

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

Dasturiy injineriing fakulteti

OYDT fanidan bajargan

KURS ISHI XISOBOTI

Bajardi: 215-12 guruh talabasi

Alimov Ikromxo'ja

Tekshirdi: Babamuhammedova M

Toshkent–2015

Mundarija

KIRISH.....	- 3 -
II. NAZARIY QISM.....	- 4 -
2.1. Java dasturlash tili haqida.....	- 4 -
2.2. Oracle malumotlar bazasi	- 11 -
2.3. Java swing	- 13 -
III. AMALIY QISM.....	- 15 -
3.1. Ma'lumotlar bazasi tuzilmasi.....	- 15 -
3.2. UML sxemasi	- 16 -
3.3. Dasturdan foydalanish	- 18 -
XULOSA.....	- 22 -
Foydalanilgan adabiyotlar	- 23 -
ILOVA	- 24 -

KIRISH

Avtomatlashtirilgan axborot tizimlar hozirgi kunga kelib, inson faoliyatining turli jabxalarida, xususan, ilmiy-tadqiqot ishlarini, xalq xo`jaligi tarmoqlarini boshqarishda, biznes va boshqa yo`nalishlarda ham rivojlanmoqda. Yaratilgan axborot tizimlari foydalanuvchilar vaqtini tejash va foydalanuvchi ishonchini yanada mustahkamlash va zamonaviy xizmat turlaridan unumli foydalanishga qaratilgan.

Obyektga yo`naltirilgan dasturlash tili fanidan kurs ishi tayyorlash mobaynida men Java dasturlash tili va Oracle ma`lumotlar bazasini Java dasturiga bog`lab ishlash bo`yicha tajribalarimni yanada oshirib Java muhiti bilan chuqurroq tanishaman. Ko`pgina yangi komponentalar bilan tanishib, ular ustida ishlash ko`nikmasini ortirishga harakat qilaman. Kurs ishimni bajarib, Oracle ma`lumotlar bazasini boshqarish tizimi va Java dasturlash tili bilan ishlashni o`rganaman.

Java dasturlash tilida dastur yaratishda ishlatiladigan operatorlar, java swing bilan ishlashni va undan dastur tuzishda foydalanishni, klaslar bilan ishlashni, obyekt yaratib, obyektlar ustida ishlash kabilarni o`rganib ulardan dastur tuzishda foydalanishni o`rganaman.

Men kurs ishimda "Kutubxona ishini avtomatlashtirish" mavzusini tanladim. Men bu kurs ishini bajarish mobaynida obyektga yo`naltirilgan java dasturlash tilini o`rganish, javada foydalanuvchi interfeyslarini(Java Swing) yaratishni o`rganish, malumotlar bazasi bilan ishlashni o`rganish, kutubxona ishini avtomatlashtirishni maqsad qilib oldim.

Agarda kutubxona ishini avtomatlashtirsak bunda xodimlar ishi osonlashadi va kerakli kitoblar bor yo`qligi, ular soni nechtaligi yoki qaysi kitobxondaligi, qachon qaytarilishi lozimligi har biri kompyuterga kiritilib borilsa shunda kutubxona xodimlari ishi yengillashadi. Java dasturlash tilida tuzadigan dasturim esa ma`lumotlardan foydalanishda va ularni qayta ishlashda vizual formalar orqali foydalanuvchiga ko`maklashadi. Bu esa albatta vaqtni tejashga va ish unumdorligini ortishiga yordam beradi.

II. NAZARIY QISM

2.1. Java dasturlash tili haqida

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

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

Java Obyektga Yo'naltirilgan Dasturlash tili va u C++ ga ancha o'xshash. Eng ko'p yo'l qo'yildigan xatolarga sabab bo'luvchi qismlari olib tashlanib, Java dasturlash tili ancha soddalashtirildi.

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

JAVA imkoniyatlari:

- WORA - Write Once, Run Anywhere (portable). Platforma tanlamaydi;
 - havfsizlik (ishonch yuq kodni havfsiz ishga tushirish);
 - hotirani havfsiz boshqarish (avtomat ravishda musorlarni yig'adi);
 - tarmoq uchun dasturlar yozish ;
 - 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.

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(Windows NT, Macintosh, Linux va Solaris). Shu bilan birga JavaSE boshqa Java turlarining asosi hisoblanadi.

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

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

Javada kompilyator aytib o'tganimizdek biz yozgan kodni bayt-kodga o'giradi, odatda kompilatsiyadan o'tgan klasslar *.class qisqartirmasi bilan tugaydi va kompilatsiyadan o'tgan klassni Java Virtual Machine(JVM) ga yuklanadi va bayt-kodli fayllarni interpretatsiya qiladi, ya'ni mashina tiliga o'giradi va shu bilan birga undagi kodni imkoni boricha optimallashtiradi.

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

Java SE Downloads



Java Platform (JDK) 8u20



JDK 8u20 & NetBeans 8.0.1

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

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

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

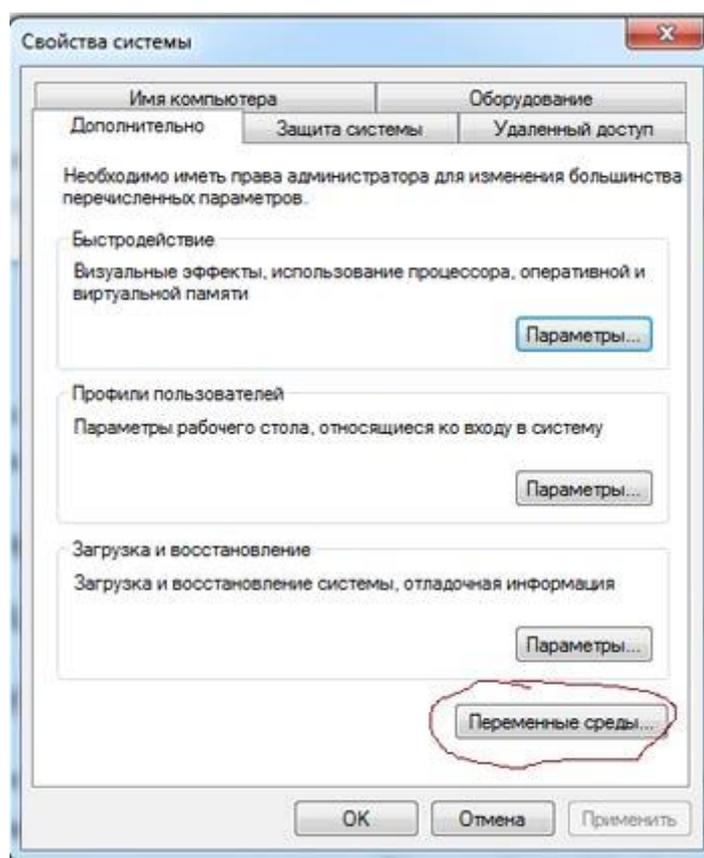
Odatda JDK "C:\Program files\Java" adresiga o'ratiladi

