

**O'ZBEKISTON RESPUBLIKASI ALOQA,
AXBOROTLASHTIRISH VA TELEKOMMUNIKATSIYA
TEXNOLOGIYALARI DAVLAT QO'MITASI
TOSHKENT AXBOROT TEXNOLOGIYALARI
UNIVERSITETI**

“Dasturiy injiniring” fakulteti

KURS ISHI

Mavzu: Restaurant tizimini avtomatlashtirish

Topshirdi: 215-12 guruh talabasi

I. I. Yoqubjonov

Qabul qildi: M. Z. Babamuhamedova

Toshkent 2015

Mundarija

I.Kirish	3
II.Nazariy qism	5
2.1 Masalaning qo`yilishi	5
2.2 Java dasturlash tili haqida umumiy tushuncha.....	5
2.3 Ma'lumotlar bazasi. MySQL haqoda tushuncha. Relyatsiyon ma'lumotlar bazasi.	124
2.4 Java Swing xaqida.	146
III. Amaliy qism	168
3.1 Ma'lumotlar bazasi tuzilmasi	168
3.2 Loyihaning UML sxemasi va algoritmi.	21
3.3 Foydalanish yo`riqnomasi.....	22
III. Xulosa	235
IV.Foydalanilgan adabiyotlar	246
V.Ilova	257

I.Kirish

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- dekabdagi «Axborotlashtirish to'g'risida»gi qonun, 2002- yil 30- maydagi «Kompyuterlashtirish va informatsion-kommunikatsion texnologiyalarni qo'llashni yanada rivojlantirish» haqidagi qaror, 2003- yil 11- dekabdagi «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.

II.Nazariy qism.

2.1 Masalaning qo`yilishi

Masalaning qo`yilishi: “Restaurant tizimini elektrolashtirish” mavzu bo`yicha dastur tuzish.

2.2 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

Quyidagilardan kelib chiqqan holda, biz Java dasturlash tili haqida to`xtalib o`tamiz.

Kompyuter dasturlari yoki dasturiy taminoti o'zi orqali o'lchanmalar majmuini ko'rsatadiki, ularni kompyuter aniq misol uchuni echish uchun qulaydir. Dasturchilar ularni maxsus tillar Basic, 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 Business 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 konsepsiyalarni bilish kerak bo'ladi:

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

* Avtanom dastur brovzer orqali ishlaydi.

* 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 firmasining Web 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 (ishonch yo'q kodni xavfsiz ishga tushirish).
- 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

- Dinamik ravishda imkoniyatini oshirish xam mumkin kerak bo'lsa.

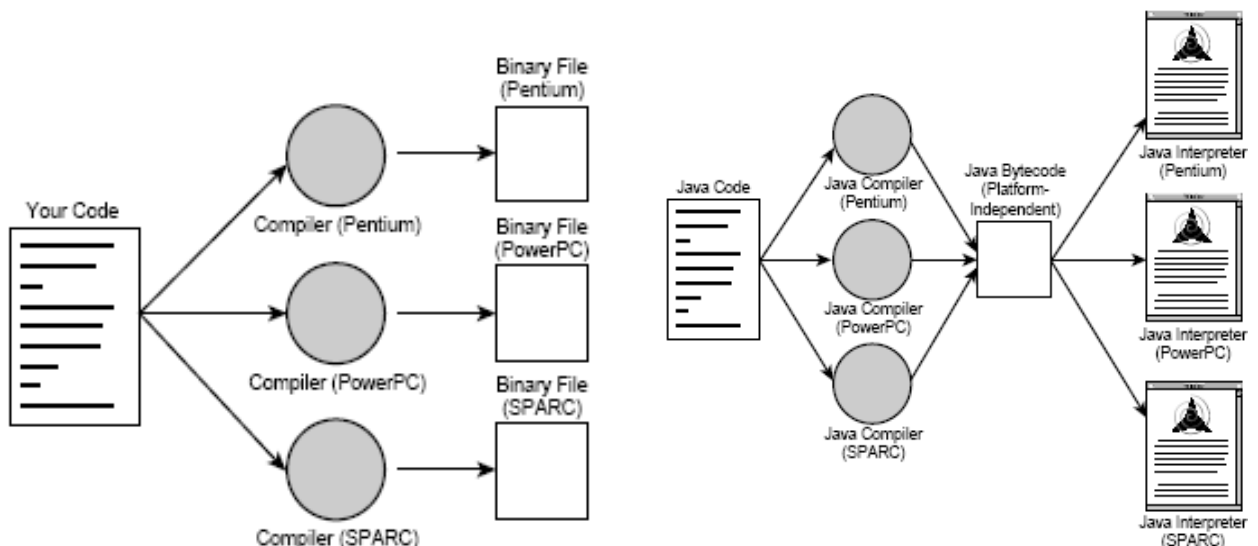
Quyidagi farqlar bilan Java C/C++ dan ajralib turadi

