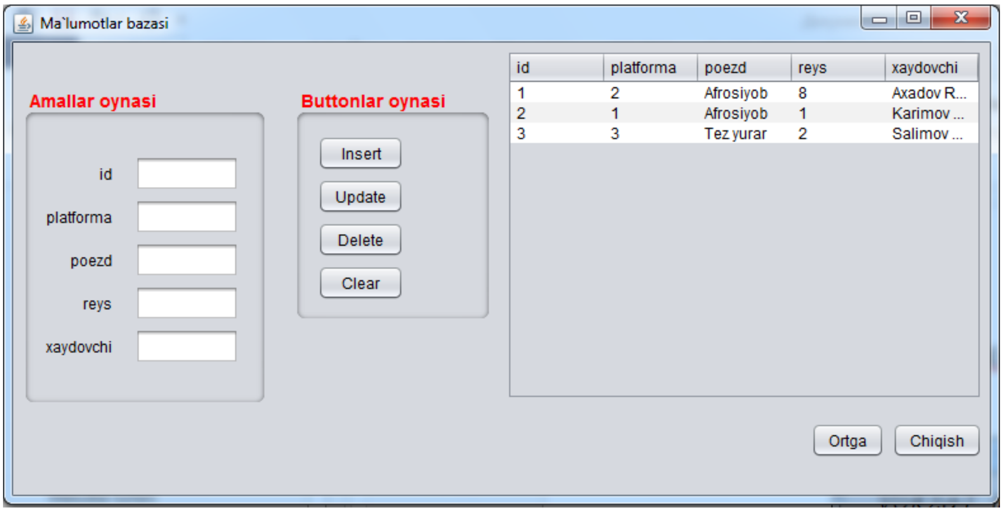
```
String selectTableSQL = "SELECT USER_ID, USERNAME from DBUSER";

try {

    dbConnection = getDBConnection();

    statement = dbConnection.createStatement();


    // Mbdan ma’lumotlarni tanlab olish

    ResultSet rs = statement.executeQuery(selectTableSQL);


    // Ma’lumot olingan bo’lsa while sikli ishga tushadi.

    while (rs.next()) {

        String userid = rs.getString("USER_ID");

        String username = rs.getString("USERNAME");


        System.out.println("userid : " + userid);

        System.out.println("username : " + username);

    }

} catch (SQLException e) {

    System.out.println(e.getMessage());

}
```

3.3 Loyihaning UML sxemasi va algoritmi.



3.4 Foydalanish yo’riqnomasi.





id	platforma	poezd	reys	xaydovchi
1	2	Afrosiyob	8	Axadov R...
2	1	Afrosiyob	1	Karimov ...
3	3	Tez yurar	2	Salimov ...

Bu oynalarda ma’lumot qo’shish, o’zgartirish, o’chirish va tozalash tugmalari qo’shilgan

III. Xulosa

Axborot texnologiyalari rivojlanib borayotgan hozirgi kunda tizim, texnologiya, barcha sohalarda keng qo`llanilib kelmoqda. Avtomatlashtiriligan tizim barcha soxalarni rivojlanishiga katta xissa qo`shadi. Shunday ekan barcha sohalarni elektron tizim sifatida yaxlit xolatga keltirilsa, kerakli soxani o`rganish, undan foydalanish ancha yengillashadi.

Hozirgi kunda mudofani ximoyalash sohasida keng qamrovli ishlar olinib borilyapti. Bu dasturiy tizim mijozlar ham ancha yengillik yaratadi. Xozirgi texnika va texnalogiyalar shuni talab etadiki mijoz har bir mussasada birinchi o’rinni egallaydi kim yaxshi xizmat ko’rsata olsa u tashkilot rivojlanadi hamda mijozlari ishonchini oqlaydi. Shuni yaqqol misoli sifatida armiya tizimini olsak elektron tarzda mijoz xohlagan talabgorga javob bera oladigan xizmat ko’rsatish xar qanday mijozni befarq qoldirmaydi. Bu esa ortiqcha sarf harajatlardan ozod qiladi. Shu bilan birga vaqtdan ham yutiladi. Bundan tashqari baza orqali xizmat turini tanlashga imkoniyat yetarli darajada bo`ladi

IV.Foydalanilgan adabiyotlar.