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

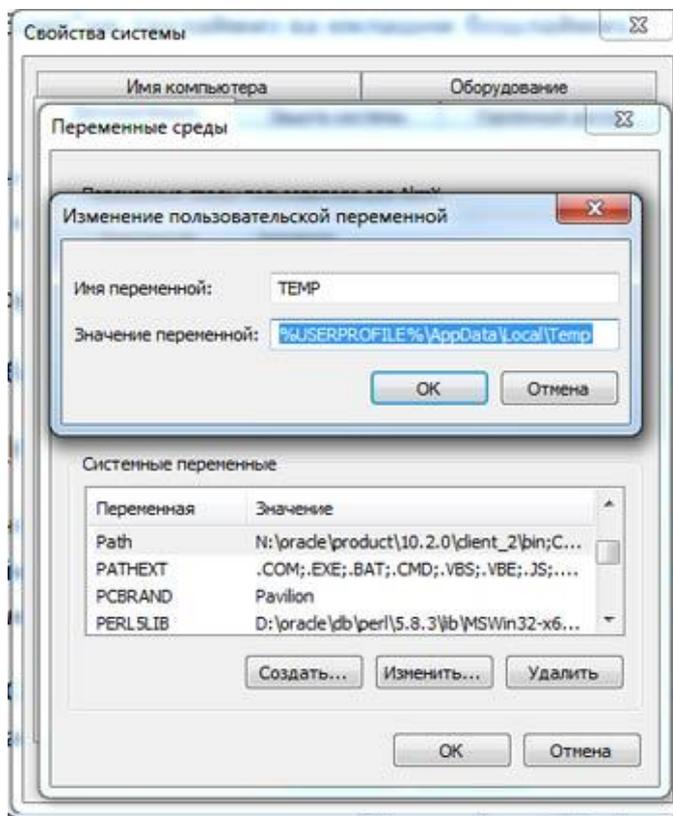
C:\Program files\jdk*\bin\

Bu adresda JDKning barcha bajariluvchi fayllari joylashgan. Biz ularni, operatsion tizimga ham ma'lum qilib qo'yishimiz lozim. Operatsion tizim bularni bilib olsa, bemalol "Командная строка" orqali ham ishlatishimiz mumkin bo'ladi. Buni Windows 7 misolida ko'rsataman.

Мой компьютер->Свойства, chap tomonda "Дополнительные параметры системы" bo'limiga kiramiz va quyidagi oynani ko'ramiz.



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



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

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

Navbat, Java dasturlash tilining asosi va uning strukturasi keldi. Barcha narsaning asosini bilib olsangiz, har qanday murakkab holni ham tushunish mumkin bo‘ladi. Nafaqat dasturlash, balki istalgan yo‘nalish, soha yoki biror bir tushunchani mukammal bilish uchun, dastlab uning strukturasi(nimalardan tashkil topganini)bilish, o‘rganishning asosiy va to‘g‘ri harakati bo‘ladi

Java dasturlash tilida dastlabki dasturimizni yaratib, undagi kodni birma-bir taxlil qilamiz:

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

Shu misol asosida, Java tili haqida, dastlabki bilimlarni egallaymiz

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

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

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

- klass nomi harfdan boshlanishi lozim;
- klass nomida faqat raqam va harflar ishtirok etishi mumkin
- uzunligi siz hohlaganingizcha bo‘lishi mumkin(kichikroq va tushunarli bo‘gani ma'qul);
- klass nomi sifatida maxsus zaxira so‘zlari ishtirok etmasligi lozim;

- odatda klass nomi katta harfdan iborat bo‘ladi

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

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

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

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

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

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

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

2.2. Oracle ma'lumotlar bazasi

Oracle (Ma'lumotlar Bazasini Boshqarish Tizimi) – katta hajmdagi ma'lumotlarni boshqarish, ulardan kerakli ma'lumotlarni so'rov orqali istalgan ko'rinishda chiqarib olish, ma'lumotlarning zahira nusxalarini olish, katta hajmdagi ma'lumotlarni siqish, qulay interfeysda baza ustidan nazorat o'rnatish, ma'lumotlar asosida hisobotlar hosil qilish va bulardan boshqa ma'lumotlar ustida juda katta ko'lamdagi ishlarni amalga oshiradigan dasturiy kompleksdir. Oracle dasturini server va klient ko'rinishida o'rnatish mumkin va berilgan ruxsat doirasida ishlatish mumkin. Bu dastur asosan katta hajmdagi ma'lumotlar bilan ishlaganligi uchun, asosan yirik korxonalarda ishlatiladi. Bu dasturning asosini ma'lumotlar bazasi tushunchasi tashkil etadi.

Ma'lumotlar bazasi – bu serverda joylashgan bir yoki bir necha fayllardan tashkil topgan, shu serverga yig'iladigan va so'rovlar orqali nazorat qilib boriladigan ma'lumotlardir. Ma'lumotlar bazasi mantiqiy va jismoniy ko'rinishda bo'lishi mumkin. Mantiqiy ko'rinishdagi ma'lumotlar bazasining asosini jadvallar tashkil etadi. Jadvallar ustun va qatorlardan tashkil topgan bo'lib, ular birlashgan joy(yacheyka) da ma'lumotlarni saqlaydi. Bu ko'rinishdagi ma'lumotlar bazasi relyatsion ma'lumotlar bazasi deyiladi. Baza eng kamida jadvaldan tashkil topgan bo'lishi shart, bu shart ma'lumotlar bazasi mavjud bo'lishning eng minimum shartidir

Bazada saqlanayotgan ma'lumotlarni xafsizligini saqlash maqsadida Oracle(MBBT) da juda ko'p ishlar amalga oshirilgan, biz ularni keyingi maqolalarda ko'ramiz. Ma'lumotlar bazasini tashkil etgan fayllarni: ma'lumotlar bazasi fayllari va ma'lumotlar bazasiga tegishli bo'lmagan fayllar ko'rinishida ajratishimiz mumkin. Ma'lumotlar bazasi fayllarida ma'lumotlar saqlanadi, ma'lumotlar bazasiga tegishli bo'lmagan fayllarda har xil protokollar, sozlashlar kabi qo'shimcha ma'lumotlar saqlanadi. Bu ma'lumotlar bilan Oracle administratorlari ruxsat bergan foydalanuvchilargina ishlay oladi

Oracle ning juda ko'p utilitlari (Rman, Oracle Data Guard, Oracle Data Grid) mavjud bo'lib, ular ham ma'lumotlarni foydalanuvchi uchun qulay

ko'rinishda taqdim etish(boshqarish, nazorat qilish) uchun ishlatiladi. Misol tariqasida Rman utilitasini oladigan bo'lsak, bu utilita bazani to'liq yoki qisman zahira nusxalarini juda katta xajmga kamaytirib(siqib) hosil qiladi va saqlash uchun o'zi chunarli bo'lgan kodlarga aylantiradi. Bazaga shikast etganda shu zahira nusxa orqali bazani qayta tiklash mumkin bo'ladi.