- 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 appletviewer bu tushunchalarni 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 funksiyasidan foydalanib, HTML fayli doirasida dasturning qiymat qabul qilishini o'rganish lozim bo'ladi. Bundan tashqari o'rganishni tamomlagach quyidagi kalit kontsepsiyalarni 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 operatorini 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 elementlarini 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 initsializatsiya qiladi:

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

Bu xolda kompilyator massiv uchun xotiradan 5 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 paketlar **JAR** faylda ham bo'lishi mumkin. JAVA paketlar 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'zolari gruppalashtiradi. Sinf appletingizga ob'ektning xamma atributlarini 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 katlit 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 – () qovus ichiga parametrlar vergul (,) bilan ajratilib yoziladi, parametr toifasi va nomi ko'rsatilib o'tilishi kerak, agar method hech qanday parametr olmasa () qovus ichi bo'sh qoldiriladi.
- Exception lar – hatoliklarni ushlab
- 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.

Constructor - bu class dan yangi obyekt yaratilganda obyektning boshlang'ich qiymatlarini berish.

- Constructor nomi class nomi bilan bir hil bo'ladi
- Constructorlarni 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-filename input-files*

Misol:

```
jar cf my_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
```

2.3 Ma'lumotlar bazasi. MySQL haqida ma'lumot. Relyatsiyon ma'lumotlar bazasi.

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 tex kompaniyasi tomonidan, ma'lumotlarni tez qayta ishlash uchun korxonada miqyosida ishlatishga yaratilgan. Keyinchalik ommalashib, saytlarning asosiy bazasi sifatida yoyildi.

So'rovlar **SQL** tili orqali amalga oshiriladi. Bu **MBBT relyatsion ma'lumotlar baza** hisoblanadi.

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 hususiyatlarga 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;

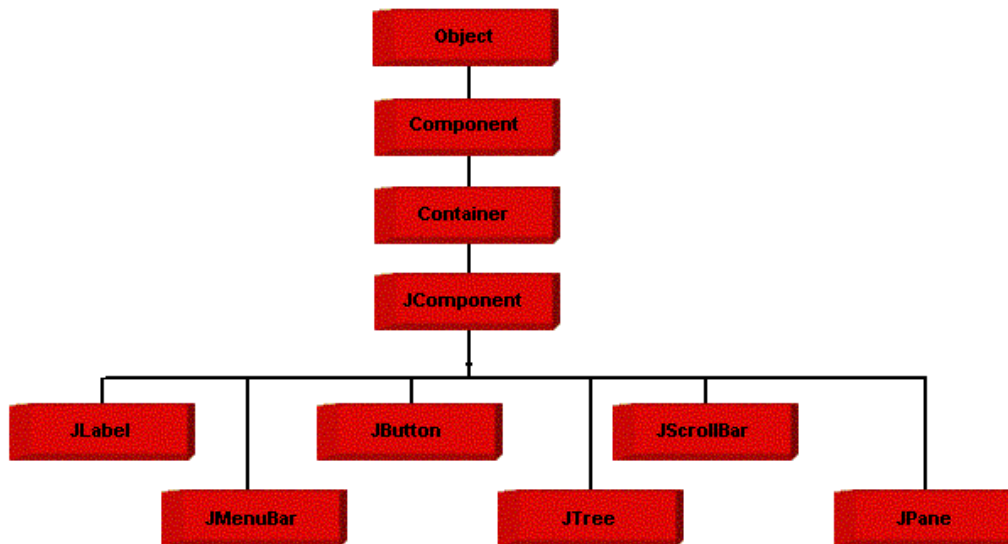
* Kortej;

