Санкт-Петербургский Государственный Политехнический Университет

Институт Информационных Технологий и Управления

Кафедра Компьютерных Систем и Программных Технологий

**Отчет**

**о лабораторной работе №7**

**Дисциплина:** Базы данных

**Тема:** Разработка клиентского приложения

**Выполнил**: гр.43501/1 Саитов Илья

**Преподаватель:** Мяснов А.В.

Санкт-Петербург 2015

1. **Цель работы:**

Создание простого клиентского приложения на языке Java, которое будет вести взаимодействие с БД.

1. **Программа работы:**

Необходимо создать консольное приложение выполняющее следующие функции:

- соединение с БД, выполнение фиксированного SQL-запроса и получение результатов запроса.

- добавление данных в одну из таблиц БД.

- выполнение хранимой процедуры.

1. **Выполнение работы:**

В ходе работы, через среду NetBeans 7.2.1 было создано консольное Java приложение реализующее следующий алгоритм:

* Инициализация драйвера.
* Создание подключения к базе данных.
* В зависимости от выбора пользователя:

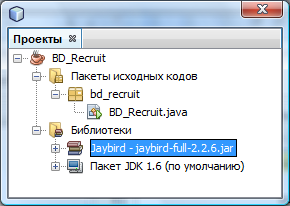
**а)** Выполнение фиксированного запроса на вывод списка всех таблиц БД и и в дальнейнешем просмотр содержимого выбранной таблицы.

**б)** Добавление записи в таблицу ListDoctor.

**в)** Запуск процедуры REC\_APPEAL (процедура создания призыва).

Также, приложение обрабатывает различные нештатные ситуации, такие как ввод посторонних символов, ошибка подключения к БД, ввод слишком больших значений.

Перед сборкой проекта была подключена библиотека драйвера Jaybird:



Листинг программы привден в **приложении №1**.

Тестирование приложения проводилось в командной строке Windows. Прописываем путь к скомпилированному файлу \*.jar и запускаем при помощи каманды: java -jar BD\_Recruit.jar

Листинг выполнения программы приведен ниже:

C:\Users\Windows Vista HP\Desktop\Учеба\Базы данных\Lab\_7 \BD\_Recruit\ dist>java -jar BD\_Recruit.jar

Нажмите на соответствующую клавишу:

1. Вывести список таблиц

2. Добавить ФИО в список врачей

3. Выполнить ХП по созданию призыва

4. Заверишить сеанс

**1**

Список таблиц базы данных военкомата:

1. ACTION

2. ADDRHOME

3. DISTRICTMILITARY

4. DIVISIONMILITARY

5. DOCTOR

6. GROUPREC

7. LISTDOCTOR

8. MEDICALDATA

9. MILITARYSERVICE

10. NAMERS

11. PARENT

12. POSTDOCTOR

13. POSTMILITARY

14. PRIVATEDATA

15. PROFESSION

16. RANKMILITARY

17. RECRUIT

18. RECSTATE

19. RELSHIP

20. SOCSTATE

21. SOCSTNAME

22. SPECIALMILITARY

23. STATUS

24. STORYRECRUIT

25. STUDY

Введите номер таблицы для отображения ее содержимого:

7

LISTDOC\_ID | FULLNAME |

1 Вяткин Игорь Владимирович

2 Жмуркин Михаил Петрович

3 Веселов Александр Кириллович

4 Игораев Никита Владимирович

5 Лубсанов Баир Александрович

6 Кравченко Андрей Георгиевич

7 Жуков Александр Степанович

8 Николаев Игорь Николаевич

9 Иванов Иван Иванович

10 Петров Петр Петрович

11 Бессонова Наталья Николаевна

12 Маришко Людмила Петровна

Нажмите на соответствующую клавишу:

1. Вывести список таблиц

2. Добавить ФИО в список врачей

3. Выполнить ХП по созданию призыва

4. Заверишить сеанс

**2**

Введите ФИО врача

Саитов Илья Петрович

Нажмите на соответствующую клавишу:

1. Вывести список таблиц

2. Добавить ФИО в список врачей

3. Выполнить ХП по созданию призыва

4. Заверишить сеанс

1

Список таблиц базы данных военкомата:

1. ACTION

2. ADDRHOME

3. DISTRICTMILITARY

4. DIVISIONMILITARY

5. DOCTOR

6. GROUPREC

7. LISTDOCTOR

8. MEDICALDATA

9. MILITARYSERVICE

10. NAMERS

11. PARENT

12. POSTDOCTOR

13. POSTMILITARY

14. PRIVATEDATA

15. PROFESSION

16. RANKMILITARY

17. RECRUIT

18. RECSTATE

19. RELSHIP

20. SOCSTATE

21. SOCSTNAME

22. SPECIALMILITARY

23. STATUS

24. STORYRECRUIT

25. STUDY

Введите номер таблицы для отображения ее содержимого:

7

LISTDOC\_ID | FULLNAME |

1 Вяткин Игорь Владимирович

2 Жмуркин Михаил Петрович

3 Веселов Александр Кириллович

4 Игораев Никита Владимирович

5 Лубсанов Баир Александрович

6 Кравченко Андрей Георгиевич

7 Жуков Александр Степанович

8 Николаев Игорь Николаевич

9 Иванов Иван Иванович

10 Петров Петр Петрович

11 Бессонова Наталья Николаевна

12 Маришко Людмила Петровна

17 Саитов Илья Петрович

Нажмите на соответствующую клавишу:

1. Вывести список таблиц

2. Добавить ФИО в список врачей

3. Выполнить ХП по созданию призыва

4. Заверишить сеанс

1

Список таблиц базы данных военкомата:

1. ACTION

2. ADDRHOME

3. DISTRICTMILITARY

4. DIVISIONMILITARY

5. DOCTOR

6. GROUPREC

7. LISTDOCTOR

8. MEDICALDATA

9. MILITARYSERVICE

10. NAMERS

11. PARENT

12. POSTDOCTOR

13. POSTMILITARY

14. PRIVATEDATA

15. PROFESSION

16. RANKMILITARY

17. RECRUIT

18. RECSTATE

19. RELSHIP

20. SOCSTATE

21. SOCSTNAME

22. SPECIALMILITARY

23. STATUS

24. STORYRECRUIT

25. STUDY

Введите номер таблицы для отображения ее содержимого:

9

ID\_SERVICE | PERSON\_ID | ACTION | RANK | POST | SPECIAL | MIL\_DIVISION | DATE\_FROM | DATE\_TO |

21 62 1 1 1 1 1 2015-01-18 2017-10-14

22 63 1 1 1 1 1 2015-01-18 2017-10-14

6 7 1 1 1 7 1 2014-06-01 2015-06-01

23 64 1 1 1 1 1 2015-01-18 2017-01-07

8 9 1 1 1 2 1 2009-09-01 2010-09-01

9 10 1 1 1 1 1 2009-03-01 2010-03-01

24 65 1 1 1 1 1 2015-01-18 2017-01-07

11 5 1 1 1 1 1 2015-01-16 2016-01-16

13 13 1 1 1 1 1 2015-01-16 2016-01-16

14 17 1 1 1 1 1 2015-01-16 2016-01-16

15 20 1 1 1 3 5 2015-01-16 2016-01-16

16 24 1 1 1 3 5 2015-01-16 2016-01-16

17 29 1 1 1 3 5 2015-01-16 2016-01-16

18 30 1 1 1 3 5 2015-01-16 2016-01-16

19 34 1 1 1 3 5 2015-01-16 2016-01-16

20 42 1 1 1 3 5 2015-01-16 2016-01-16

Нажмите на соответствующую клавишу:

1. Вывести список таблиц

2. Добавить ФИО в список врачей

3. Выполнить ХП по созданию призыва

4. Заверишить сеанс

**3**

Введите количество людей, подлежащих призыву:

3

Введите идентификатор звания:

1

Введите идентификатор должности:

1

Введите идентификатор специальности:

1

Введите идентификатор военной части:

8

Введите срок службы призывников:

735

Было успешно призвано 3 человек сроком на 735 дней

Нажмите на соответствующую клавишу:

1. Вывести список таблиц

2. Добавить ФИО в список врачей

3. Выполнить ХП по созданию призыва

4. Заверишить сеанс

1

Список таблиц базы данных военкомата:

1. ACTION

2. ADDRHOME

3. DISTRICTMILITARY

4. DIVISIONMILITARY

5. DOCTOR

6. GROUPREC

7. LISTDOCTOR

8. MEDICALDATA

9. MILITARYSERVICE

10. NAMERS

11. PARENT

12. POSTDOCTOR

13. POSTMILITARY

14. PRIVATEDATA

15. PROFESSION

16. RANKMILITARY

17. RECRUIT

18. RECSTATE

19. RELSHIP

20. SOCSTATE

21. SOCSTNAME

22. SPECIALMILITARY

23. STATUS

24. STORYRECRUIT

25. STUDY

Введите номер таблицы для отображения ее содержимого:

9

ID\_SERVICE | PERSON\_ID | ACTION | RANK | POST | SPECIAL | MIL\_DIVISION | DATE\_FROM | DATE\_TO |

21 62 1 1 1 1 1 2015-01-18 2017-10-14

22 63 1 1 1 1 1 2015-01-18 2017-10-14

6 7 1 1 1 7 1 2014-06-01 2015-06-01

23 64 1 1 1 1 1 2015-01-18 2017-01-07

8 9 1 1 1 2 1 2009-09-01 2010-09-01

9 10 1 1 1 1 1 2009-03-01 2010-03-01

24 65 1 1 1 1 1 2015-01-18 2017-01-07

11 5 1 1 1 1 1 2015-01-16 2016-01-16

25 66 1 1 1 1 8 2015-01-19 2017-01-23

13 13 1 1 1 1 1 2015-01-16 2016-01-16

14 17 1 1 1 1 1 2015-01-16 2016-01-16

15 20 1 1 1 3 5 2015-01-16 2016-01-16

16 24 1 1 1 3 5 2015-01-16 2016-01-16

17 29 1 1 1 3 5 2015-01-16 2016-01-16

18 30 1 1 1 3 5 2015-01-16 2016-01-16

19 34 1 1 1 3 5 2015-01-16 2016-01-16

20 42 1 1 1 3 5 2015-01-16 2016-01-16

26 67 1 1 1 1 8 2015-01-19 2017-01-23

27 84 1 1 1 1 8 2015-01-19 2017-01-23

Нажмите на соответствующую клавишу:

1. Вывести список таблиц

2. Добавить ФИО в список врачей

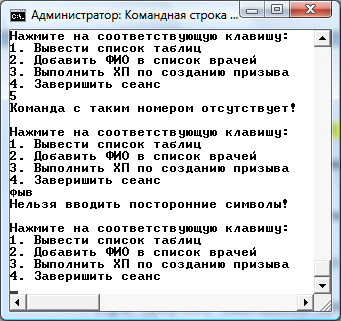
3. Выполнить ХП по созданию призыва

4. Заверишить сеанс

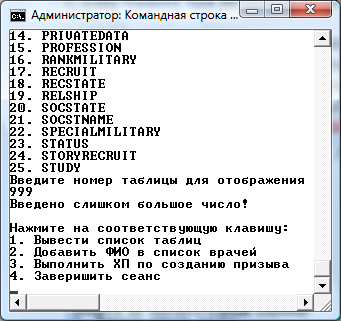
**4**

C:\Users\Windows Vista HP\Desktop\Учеба\Базы данных\Lab\_7\BD\_Recruit \dist>

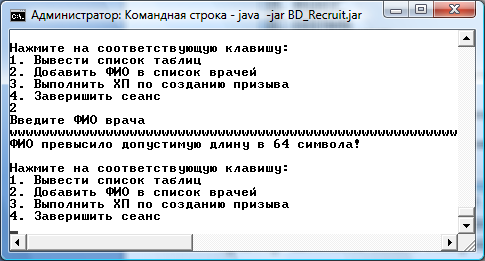
При запуске, выводится сообщение с приглашением ввести одну из имеющихся четырех команд. Стоит отметить, что при вводе числа больше, чем 4 или посторонних символов выводятся сообщения с предупреждениями, и пользователю вновь предлагается выбрать одну из четырех команд:



При выполнении первой команды, у нас выводится всех имеющихся таблиц в БД, затем, по требованию пользователя выводится содержимое одной из таблиц. Здесь также, как и выше имеется защита от неверного ввода номера таблицы, при этом, работа приложения продолжается, но только уже требуется снова вводить первую команду:



При выполнении команды №2, пользователю сразу предлагается ввести ФИО врача, которого хотим добавить в таблицу. На данной таблице стоит триггер автоматической генерации первичного ключа, что очень удобно для пользователя. При вводе ФИО больше, чем 64 символа сработает защита:



При выполнении третьей команды, пользователю поочередно предлагается ввести входные параметры запускаемой процедуры. При вводе символов сработает защита как и в вводе номера команды.

1. **Вывод:**

Выполнив лабораторную работу №7, мы ознакомились с принципом создания простого клиентского приложения на языке Java, которое будет вести взаимодействие с БД. Были изучены механизмы и способы реализации клиентских приложений для организации взаимодействия с базой данных. Приложение прошло проверку на работоспособность – все команды дают ожидаемый результат, обработка нештатных ситуаций ведется корректно.

Разработанное приложение использует трех уровневую модель программного комплекса, организующего взаимодействие между конечным пользователем и базой данных. Трехуровневая модель – это модель, которая включает в себя **клиентское приложение** (с помощью драйвера оно связывается с сервером приложений и сервером БД, и содержит в себе основную бизнес логику), **сервер приложений** (содержит в себе большую часть бизнес логики и распределяет нагрузку) и **сервер БД** (нужный для хранения данных и выполнения хранимых процедур, триггеров).

Преимущество данной модели по сравнению с обычной клиент серверной:

- высокая безопасность и надежность построенного комплекса;

- масштабируемость;

- быстродействие и балансировка нагрузки;

- невысокие требования к качеству связи между клиентом и сервером приложений.

К недостаткам трехуровневой модели можно отнести сложность разработки программного комплекса и возможное наложение дополнительных издержек в администрировании.

Цель данной работы была достигнута. Полученные знания могут быть использованы в дальнейшем на практике при работе с БД.

**Приложение №1**. **Листинг приложения:**

package bd\_recruit;

import java.sql.\*;

import java.io.PrintStream;

import java.util.Scanner;

import java.util.Vector;

import javax.print.DocFlavor;

/\*\*

\*

\* @author Саитов Илья

\*/

public class BD\_Recruit {

public static void main(String[] args) throws Exception {

// Объявляем переменные:

String temp2 = null; // Временные строковые

String [] temp3 = null; //строковые переменные

Vector<String> vec\_tab = new Vector<String>(); // Вектор для хранения списка таблиц

ResultSet rs\_data; // "Курсор" для указания текущей строки

int numb\_command = 0; // Номер введенной команды

int numb\_table = 0; // Номер введенной таблицы

int t1, t2, t3, t4, t5, t6; // Временные целые для аргументов процедуры

int cnt\_col = 0; // Количество столбцов в таблице

// Указываем путь к БД, записывая его в переменную:

String strURL = "jdbc:firebirdsql://localhost/G:\\\\RECRUIT.fdb";

// Инициализация Firebird JDBC driver:

Class.forName("org.firebirdsql.jdbc.FBDriver").newInstance();

PrintStream printStream = new PrintStream(System.out, true, "cp866");

Scanner sc = new Scanner(System.in, "cp866");

//Создание подключение к БД:

Connection connect=null;

connect = DriverManager.getConnection(strURL,"SYSDBA", "masterkey");

if (connect==null)

{

printStream.println("Подключение к БД завершилось ошибкой");

}

// Получаем список всех таблиц БД:

DatabaseMetaData metaData = connect.getMetaData();

ResultSet temp=metaData.getTables(temp2, temp2, temp2, temp3);

while(temp.next())

{

temp2=temp.getString(3);

if(!temp2.contains("$"))

vec\_tab.add(temp2);