Oracle SQL kodlari orqali buyruqlarni qabul qiladi. Bu so'rov tili ma'lumotlar bazasini yuklash, uni to'xtatish, montirovka qilish, jadvallar yaratish, o'chirish, o'zgartirish, ma'lumotlar ichidan kerakligini chiqarib olish va boshqa vazifalar uchun ishlatiladi. Bu til juda ko'p Ma'lumotlar Bazasini Boshqarish Tizim lari uchun umumiy hisoblanadi. SQL kodlari bir yoki bir necha jadvallar ustida so'rovlarni amalga oshira oladi

Ma'lumotlarni boshqarishda foydalanuvchilarga qulaylik yaratish maqsadida Oracle dasturi yaratuvchilari Oracle Enterprise Manager deb nomlangan web sahifa yaratishgan bo'lib, bu sahifa orqali butun Oracle ni boshqarish mumkin bo'ladi. SQL kodlarini bilmaydigan foydalanuvchilar ham bu web sahifa orqali barcha vazifalarni bajarishi mumkin, faqatgina kerakli "silka" larni bosish kifoya. Har bir ma'lumotlar bazasi uchun alohida-alohida Enterprise Manager sahifasi bo'ladi, har bir sahifa portlar orqali ajratiladi. Barcha brauzerlarda bu sahifa yaxshi namoyon bo'ladi va bazani xafsizligiga portlarni boshqarish yo'li bilan erishish mumkindir.

2.3. Java swing

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

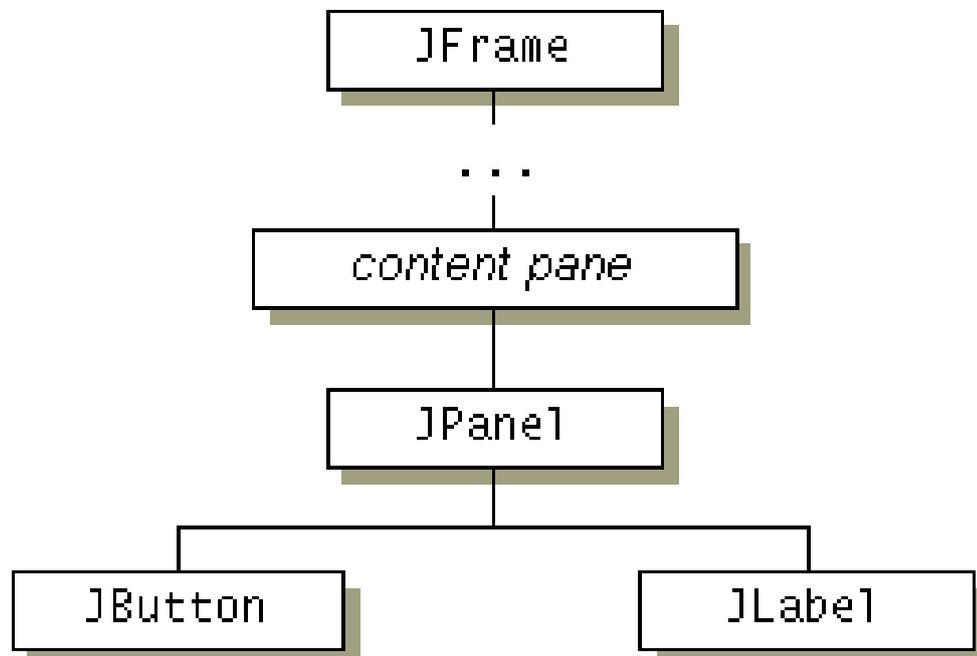
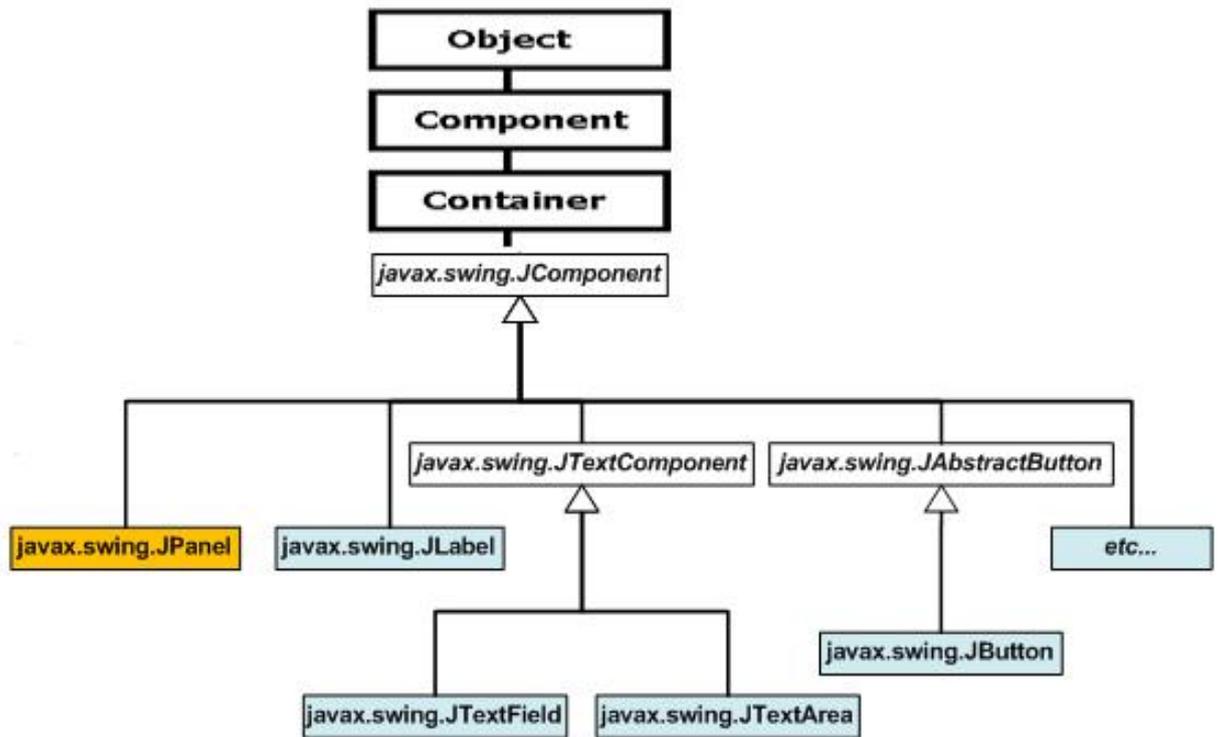
Java Swing Component’larini o’zgartirish uchun dasturlar