2.4 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 funkcionallkin jihatidan Windows, Mac va Linux OS’ laridaka

Java Swing class ierarxiyasi



Swing API 18 ta public paketlardan iborat

javax.accessibility	javax.swing.plaf	javax.swing.text
javax.swing	javax.swing.plaf.basic	javax.swing.text.html
javax.swing.border	javax.swing.plaf.metal	javax.swing.text.html.parser
javax.swing.colorchooser	javax.swing.plaf.multi	javax.swing.text.rtf
javax.swing.event	javax.swing.plaf.synth	javax.swing.tree
javax.swing.filechooser	javax.swing.table	javax.swing.undo

SWING Layots

AWT va Swing klaslarining layout manager’lari:

BorderLayout

BoxLayout

CardLayout

FlowLayout

GridBagLayout

GridLayout

GroupLayout

SpringLayout

Java/SWING BorderLayout

BorderLayout

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

III. Amaliy qism.

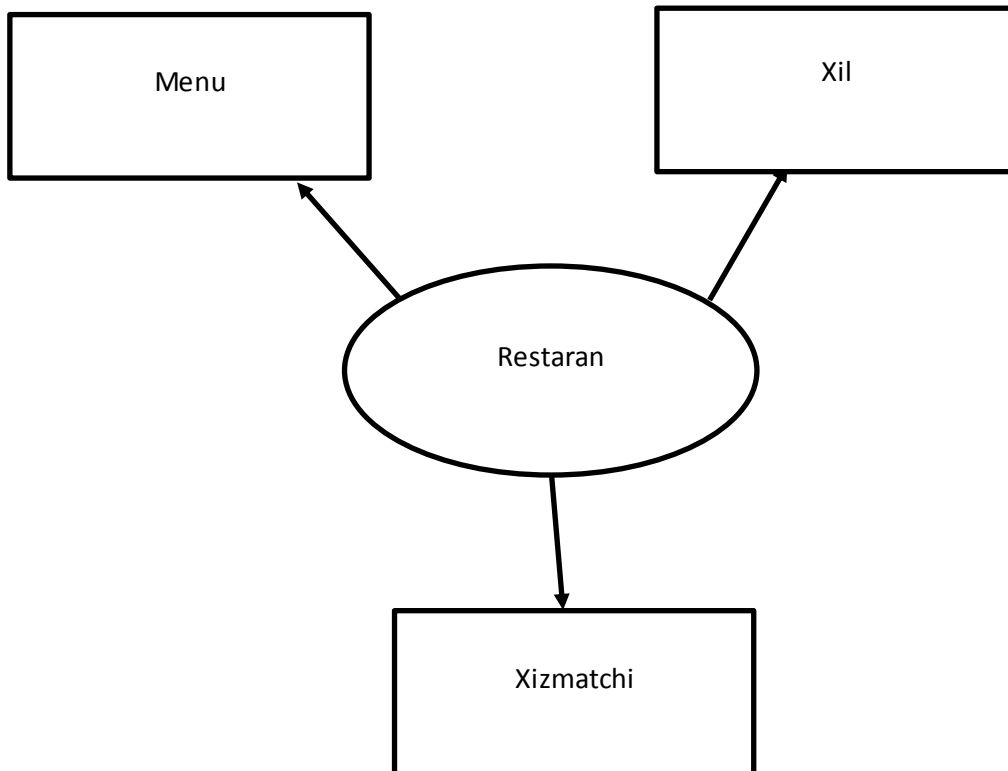
3.1 Ma'lumotlar bazasi tuzilmasi.

Turli xil dasturlar yaratishda bazalar bilan ishlash juda qo'l keladi. Bazalar dastur kodini kamroq, dasturdan ma'lumotni oson qidirib topish uchun kerak bo'ladi. Dasturlarni yaratishda asosan MySQL, SQL, ORACLE, Access kabi ma'lumotlar bazalaridan foydalaniladi. Bular orasida MySQL ma'lumotlar bazasi o'zining soddaligi, ishlashga qulayligi va foydalanuvchilarining ko'pligi bilan ajralib turadi. Shu sababli, men ham dasturimda MySQL ma'lumotlar bazasidan foydalanishga harakat qildim.