1. 2002-yil 30-maydagi «Kompyuterlashtirish va informatsion-kommunikatsion texnologiyalarni qo'llashni yanada rivojlantirish» haqidagi qaror.
2. Ivor Horton – Beginning Java.
3. Object oriented programming Java.
4. Internet saytlari: http:// google.com, http:// stackoverflow.com, http:// tutorialspoint.com, http:// oracle.com.
5. Gerbert Shildt – “Kak napisat programmu na Java”.
6. Fayn A.T – “Java dlya dedushek i malchikam”.
7. Майкл Томас, Пратик Пател, Алан Хадсон, Доналд Болл(мл.) **Секреты программирования для Internet на Java**.- Ventana Press, Ventana Communications Group, U.S.A.,1996, Издательство "ПитерПресс", 1997
8. Аарон И.Волш **Основы программирования на Java для World Wide Web**.- IDG Books Worldwide,Inc.,1996, Издательство "Диалектика",1996
9. Кен Арнольд, Джеймс Гослинг **Язык программирования Java**.- Addison-Wesley Longman,U.S.A.,1996, Издательство "Питер-Пресс", 1997
10. Нейл Бартлетт, Алекс Лесли, Стив Симкин **Программирование на Java. Путеводитель**.- The Coriolis Group,Inc.,1996, Издательство НИПФ "ДиаСофт Лтд.",1996

    Крис Джамса **Библиотека программиста Java**.- Jamsa Press, 1996, ООО "Попурри", 1996