Etima, Jvider, Jgo, Visual Java (**SWING**) **Components**

Library, GUI Commands

Software Development Environments (SDE) for Eclipse,

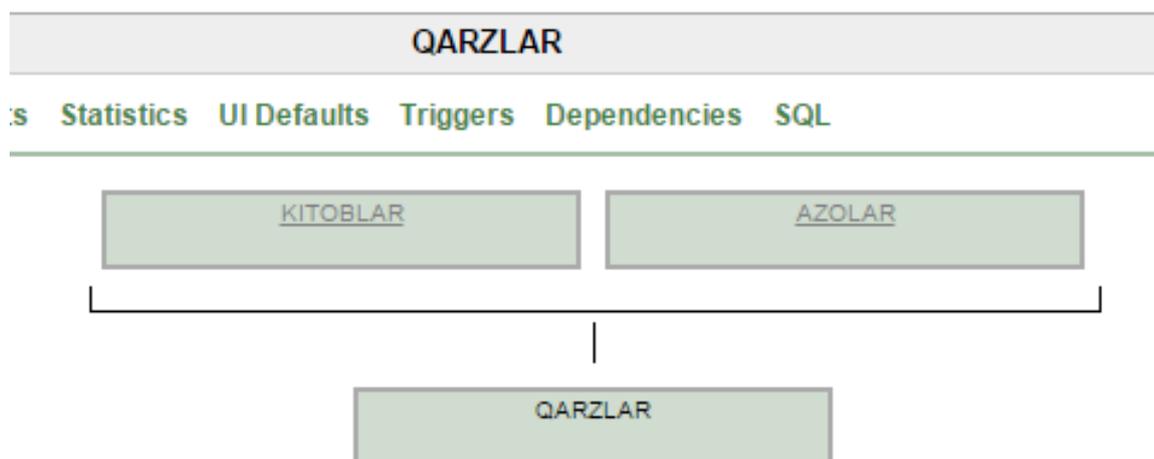
Netbeans, SunOne, Jbuilder



III. AMALIY QISM

3.1. Ma'lumotlar bazasi tuzilmasi

Ushbu kutubxona ishini avtomatlashtirish mavzusidagi dasturning ma'lumotlar bazasi tuzilmasi quyidagi ko'rinishga ega:



Ushbu tuzilmadan ko'rinish turibdiki “Qarzlar” jadvali “Kitoblar” va “Azolar” jadvaliga bog'langan.

Kitoblar jadvalida kutubxonada mavjud va foydalanishga olingan kitoblarning ro'yxatini ko'rishimiz mumkin. Uning ko'rinishi quyidagicha:

KITOBLAR										
Table	Data	Indexes	Model	Constraints	Grants	Statistics	UI Defaults	Triggers	Dependencies	SQL
Query	Count Rows	Insert Row								
EDIT	KITOBID	FAN	SARLAVHA	MUALLIF	NASHRIYOT	NASHRYILI	NASHRSONI	BETLAR	ISBN	KITOBLARSONI
	1	Dasturlash	Java dasturlash asoslari	Bill Geyts	USA	2012	1	555	82357326564	100
	21	Matematika	Diskret matematika	HolMatov A	Toshkent	2013	1	300	32435676573	100
	22	Iqtisod	Iqtisod asoslari	Tishtemirov Z	XXX	2001	2	232	324234234233	100
	23	Dasturlash	PHP boshlovchilar uchun	Hakimov Z	Oqituvchi	2009	1	230	76467757465	100

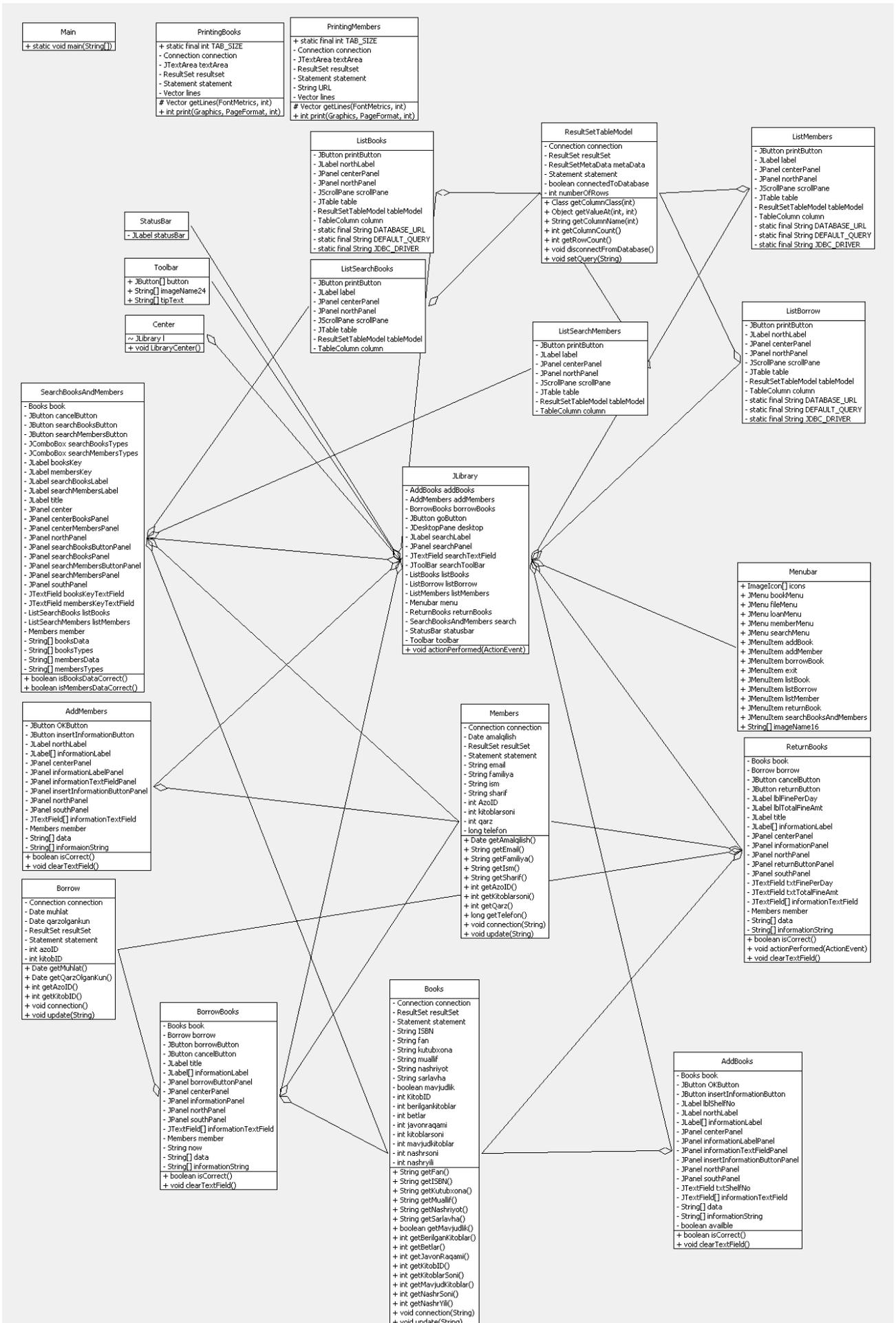
Azolar jadvalida kutubxonadan foydalanish uchun ro'yxatdan o'tgan a'zolarning ro'yxatini ko'rishimiz mumkin. Uning ko'rinishi quyidagicha:

AZOLAR										
Table	Data	Indexes	Model	Constraints	Grants	Statistics	UI Defaults	Triggers	Dependencies	SQL
Query	Count Rows	Insert Row								
EDIT	AZOID	TELEFON	ISM	FAMILIYA	SHARIF	EMAIL	KITOBLARSONI	QARZ	AMALQILISH	
	1	998909501218	Ikrom	Alimov	Axmadxon oqli	ikrom_2701@mail.ru	1	-	27-JAN-17	
	2	998909876543	Xumoyun	Allayarov	Sherali oqli	star@gmail.com	0	-	21-DEC-17	
	21	998909907777	Zokir	Zaripjonov	Sunnatovich	zokir@mail.ru	0	-	22-JAN-17	