}

// Создаём класс, с помощью которого будут выполняться SQL запросы:

Statement stmt = connect.createStatement();

while(numb\_command!=4)

{

printStream.println();

printStream.println("Нажмите на соответствующую клавишу:");

printStream.println("1. Вывести список таблиц");

printStream.println("2. Добавить ФИО в список врачей");

printStream.println("3. Выполнить ХП по созданию призыва");

printStream.println("4. Заверишить сеанс");

try{

numb\_command=Integer.parseInt(sc.nextLine());

}catch(NumberFormatException e){

printStream.println("Нельзя вводить посторонние символы!");

continue;

}

if(numb\_command>4){

printStream.println("Команда с таким номером отсутствует!");

}

// Если нажали на 1, то выполняем команду 1:

if(numb\_command==1)

{

printStream.println("Список таблиц базы данных военкомата:");

for(int i=1;i<=vec\_tab.size();i++)

{

System.out.printf("%d. %s\n",i,vec\_tab.elementAt(i-1));

}

printStream.println("Введите номер таблицы для отображения ее содержимого:");

try{

numb\_table=Integer.parseInt(sc.nextLine());

}catch(NumberFormatException e){

printStream.println("Нельзя вводить посторонние символы!");

continue;

}

if(numb\_table > vec\_tab.size()){

printStream.println("Введено слишком большое число!");

continue;

}

printStream.println();

//Выполняем SQL запрос:

rs\_data = stmt.executeQuery("SELECT \* from "+ vec\_tab.elementAt(numb\_table-1));

// Выводим результат:

cnt\_col = rs\_data.getMetaData().getColumnCount();

// Выводим содержание таблицы:

// Сначала имена столбцов:

for(int i = 1; i < cnt\_col + 1; i++){

printStream.print(rs\_data.getMetaData().getColumnName(i)+

" | ");

}

// Затем сами записи в таблице:

while(rs\_data.next())

{

printStream.println();

for (int i = 1;i < cnt\_col + 1;i++)

{

Object obj = rs\_data.getObject(i);

if (obj!=null)

{

printStream.print(obj+" \t ");

}

}

}

printStream.println();

continue;

}

// Если нажали на 2, то выполняем команду 2:

if(numb\_command==2)

{

printStream.println("Введите ФИО врача");

String tstr=sc.nextLine();

if(tstr.length()>64)

{

printStream.println("ФИО превысило допустимую длину в 64 символа!");

continue;

}

tstr="INSERT INTO listdoctor(FULLNAME) VALUES ('"+tstr+"')";

stmt.executeUpdate(tstr);

continue;

}

// Если нажали на 3, то выполняем команду 3:

if(numb\_command==3)

{

try{

printStream.println("Введите количество людей, подлежащих призыву:");

t1=Integer.parseInt(sc.nextLine());

printStream.println("Введите идентификатор звания:");

t2=Integer.parseInt(sc.nextLine());

printStream.println("Введите идентификатор должности:");

t3=Integer.parseInt(sc.nextLine());

printStream.println("Введите идентификатор специальности:");

t4=Integer.parseInt(sc.nextLine());

printStream.println("Введите идентификатор военной части:");

t5=Integer.parseInt(sc.nextLine());

printStream.println("Введите срок службы призывников:");

t6=Integer.parseInt(sc.nextLine());

// Вызов хранимой процедуры без возвращаемых параметров:

PreparedStatement call\_stmt=connect.prepareStatement("{call rec\_appeal(?,?,?,?,?,?)}");

call\_stmt.setInt(1, t1);

call\_stmt.setInt(2, t2);

call\_stmt.setInt(3, t3);

call\_stmt.setInt(4, t4);

call\_stmt.setInt(5, t5);

call\_stmt.setInt(6, t6);

call\_stmt.execute();

printStream.println();

printStream.println("Было успешно призвано "+t1+" человек сроком на "+t6+" дней");

continue;

}catch(NumberFormatException e){

printStream.println("Вводить можно только числа!");

continue;

}

}

}

// Если нажали 4, то освобождаем ресурсы и завершаем работу.

stmt.close();

connect.close();

}

}