Ma'lumotlar bazasi o'ta tez rivojlangan hamda MySQL va NoSQL o'zini ko'rsatgan soha, Internet uchun dasturlar yaratishdir. Internet uchun murakkab va ishonchli dasturlarga ehtiyoj oshgan sari ma'lumotlar bazasiga ehtiyoj ham oshib bormoqda. Server ma'lumotlar bazasi Internetda ko'p funksiyalarni qo'llashi mumkin. Har qanday dasturlar ma'lumotlar bazasi tomonidan boshqarilishi mumkin. Dasturimizda restoran deb nomlangan ma'lumotlar bazasi tashkil etilgan. Uning ichida uchta jadval yaratilgan bo'lib, menu, xil va xizmatchi deb nomlangan. Menu jadvalim 4 ta ustundan iborat bo'lib, ular : Id (INT), nom(varchar), tur(varchar), narx(int). Bunda biz bazaga ta'om nomlarini kiritishimiz mumkin.

Ikkinchi jadvalim esa xil – ta'omlar guruhlab chiqilgan.

Uchinchi jadvalim bu – xizmatchi jadvali bo'lib, unda ishchilar haqida ma'lumotlar saqlanadi.



localhost > restaran

Структура SQL Поиск Запрос по шаблону Экспорт Импорт Операции Привилегии Процедуры Ещё

Таблица	Действие	Строки	Тип	Сравнение	Размер	Фрагментировано
menu	Обзор Структура Поиск Вставить Очистить Удалить	6	MyISAM	cp1251_general_ci	2.2 КБ	-
xil	Обзор Структура Поиск Вставить Очистить Удалить	6	MyISAM	cp1251_general_ci	2.1 КБ	-
xizmatchi	Обзор Структура Поиск Вставить Очистить Удалить	4	MyISAM	cp1251_general_ci	2.1 КБ	-
3 таблицы	Всего	16	InnoDB	utf8_general_ci	6.4 КБ	0 Байт

↑ Отметить все С отмеченными:

Версия для печати Словарь данных

Создать таблицу

Имя: Количество столбцов:

OK

```
SELECT *
FROM `menu`
LIMIT 0 , 30
```

Профилирование [Быстрая правка] [Изменить]

Показать : Начальная строка: 0 Количество строк: 30 Заголовки каждые 100 строк

Сортировать по индексу: Нет

+ Параметры

	id	nom	tur	narx
<input type="checkbox"/> Изменить <input type="checkbox"/> Копировать <input type="checkbox"/> Удалить	1 osh	quyuq ovqatlar		10000
<input type="checkbox"/> Изменить <input type="checkbox"/> Копировать <input type="checkbox"/> Удалить	2 Chuchvara	suyuq ovqat		6000
<input type="checkbox"/> Изменить <input type="checkbox"/> Копировать <input type="checkbox"/> Удалить	3 Do`lma	Quyuq ovqat		7000
<input type="checkbox"/> Изменить <input type="checkbox"/> Копировать <input type="checkbox"/> Удалить	4 Kola	Ichimlik		5000
<input type="checkbox"/> Изменить <input type="checkbox"/> Копировать <input type="checkbox"/> Удалить	5 Svejy	salat		1000
<input type="checkbox"/> Изменить <input type="checkbox"/> Копировать <input type="checkbox"/> Удалить	6 Pirojniy	Shirinlik		2000

↑ Отметить все С отмеченными: Изменить Удалить Экспорт

Показать : Начальная строка: 0 Количество строк: 30 Заголовки каждые 100 строк

Использование результатов запроса

localhost » restoran » xil

Обзор Структура SQL Поиск Вставить Экспорт Импорт Операции Триггеры

Отображает строки 0 - 5 (6 всего, Запрос занял 0.0004 сек.)

```
SELECT *
FROM `xil`
LIMIT 0 , 30
```

Профилирование [Быстрая правка] [Изменить] [Анализ SQL запроса]