Qarzar jadvalida kutubxona a'zosi tomonidan foydalanish uchun olingan kitoblar ro'yxati va qaytarish kunini ko'rishimiz mumkin. Uning ko'rinishi quyidagicha :

QARZLAR							
Table	Data	Indexes	Model	Constraints	Grants	Statistics	U
Query	Count Rows	Insert Row					
EDIT	KITOB_ID	AZO_ID	QARZOLGANKUN	MUHLAT			
	1	1	07-JAN-15	08-JAN-15			
	1	2	20-JAN-15	22-JAN-15			
	21	21	20-JAN-15	22-JAN-15			

3.2. UML sxemasi

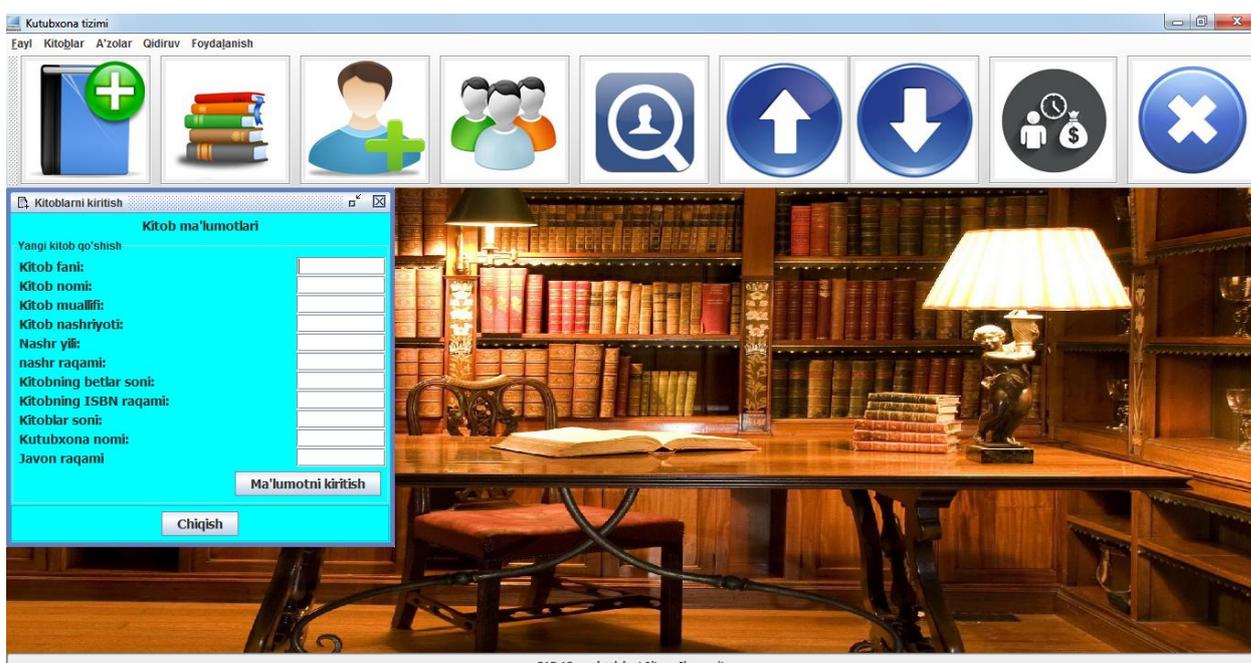


3.3. Dasturdan foydalanish

Dasturni ishga tushirganimizdan so'ng quyidagi oyna namoyon bo'ladi:



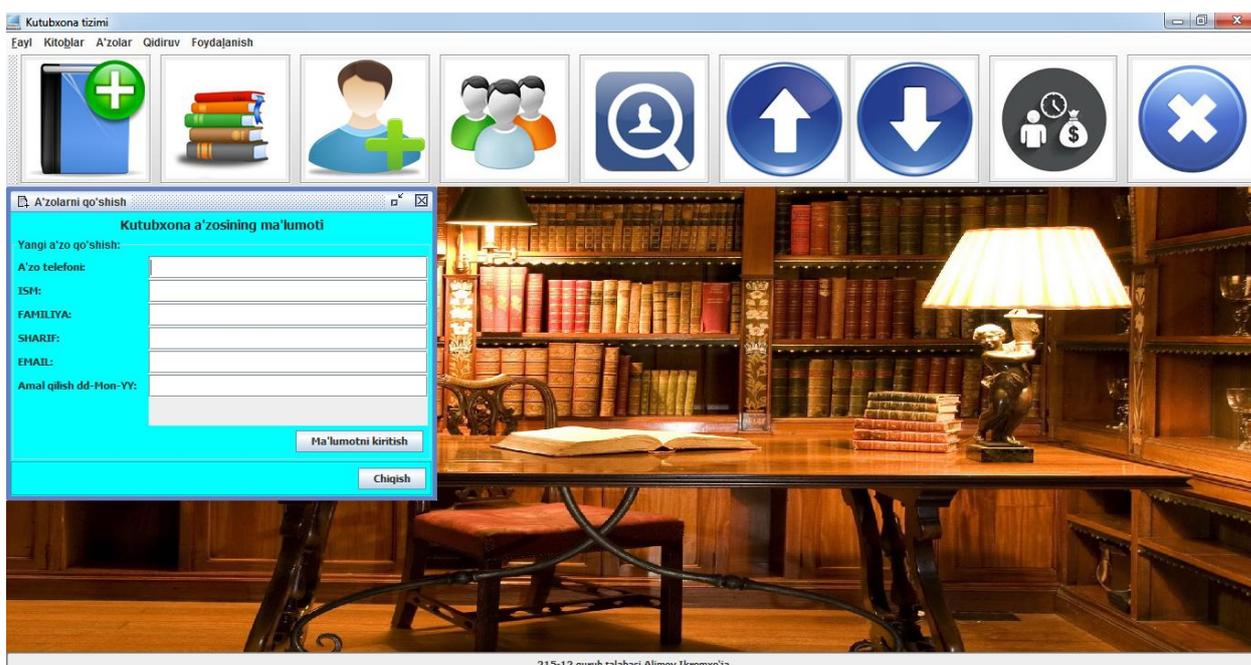
Dasturdagi "kitob qo'shish" tugmasi bosilganida quyidagi oyna hosil bo'ladi. Bu oynadagi sohalarni to'ldirib, "Malumotni kiritish" tugmasini bosish orqali ma'lumotlar bazasiga yangi kitobni qo'shishimiz mumkin.