V.Ilova

```
package vagzal;

import com.mysql.jdbc.Connection;
import java.sql.*;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import net.proteanit.sql.*;

public class info_vok extends javax.swing.JFrame {

     Connection conn=null;
    ResultSet rs=null;
    PreparedStatement ps=null;

    public info_vok() throws SQLException, ClassNotFoundException {
        initComponents();
        Class.forName("com.mysql.jdbc.Driver");
        conn=(Connection)
DriverManager.getConnection("jdbc:mysql://localhost/mysql","root","");

        Jtable();
    }

    private void Jtable() throws SQLException{

        String sql="SELECT * FROM vagzal.vagzal order by id";
         ps=conn.prepareStatement(sql);
         rs=ps.executeQuery();
         jtab.setModel(DbUtils.resultSetToTableModel(rs));
            }
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jScrollPane1 = new javax.swing.JScrollPane();
        jtab = new javax.swing.JTable();
        jButton1 = new javax.swing.JButton();
        jButton2 = new javax.swing.JButton();
        jPanel1 = new javax.swing.JPanel();
        jLabel1 = new javax.swing.JLabel();
        platforma = new javax.swing.JTextField();
        jLabel2 = new javax.swing.JLabel();
        jLabel3 = new javax.swing.JLabel();
        poezd = new javax.swing.JTextField();
```

```
jLabel4 = new javax.swing.JLabel();
reys = new javax.swing.JTextField();
xaydovchi = new javax.swing.JTextField();
jLabel5 = new javax.swing.JLabel();
id = new javax.swing.JTextField();
jPanel2 = new javax.swing.JPanel();
jButton3 = new javax.swing.JButton();
jButton6 = new javax.swing.JButton();
jButton4 = new javax.swing.JButton();
jButton5 = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setTitle("Ma`lumotlar bazasi");
setBackground(new java.awt.Color(-20561,true));

jtab.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null}
    },
    new String [] {
        "Title 1", "Title 2", "Title 3", "Title 4"
    }
));
jtab.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        jtabMouseClicked(evt);
    }
});
jScrollPane1.setViewportView(jtab);

jButton1.setText("Chiqish");
jButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton1ActionPerformed(evt);
    }
});
jButton2.setText("Ortga");
jButton2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton2ActionPerformed(evt);
    }
});
```

```
    jPanel1.setBorder(javax.swing.BorderFactory.createTitledBorder(null, "Amallar oynasi", 0,
0, new java.awt.Font("SansSerif", 1, 14), new java.awt.Color(-65536,true))); // NOI18N

    jLabel1.setText("platforma");

    jLabel2.setText("poezd");

    jLabel3.setText("reys");

    jLabel4.setText("xaydovchi");

    jLabel5.setText("id");

    javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
    jPanel1.setLayout(jPanel1Layout);
    jPanel1Layout.setHorizontalGroup(
      jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
      .addGroup(jPanel1Layout.createSequentialGroup()
        .addContainerGap()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING
)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING
)
            .addGroup(jPanel1Layout.createSequentialGroup()
              .addComponent(jLabel1)
              .addGap(18, 18, 18)
              .addComponent(platforma, javax.swing.GroupLayout.PREFERRED_SIZE, 82,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGroup(jPanel1Layout.createSequentialGroup()
              .addComponent(jLabel2)
              .addGap(18, 18, 18)
              .addComponent(poezd, javax.swing.GroupLayout.PREFERRED_SIZE, 82,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGroup(jPanel1Layout.createSequentialGroup()
              .addComponent(jLabel3)
              .addGap(18, 18, 18)
              .addComponent(reys, javax.swing.GroupLayout.PREFERRED_SIZE, 82,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGroup(jPanel1Layout.createSequentialGroup()
              .addComponent(jLabel4)
              .addGap(18, 18, 18)
              .addComponent(xaydovchi, javax.swing.GroupLayout.PREFERRED_SIZE,
82, javax.swing.GroupLayout.PREFERRED_SIZE)))
```

```
               .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup()
                  .addComponent(jLabel5)
                  .addGap(18, 18, 18)
                  .addComponent(id, javax.swing.GroupLayout.PREFERRED_SIZE, 82,
javax.swing.GroupLayout.PREFERRED_SIZE)))
            .addContainerGap(10, Short.MAX_VALUE))
      );
      jPanel1Layout.setVerticalGroup(
         jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
         .addGroup(jPanel1Layout.createSequentialGroup()
            .addGap(22, 22, 22)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELIN
E)
               .addComponent(id, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
               .addComponent(jLabel5))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELIN
E)
               .addComponent(jLabel1)
               .addComponent(platforma, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELIN
E)
               .addComponent(jLabel2)
               .addComponent(poezd, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELIN
E)
               .addComponent(jLabel3)
               .addComponent(reys, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELIN
E)
               .addComponent(jLabel4)
               .addComponent(xaydovchi, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
```

```
                .addContainerGap(22, Short.MAX_VALUE))
        );

        jPanel2.setBorder(javax.swing.BorderFactory.createTitledBorder(null, "Buttonlar oynasi",
0, 0, new java.awt.Font("SansSerif", 1, 14), new java.awt.Color(-65536,true))); // NOI18N

        jButton3.setText("Insert");
        jButton3.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton3ActionPerformed(evt);
            }
        });
        jButton6.setText("Clear");
        jButton6.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton6ActionPerformed(evt);
            }
        });
        jButton4.setText("Update");
        jButton4.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton4ActionPerformed(evt);
            }
        });

        jButton5.setText("Delete");
        jButton5.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton5ActionPerformed(evt);
            }
        });

        javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
        jPanel2.setLayout(jPanel2Layout);
        jPanel2Layout.setHorizontalGroup(
            jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel2Layout.createSequentialGroup()
                .addContainerGap()

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
false)
                    .addComponent(jButton4, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .addComponent(jButton3, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
```

```
                .addComponent(jButton5, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(jButton6, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