Показать : Начальная строка: 0 Количество строк: 30 Заголовки каждые 100 строк

Сортировать по индексу: Нет

+ Параметры

	id	xil
<input type="checkbox"/> Изменить <input type="checkbox"/> Копировать <input type="checkbox"/> Удалить	1 quyuq	
<input type="checkbox"/> Изменить <input type="checkbox"/> Копировать <input type="checkbox"/> Удалить	2 suyuq	
<input type="checkbox"/> Изменить <input type="checkbox"/> Копировать <input type="checkbox"/> Удалить	3 salat	
<input type="checkbox"/> Изменить <input type="checkbox"/> Копировать <input type="checkbox"/> Удалить	4 disert	
<input type="checkbox"/> Изменить <input type="checkbox"/> Копировать <input type="checkbox"/> Удалить	5 ichimlik	
<input type="checkbox"/> Изменить <input type="checkbox"/> Копировать <input type="checkbox"/> Удалить	6	

↑ Отметить все С отмеченными: Изменить Удалить Экспорт

```
SELECT *
FROM `xizmatchi`
LIMIT 0 , 30
```

Профилирование [Быстрая правка] [Изменить] [Анализ SQL запроса]

Показать : Начальная строка: 0 Количество строк: 30 Заголовки каждые 100 строк

Сортировать по индексу: Нет

+ Параметры

	id	FIO	lavozim
<input type="checkbox"/> Изменить <input type="checkbox"/> Копировать <input type="checkbox"/> Удалить	1 Kanimov Dilshod	Dorektor	
<input type="checkbox"/> Изменить <input type="checkbox"/> Копировать <input type="checkbox"/> Удалить	2 Salimov Baxodir	Dirktor o`rinbosari	
<input type="checkbox"/> Изменить <input type="checkbox"/> Копировать <input type="checkbox"/> Удалить	3 Frqatov Sobir	Hisobchi	
<input type="checkbox"/> Изменить <input type="checkbox"/> Копировать <input type="checkbox"/> Удалить	4 Ravshanov Valijon	afitsant	

↑ Отметить все С отмеченными: Изменить Удалить Экспорт

Показать : Начальная строка: 0 Количество строк: 30 Заголовки каждые 100 строк

Использование результатов запроса

3.2 Loyihaning UML sxemasi va algoritmi.

restoran
<ul style="list-style-type: none"> - JButton jButton1 - JButton jButton2 - JButton jButton3 - JButton jButton4 - JButton jButton5 - JLabel jLabel1
<ul style="list-style-type: none"> + static void main(String[]) - void initComponents() - void jButton1ActionPerformed(ActionEvent) - void jButton2ActionPerformed(ActionEvent) - void jButton3ActionPerformed(ActionEvent) - void jButton5ActionPerformed(ActionEvent)

xizmat
<ul style="list-style-type: none"> - JButton jButton1 - JButton jButton2 - JLabel jLabel1 - JLabel jLabel2 - JPanel jPanel1 - JScrollPane jScrollPane1 - JTable jTable1 - JTextField jTextField1 ~ Connection conn ~ PreparedStatement ps ~ ResultSet rs
<ul style="list-style-type: none"> + static void main(String[]) - void jTable() - void initComponents() - void jButton1ActionPerformed(ActionEvent) - void jButton2ActionPerformed(ActionEvent) - void jTextField1KeyReleased(KeyEvent)

menu
<ul style="list-style-type: none"> - JButton jButton1 - JButton jButton2 - JButton jButton3 - JButton jButton4 - JButton jButton5 - JButton jButton6 - JButton jButton7 - JLabel jLabel1 - JLabel jLabel2 - JLabel jLabel3 - JScrollPane jScrollPane1 - JTable jTable1 ~ Connection conn ~ PreparedStatement ps ~ ResultSet rs
<ul style="list-style-type: none"> + static void main(String[]) - void jTable() - void initComponents() - void jButton1ActionPerformed(ActionEvent) - void jButton2ActionPerformed(ActionEvent) - void jButton3ActionPerformed(ActionEvent) - void jButton4ActionPerformed(ActionEvent) - void jButton5ActionPerformed(ActionEvent) - void jButton6ActionPerformed(ActionEvent) - void jButton7ActionPerformed(ActionEvent)