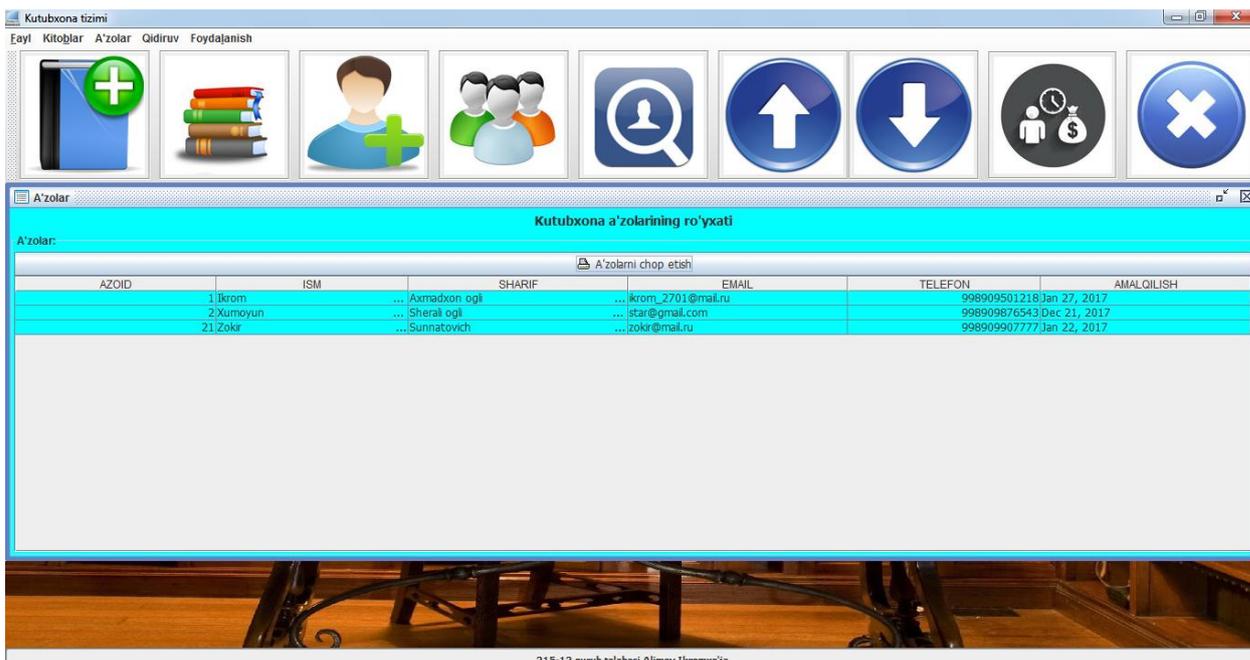
Kutubxonadagi mavjud kitoblar to'g'risida ma'lumot olish uchun “Kitoblar ro'yxati” tugmasi bosamiz. Shunda ekranga quyidagi jadval chiqadi:



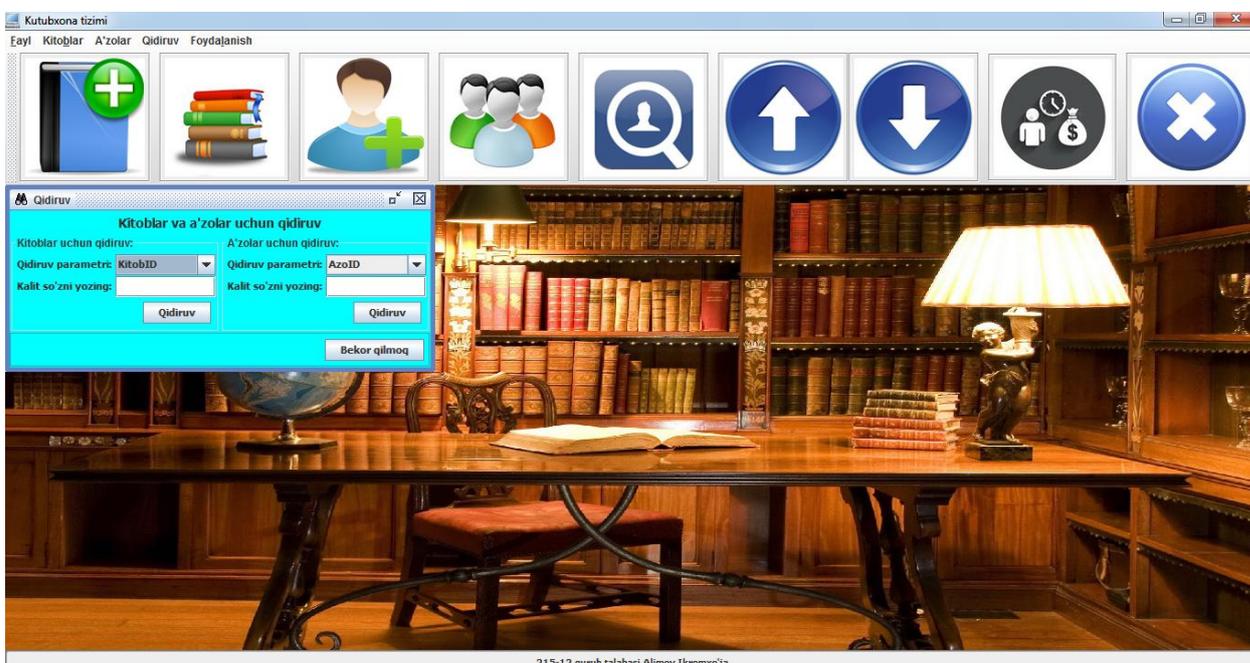
Kutubxonadan foydalanish uchun yangi a'zo ro'yxatga olish uchun “A'zo qo'shish” tugmasini bosim. Uning ko'rinishi quyidagicha:



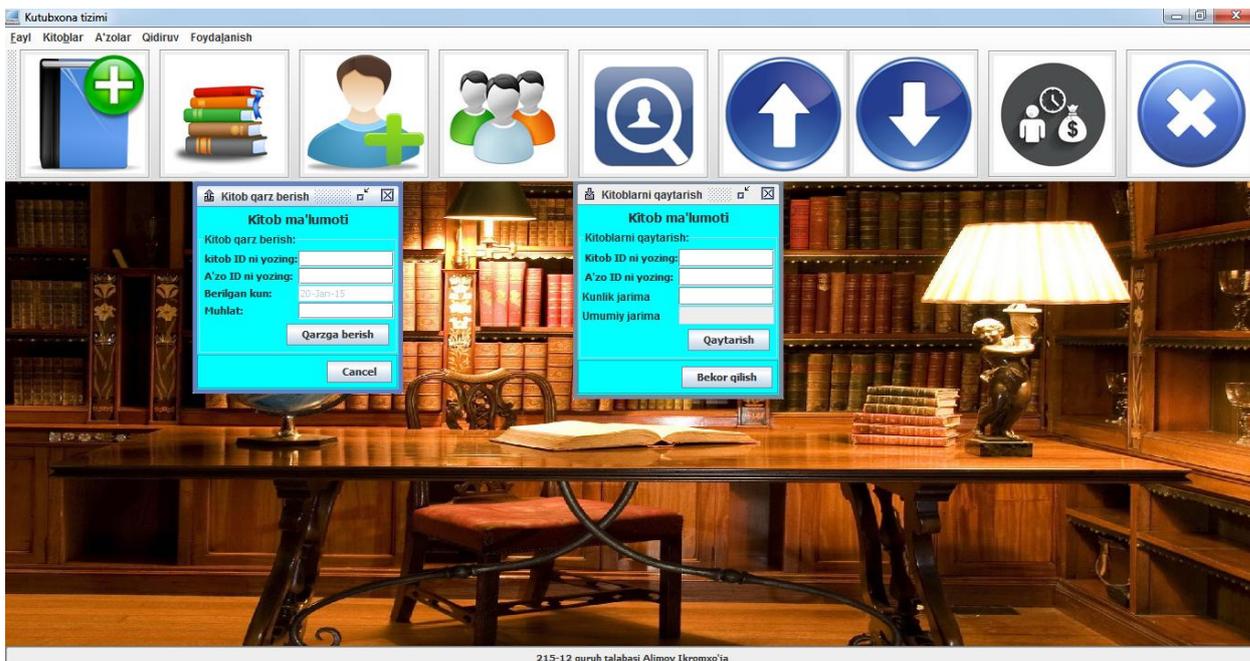
Kutubxonada mavjud a'zolar ro'yxatini ko'rish uchun “Azolar ro'yxati” tugmasini bosim. Tugmani bosish orqali quyidagi oyna hosil bo'ladi:



Agar bizga kutubxonadagi biror a'zo yoki kitob haqida ma'lumot kerak bo'lib qolsa uni qidiruv bo'limi orqali oson hal qilishimiz mumkin. Buning uchun dasturda "qidiruv" tugmasini bosamiz. Bu orqali kitob yoki a'zoning biror parametri bo'yicha ma'lumotni izlaymiz. Bu oynaning ko'rinishi quyidagicha:



Kutubxona a'zosiga kitobni foydalanishga berish va qaytarib olish quyidagi jadvallar asosida amalga oshiriladi:



Qaysi a'zoga qanday kitob foydalanishga berilganini bilish uchun "qarzlarni ro'yxati" tugmasini bosamiz. Shundan so'ng ekranga quyidagi oyna chiqadi:



Dasturdan chiqish uchun "chiqish" tugmasini bosing.

XULOSA

Xulosa qilib aytganda men bu kurs ishini bajarish mobaynida ko'pgina bilimlarga ega bo'ldim. Jumladan obyektga yo'naltirilgan java dasturlash tilini, javada foydalanuvchi interfeyslarini(Java Swing) yaratishni, malumotlar bazasi bilan ishlashni va uni java dasturlash tiliga bog'lashni, kutubxona ishini avtomatlashtirishni o'rgandim.

Hozirgi kunda axborot texnologiyalari deyarli barcha sohaga kirib bormoqda. Buning natijasida har xil sohaga tegishli masalalarni hal qilish uchun dasturlarga ehtiyoj sezilmoqda. Bu esa dasturchilarga talabni oshirmoqda. Men ushbu kurs ishida egallagan bilimlarimni yanada oshirib, o'zimga berilgan imkoniyatlardan foydalangan holda kelajakda turli masalalarni hal qilishga yo'naltirilgan dasturlar yaratishga harakat qilaman. Shu yo'l bilan yurtimizda axborot texnologiyalarining rivojlanishida o'z xissamni qo'shmoqchiman...

Foydalanilgan adabiyotlar

1. P. Nouton, G.Schildt Java
2. М.Арипов, А.Тиллаев «Веб-саҳифалар яратиш технологиялари» Тошкент.
2010
3. SQL ва маълумотлар базаларининг кейинги дастурлаш. Ш.Назирова,
Р.Қобулова. Тошкент - 2006.
4. www.google.com
5. www.java2s.com
6. www.oracle.com
7. www.akmx.uz va boshqa internet saytlari

ILOVA

```
import java.sql.*;
//import
oracle.jdbc.pool.OracleDataSource;

public class Books {
    private Connection connection
= null;
    private Statement statement =
null;
    private ResultSet resultSet =
null;

    private int KitobID;
    private String fan;
    private String sarlavha;
    private String muallif;
    private String nashriyot;
    private int nashryili;
    private int nashrsoni;
    private int betlar;
    private String ISBN;
    private int kitoblarsoni;
    private int mavjudkitoblar;
    private int berilgankitoblar;
    private String kutubxona;
    private boolean mavjudlik;
    private int javonraqami;
    public Books() {
    }

    public int getKitobID() {
        return KitobID;
    }

    public String getISBN() {
        return ISBN;
    }

    public int getKitoblarSoni() {
        return kitoblarsoni;
    }

    public int getMavjudKitoblar()
    {
        return mavjudkitoblar;
    }

    public int
getBerilganKitoblar() {
        return berilgankitoblar;
    }

    public boolean getMavjudlik()
    {
        return mavjudlik;
    }

    public void connection(String
Query) {
        try {

            Class.forName("oracle.jdbc.dri
ver.OracleDriver");
        }
        catch
(ClassNotFoundException cnfe) {

            System.out.println("Books.java
\n" + cnfe.toString());
        }
        catch (Exception e) {

            System.out.println("Books.java
\n" + e.toString());
        }

        try {
            connection =
DriverManager.getConnection("jdbc:
oracle:thin:@//127.0.0.1:1521/XE",
"system", "talaba21512");
            statement =
connection.createStatement();

```

```

        resultSet =
statement.executeQuery(Query);
        while
(resultSet.next()) {
            KitobID =
resultSet.getInt(1);
            fan =
resultSet.getString(2);
            sarlavha =
resultSet.getString(3);
            muallif =
resultSet.getString(4);
            nashriyot =
resultSet.getString(5);
            nashryili =
resultSet.getInt(6);
            nashrsoni =
resultSet.getInt(7);
            betlar =
resultSet.getInt(8);
            ISBN =
resultSet.getString(9);
            kitoblarsoni
= resultSet.getInt(10);

            mavjudkitoblar =
resultSet.getInt(11);

            berilgankitoblar =
resultSet.getInt(12);
            kutubxona =
resultSet.getString(13);
            mavjudlik =
resultSet.getBoolean(14);
            javonraqami
= resultSet.getInt(15);
        }
        resultSet.close();
        statement.close();
        connection.close();
    }
    catch (SQLException
SQLException) {
        System.out.println("Books.java
\n" + SQLLe.toString());
    }

    public void update(String
Query) {
        try {
            Class.forName("oracle.jdbc.dri
ver.OracleDriver");
        }
        catch
(ClassNotFoundException cnfe) {
            System.out.println("Books.java
\n" + cnfe.toString());
        }
        catch (Exception e) {
            System.out.println("Books.java
\n" + e.toString());
        }

        try {
            connection =
DriverManager.getConnection("jdbc:
oracle:thin:@//127.0.0.1:1521/XE",
"system", "talaba21512");
            statement =
connection.createStatement();

            statement.executeUpdate(Query
);

            statement.close();
            connection.close();
        }
        catch (SQLException
SQLException) {
            System.out.println("Boooks.jav
a\nError:" + SQLLe.toString());
        }
    }

```



```

public void connection()
{
    try {

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

    }
    catch
(ClassNotFoundException
cnfe) {

        System.out.println("Borr
ow.java\n" + cnfe.toString());

    }
    catch (Exception e)
{

        System.out.println("Borr
ow.java\n" + e.toString());

    }

    /**/
    /**/
    /**/

    * for making the
connection,creating the
statement and update *

    * the table in the
database. After that,closing the
statmenet *