            .addContainerGap(57, Short.MAX_VALUE))
        );
        jPanel2Layout.setVerticalGroup(
            jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel2Layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jButton3)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jButton4)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jButton5)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jButton6)
                .addContainerGap())
        );

        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                    .addGroup(layout.createSequentialGroup()
                        .addComponent(jButton2)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent(jButton1))
                    .addGroup(layout.createSequentialGroup()
                        .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addGap(18, 18, 18)
                        .addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addGap(10, 10, 10)
                        .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE,
375, javax.swing.GroupLayout.PREFERRED_SIZE)))
                .addContainerGap())
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(layout.createSequentialGroup()
                        .addContainerGap()
                        .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE,
275, javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                            .addComponent(jButton2, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                            .addComponent(jButton1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)))
                    .addGroup(layout.createSequentialGroup()
                        .addGap(36, 36, 36)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                            .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))))
                .addContainerGap(29, Short.MAX_VALUE))
        );

        pack();
    }// </editor-fold>

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    dispose();
}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    new vokzal().setVisible(true);
    dispose();
}

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
    String sql="insert into vagzal.vagzal values(null,?,?,?,?)";
        try {
            ps=conn.prepareStatement(sql);

            ps.setInt(1,Integer.parseInt(platforma.getText()));
            ps.setString(2,poezd.getText());
            ps.setString(3,reys.getText());
            ps.setString(4,xaydovchi.getText());
            ps.executeUpdate();
                        JOptionPane.showMessageDialog(null,"Saved");

        } catch (SQLException ex) {
            Logger.getLogger(info_vok.class.getName()).log(Level.SEVERE, null, ex);
        }
        try {
            Jtable();

        } catch (SQLException ex) {
            Logger.getLogger(info_vok.class.getName()).log(Level.SEVERE, null, ex);
        }
}

private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {
    id.setText("");
    platforma.setText("");
    reys.setText("");
    xaydovchi.setText("");
    poezd.setText("");
}

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
    try{
           String value4=id.getText();
           String value0=platforma.getText();
           String value1=poezd.getText();
           String value2=reys.getText();
           String value3=xaydovchi.getText();



           String sql="update vagzal.vagzal set
id='"+value4+"',platforma='"+value0+"',poezd='"+value1+"',reys='"+value2+"',xaydovchi='"+val
ue3+"' where id='"+value4+"'";
           ps=conn.prepareStatement(sql);
           ps.execute();
           JOptionPane.showMessageDialog(null, "UPDATE");



       }catch(Exception e){
           JOptionPane.showMessageDialog(null, e);
       }
        try {
            Jtable();
        } catch (SQLException ex) {
            Logger.getLogger(info_vok.class.getName()).log(Level.SEVERE, null, ex);
        }
}

private void jtabMouseClicked(java.awt.event.MouseEvent evt) {
    try{

        int row=jtab.getSelectedRow();
        String table_click=(jtab.getModel().getValueAt(row,0).toString());
        String sql="SELECT * FROM vagzal.vagzal where id='"+table_click+"'";
        ps=conn.prepareStatement(sql);
        rs=ps.executeQuery();
        if(rs.next()){
            String add4=rs.getString("id");
            id.setText(add4);
            String add0=rs.getString("platforma");
            platforma.setText(add0);
            String add1=rs.getString("poezd");
            poezd.setText(add1);
            String add2=rs.getString("reys");
            reys.setText(add2);
            String add3=rs.getString("xaydovchi");
            xaydovchi.setText(add3);

        }

    }catch(Exception e){
        System.out.print(e);
    }
}

private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {
    try{

            String sql="delete from vagzal.vagzal where id=? ";
            ps=conn.prepareStatement(sql);

            ps.setString(1,id.getText());

            ps.execute();
            JOptionPane.showMessageDialog(null, "Delete");

        }catch(Exception e){
            JOptionPane.showMessageDialog(null, e);
        }
        try {
            Jtable();
        } catch (SQLException ex) {
            Logger.getLogger(info_vok.class.getName()).log(Level.SEVERE, null, ex);
        }
}

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
         * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(info_vok.class.getName()).log(java.util.logging.Level.SEVE
RE, null, ex);
        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(info_vok.class.getName()).log(java.util.logging.Level.SEVE
RE, null, ex);
        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(info_vok.class.getName()).log(java.util.logging.Level.SEVE
RE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(info_vok.class.getName()).log(java.util.logging.Level.SEVE
RE, null, ex);
        }
        //</editor-fold>

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {

            public void run() {
```

```
            try {
                new info_vok().setVisible(true);
            } catch (SQLException ex) {
                Logger.getLogger(info_vok.class.getName()).log(Level.SEVERE, null, ex);
            } catch (ClassNotFoundException ex) {
                Logger.getLogger(info_vok.class.getName()).log(Level.SEVERE, null, ex);
            }
        }
    });
}
// Variables declaration - do not modify
private javax.swing.JTextField id;
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JButton jButton3;
private javax.swing.JButton jButton4;
private javax.swing.JButton jButton5;
private javax.swing.JButton jButton6;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTable jtab;
private javax.swing.JTextField platforma;
private javax.swing.JTextField poezd;
private javax.swing.JTextField reys;
private javax.swing.JTextField xaydovchi;
// End of variables declaration
                                                                            }
```