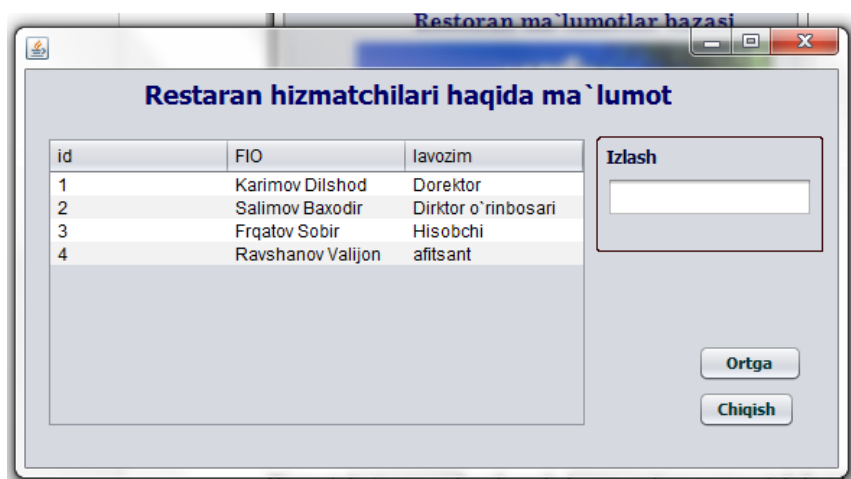
buyurtma
<ul style="list-style-type: none"> - JButton jButton1 - JButton jButton6 - JButton jButton7 - JComboBox jComboBox1 - JComboBox jComboBox2 - JLabel jLabel1 - JLabel jLabel2 - JLabel jLabel3 - JLabel jLabel4 - JLabel jLabel5 - JPanel jPanel1 - JTextField jTextField1 - JTextField jTextField2 ~ static Connection conn ~ static PreparedStatement ps ~ static ResultSet rs
<ul style="list-style-type: none"> + static void main(String[]) - void combo() - void combo1(String) - void initComponents() - void jButton1ActionPerformed(ActionEvent) - void jButton6ActionPerformed(ActionEvent) - void jButton7ActionPerformed(ActionEvent) - void jComboBox1ActionPerformed(ActionEvent) - void jComboBox2ActionPerformed(ActionEvent)

3.3 Foydalanish yo'riqnomasi.

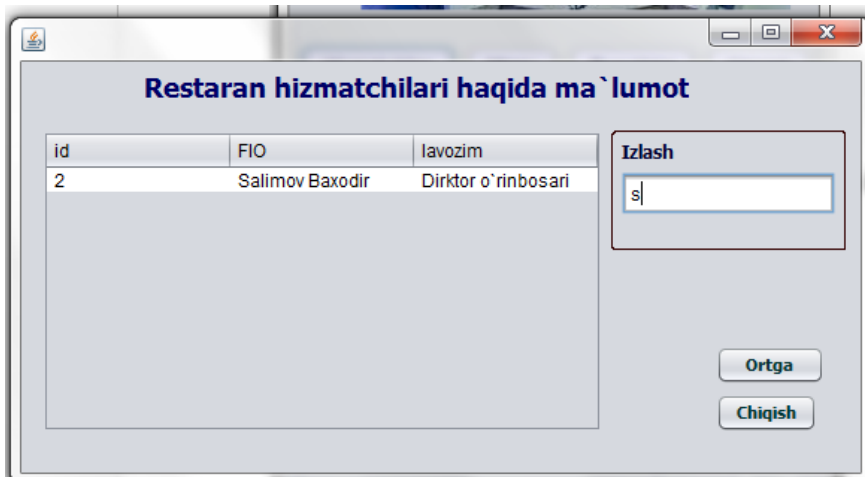
Dastur ishga tushganda quyidagi oyna ochiladi:



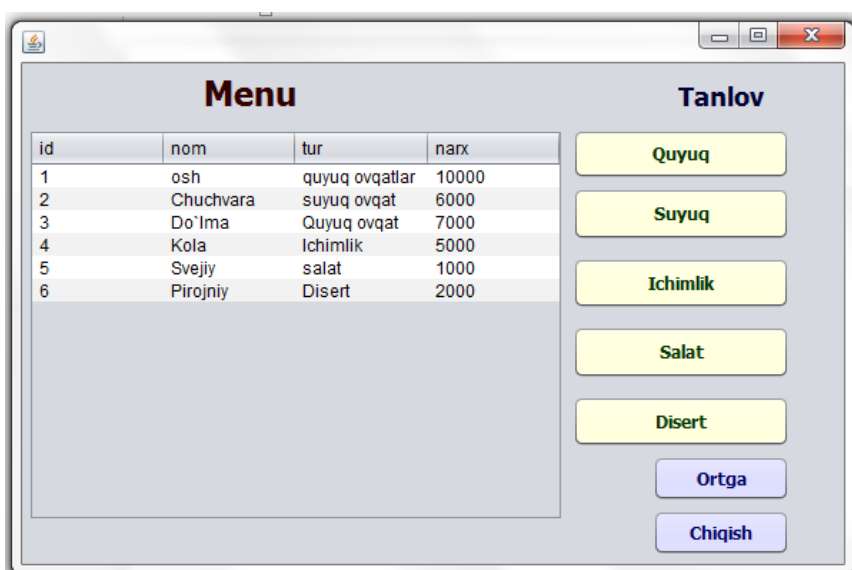
Xizmatchi tugmasi bosilganda bizga restaran xizmatchilari haqida ma'lumot chiqadi:



Izlash qismida kerakli ishchilarni qidirish mumkin.



Menu tugmasini bosilganda menuga kiritilgan taomlarni korishingiz mumkin



Tanlov qismida taomlarni turlarga ajratilgan



Buyurtma qismida taomlarga buyurtma berishingiz mumkin:

Buyurtma bo`limi

Quyuc ovqat ▾ Do`lma ▾

7000 Narx

3 Miqdorni kiriting

OK

Umumiy: 21000

Ortga

Chiqish

III. Xulosa

Hozirgi kunda O`zbekistonda Informatcion Texnologiyalar va Axborot Texnologiyalarini rivojlantirish yo`lida bir talay ishlar amalga oshirilmoqda. Buni birgina bu sohada ishlab chiqilgan qaror, qonun va farmonlar misolida ham ko`rishimiz mumkin. Bizga fan sifatida o`tilayotgan “ Tizimli dasturiy ta`minot ” fani ham so`zimizning yaqqol isboti bo`la oladi .

Hozirda biz hayotimizni komputersiz tasavvur qila olmaganimizdek, dunyodagi barcha katta – yu kichik ishlarning hammasi avtomatlashtirilgan tizimga asta – sekinlik bilan o`tib bormoqda. Albatta, bunday sistemaga o`z – o`zidan o`tib bo`lmaydi. Bunda bizga bir nechta dasturlash tillari yordam beradi. Shunday ekan men ham kurs ishimni dasturini tuzish mobaynida dasturlash tillaridan biri Java dasturlash tilidan foydalandim.

Bu dasturni tuzishda Java (dasturlash tili), MySQL (Ma`lumotlar ba`zasi) larni o`rgandim. O`ylaymanki bu tuzgan dasturim kelajakda foydalanuvchilar uchun ko`p qulayliklar yaratadi.

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://google.com), [http:// stackoverflow.com](http://stackoverflow.com), [http:// tutorialspoint.com](http://tutorialspoint.com), [http:// oracle.com](http://oracle.com).

V.Ilova

```
package restoran;

import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;

public class restoran extends javax.swing.JFrame {

    public restoran() {
        initComponents();
        this.setLocationRelativeTo(null);
    }

    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN: initComponents
    private void initComponents() {

        jButton4 = new javax.swing.JButton();
        jLabel1 = new javax.swing.JLabel();
        jButton1 = new javax.swing.JButton();
        jButton2 = new javax.swing.JButton();
        jButton3 = new javax.swing.JButton();
        jButton5 = new javax.swing.JButton();
        jLabel2 = new javax.swing.JLabel();

        jButton4.setText("jButton4");

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

        jLabel1.setFont(new java.awt.Font("Century Schoolbook", 1, 14)); //
NOI18N
        jLabel1.setForeground(new java.awt.Color(0, 0, 102));
        jLabel1.setText("Restoran ma`lumotlar bazasi");

        jButton1.setBackground(new java.awt.Color(204, 204, 255));
        jButton1.setFont(new java.awt.Font("Segoe UI Semilight", 1, 11)); //
NOI18N
        jButton1.setForeground(new java.awt.Color(0, 0, 51));
        jButton1.setText("Hizmatchilar");
```

```
jButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton1ActionPerformed(evt);
    }
});
```

```
jButton2.setBackground(new java.awt.Color(204, 204, 255));
jButton2.setFont(new java.awt.Font("Segoe UI Semilight", 1, 11)); //
```