```

```

* and connection.
There is catch block
SQLException for error *

    /**/
    /**/
    /***/

    try {

        connection =
DriverManager.getConnection(
"jdbc:oracle:thin:@//127.0.0.1:
1521/XE", "system",
"talaba21512");

        statement =
connection.createStatement();

        resultSet =
statement.executeQuery("SELE
CT * FROM Qarzlar");

        while
(resultSet.next()) {

            kitobID =
resultSet.getInt(1);

            azoID
= resultSet.getInt(2);

            qarzolgankun =
resultSet.getDate(3);

            muhlat
= resultSet.getDate(4);

```

```

        }
        resultSet.close();
        statement.close();
        connection.close();
    }
    catch
(SQLException SQLe) {
        System.out.println("Borr
ow.java\n" + SQLe.toString());
    }
}

public void update(String
Query) {
    try {
        Class.forName("oracle.jd
bc.driver.OracleDriver");
    }
    catch
(ClassNotFoundException
cnfe) {
        System.out.println("Borr
ow.java\n" + cnfe.toString());
    }
}
}
}

catch (Exception e)
{
    System.out.println("Borr
ow.java\n" + e.toString());
}

/*****
*****
*****

try {
    connection =
DriverManager.getConnection(
"jdbc:oracle:thin:@//127.0.0.1:
1521/XE", "system",
"talaba21512");
    statement =
connection.createStatement();
    statement.executeUpdate(
Query);
    statement.close();
    connection.close();
}
catch
(SQLException SQLe) {

```

```

        System.out.println("Borrow.java\n" + SQLe.toString());
    }
}

import java.sql.*;
import java.util.Date;

//import
oracle.jdbc.pool.OracleDataSource;

public class Members {
    /**
     * declaration of the
     * private variables used in the program
     */

    /**
     *
     */

    private Connection connection
= null;
    private Statement statement =
null;
    private ResultSet resultSet =
null;

    private int AzoID;
    private long telefon;
    private String ism;
    private String familiya;
    private String sharif;
    private String email;
    private int kitoblarsoni;
    private int qarz;
    private Date amalqilish;

```

```

        //private String URL =
        "jdbc:oracle:thin:system/talaba21512
        @//127.0.0.1:1521/XE";

        public Members() {
        }
        /*public Members(int
        memberID, int ID, String password,
        String name, String email, String
        major, int numberOfBooks, int mony,
        Date expired) {
            this.memberID =
            memberID;
            this.ID = ID;
            this.password =
            password;
            this.name = name;
            this.email = email;
            this.major = major;
            this.numberOfBooks =
            numberOfBooks;
            this.mony = mony;
            this.expired = expired;
        }*/

        public int getAzoID() {
            return AzoID;
        }
        public String getEmail() {
            return email;
        }

        public int getKitoblarsoni() {
            return kitoblarsoni;
        }

        public Date getAmalqilish() {
            return amalqilish;
        }

        public void connection(String
        Query) {

```

```

        try {
            Class.forName("oracle.jdbc.dri
ver.OracleDriver");
        }
        catch
(ClassNotFoundException cnfe) {
            System.out.println("Members.j
ava\n" + cnfe.toString());
        }
        catch (Exception e) {
            System.out.println("Members.j
ava\n" + e.toString());
        }

```

```

*****
*****
***/

```

```

        try {
            connection =
DriverManager.getConnection("jdbc:
oracle:thin:@//127.0.0.1:1521/XE",
"system", "talaba21512");
            statement =
connection.createStatement();
            resultSet =
statement.executeQuery(Query);
            while
(resultSet.next()) {
                AzoID=
resultSet.getInt(1);
                telefon =
resultSet.getLong(2);
                ism =
resultSet.getString(3);
                familiya =
resultSet.getString(4);
                sharif =
resultSet.getString(5);
                email =
resultSet.getString(6);

```

```

            kitoblarsoni
= resultSet.getInt(7);
            qarz =
resultSet.getInt(8);
            amalqilish =
resultSet.getDate(9);
        }
        resultSet.close();
        statement.close();
        connection.close();
    }
    catch (SQLException
SQLException) {
        System.out.println("Members.j
ava\n" + SQLLe.toString());
    }
}

```

```

    public void update(String
Query) {
        try {
            Class.forName("sun.jdbc.odbc.J
dbcOdbcDriver");
        }
        catch
(ClassNotFoundException cnfe) {
            System.out.println("Members.j
ava\n" + cnfe.toString());
        }
        catch (Exception e) {
            System.out.println("Members.j
ava\n" + e.toString());
        }
    }
}

```

```

/*****
*****
*****

```

```

        * for making the
connection,creating the statement and
update *

```



```

        this.add(searchMenu = new
JMenu("Qidiruv"));
        this.add(loanMenu = new
JMenu("Foydalanish"));

        icons = new ImageIcon[12];
        for (int i = 0; i <
imageName16.length; i++) {
            icons[i] = new
ImageIcon(java.lang.ClassLoader.get
SystemResource(imageName16[i]));
        }

        //for adding print books & exit
fileMenu.add(exit = new
JMenuItem("Chiqish", icons[1]));

        //for adding add, list,
listAvailble, listBorrowed, edit &
remove Books and book information
to the bookMenu
        bookMenu.add(addBook = new
JMenuItem("Kitob qo'shish",
icons[2]));
        bookMenu.add(listBook = new
JMenuItem("Kitoblar ro'yxati",
icons[3]));

        //for adding add, list, edit &
remove Members and member
information to the memberMenu
        memberMenu.add(addMember =
new JMenuItem("A'zo qo'shish",
icons[2]));
        memberMenu.add(listMember =
new JMenuItem("A'zolar ro'yxati",
icons[3]));

        //for adding add, list & remove
Members to the memberMenu

searchMenu.add(searchBooksAndMe
mbers = new JMenuItem("Qidiruv",
icons[7]));

        //for adding borrow & return
books to the loanMenu
        loanMenu.add(borrowBook =
new JMenuItem("Kitobni berish",
icons[8]));
        loanMenu.add(returnBook = new
JMenuItem("Kitobni qaytarib olish",
icons[9]));
        loanMenu.add(listBorrow = new
JMenuItem("Qarzlar ro'yxati",
icons[3]));

        exit.setAccelerator(KeyStroke.getKe
yStroke(KeyEvent.VK_F4,
ActionEvent.ALT_MASK));

        searchBooksAndMembers.setAcceler
ator(KeyStroke.getKeyStroke(KeyEve
nt.VK_S,
ActionEvent.CTRL_MASK));

        addBook.setAccelerator(KeyStroke.ge
tKeyStroke(KeyEvent.VK_A,
ActionEvent.CTRL_MASK));

        listBook.setAccelerator(KeyStroke.ge
tKeyStroke(KeyEvent.VK_L,
ActionEvent.CTRL_MASK));

        addMember.setAccelerator(KeyStrok
e.getKeyStroke(KeyEvent.VK_M,
ActionEvent.CTRL_MASK));

        listMember.setAccelerator(KeyStroke
.getKeyStroke(KeyEvent.VK_I,
ActionEvent.CTRL_MASK));

        returnBook.setAccelerator(KeyStroke.
getKeyStroke(KeyEvent.VK_R,
ActionEvent.CTRL_MASK));
    }
}

```