NOI18N

```
jButton2.setForeground(new java.awt.Color(0, 0, 51));
jButton2.setText("Menu");
jButton2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton2ActionPerformed(evt);
    }
});
```

```
jButton3.setBackground(new java.awt.Color(204, 204, 255));
jButton3.setFont(new java.awt.Font("Segoe UI Semilight", 1, 11)); //
```

NOI18N

```
jButton3.setForeground(new java.awt.Color(0, 0, 51));
jButton3.setText("Buyurtma");
jButton3.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton3ActionPerformed(evt);
    }
});
```

```
jButton5.setBackground(new java.awt.Color(204, 204, 255));
jButton5.setFont(new java.awt.Font("Segoe UI Semilight", 1, 11)); //
```

NOI18N

```
jButton5.setForeground(new java.awt.Color(0, 0, 51));
jButton5.setText("Chiqish");
jButton5.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton5ActionPerformed(evt);
    }
});
```

```
jLabel2.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/restoran/restuant.jpg"))); //
```

NOI18N

```
javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
```

```

        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGroup(layout.createSequentialGroup()
                .addGroup(layout.createSequentialGroup()
                    .addGap(85, 85, 85)
                    .addComponent(jLabel1))
                .addGroup(layout.createSequentialGroup()
                    .addContainerGap()
                    .addComponent(jButton1))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(jButton2))
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jButton3))
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addComponent(jButton5))
                .addGroup(layout.createSequentialGroup()
                    .addGroup(layout.createSequentialGroup()
                        .addGap(46, 46, 46)
                        .addComponent(jLabel2)))
                    .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
            );
        layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGroup(layout.createSequentialGroup()
                .addGap(19, 19, 19)
                .addComponent(jLabel1))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(jLabel2,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addGap(23, 23, 23))
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(jButton1)

```

```

        .addComponent(jButton2)
        .addComponent(jButton3)
        .addComponent(jButton5))
        .addGap(21, 21, 21))
    );

    pack();
} // </editor-fold> // GEN-END: initComponents

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)
{ // GEN-FIRST:event_jButton1ActionPerformed
    try {
        new xizmat().setVisible(true);
    } catch (ClassNotFoundException ex) {
        Logger.getLogger(restoran.class.getName()).log(Level.SEVERE, null,
ex);
    } catch (SQLException ex) {
        Logger.getLogger(restoran.class.getName()).log(Level.SEVERE, null,
ex);
    }
    dispose();
} // GEN-LAST:event_jButton1ActionPerformed

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt)
{ // GEN-FIRST:event_jButton2ActionPerformed
    try {
        new menu().setVisible(true);
    } catch (ClassNotFoundException ex) {
        Logger.getLogger(restoran.class.getName()).log(Level.SEVERE, null,
ex);
    } catch (SQLException ex) {
        Logger.getLogger(restoran.class.getName()).log(Level.SEVERE, null,
ex);
    }
    dispose();
} // GEN-LAST:event_jButton2ActionPerformed

private void jButton5ActionPerformed(java.awt.event.ActionEvent evt)
{ // GEN-FIRST:event_jButton5ActionPerformed
    dispose();
} // GEN-LAST:event_jButton5ActionPerformed

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt)
{ // GEN-FIRST:event_jButton3ActionPerformed
    try {

```

```

        new buyurtma().setVisible(true);
    } catch (ClassNotFoundException ex) {
        Logger.getLogger(restoran.class.getName()).log(Level.SEVERE, null,
ex);
    } catch (SQLException ex) {
        Logger.getLogger(restoran.class.getName()).log(Level.SEVERE, null,
ex);
    }
    dispose();
}
//GEN-LAST:event_jButton3ActionPerformed

```

```

public static void main(String args[]) {

    //<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(restoran.class.getName()).log(java.util.logging.
Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {

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

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

        java.util.logging.Logger.getLogger(restoran.class.getName()).log(java.util.logging.
Level.SEVERE, null, ex);
    }
}

```

```

    }
    //</editor-fold>

    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new restoran().setVisible(true);
        }
    });
}
// Variables declaration - do not modify//GEN-BEGIN:variables
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.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
// End of variables declaration//GEN-END:variables
}

```