

Санкт-Петербургский политехнический университет Петра Великого  
Институт компьютерных наук и технологий  
Кафедра компьютерных систем и программных технологий

**Отчёт о лабораторной работе №8**

**Дисциплина:** Базы данных

**Тема:** Консольное приложение

Выполнил студент гр. 43501/1

\_\_\_\_\_ Данг Хань  
(подпись)

Руководитель

\_\_\_\_\_ А.В. Мяснов  
(подпись)

“\_\_” \_\_\_\_\_ 2016 г.

Санкт-Петербург

2016

## 1. Цель работы

Ознакомиться с разработкой клиентских приложений.

## 2. 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
- Реализовать импорт данных не менее, чем из двух связанных таблиц из файлов в формате JSON. Вызов хранимой процедуры из приложения. Экспорт данных в формат XML.

## 3. Выполнение работы

Листинг:

**bookstore.java**

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package book_store;

import org.json.simple.JSONArray;
import org.json.simple.JSONObject;
import org.json.simple.parser.JSONParser;
import org.json.simple.parser.ParseException;
import org.w3c.dom.Document;

import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.sql.Connection;
import java.sql.DatabaseMetaData;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import book_store.RS2DOM.RS2DOM;
import java.io.FileReader;
import java.io.IOException;
import java.util.Scanner;
import java.util.Vector;
import javax.xml.transform.Transformer;
import javax.xml.transform.TransformerFactory;
import javax.xml.transform.dom.DOMSource;
import javax.xml.transform.stream.StreamResult;
```

```

import org.w3c.dom.Node;
/**
 *
 * @author khanh
 */
public class book_store {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) throws ClassNotFoundException,
SQLException,
        InstantiationException, IllegalAccessException, FileNotFoundException,
ParseException, IOException {
        // TODO code application logic here
        Connection connect = null;
        java.sql.Statement rqst = null; // объект для выполнения SQL запросов
        Scanner sc = new Scanner(System.in); // класс для работы с консолью
        int table_number = 0; // номер введенной таблицы
        Vector<String> vec_tab = new Vector<String>();
        String temp2 = null; // временные строковые переменные
        String [] temp3 = null;
        StringBuilder sb = new StringBuilder(); //объект для построения строки
        int menu_select = 0; // переменная = выбранный пункт меню
        int argz; // входной аргумент для операции INSERT
        int argz3; // входной аргумент для операции INSERT
        int argz4; // входной аргумент для операции INSERT
        int cnt_col = 0; //переменная для вывода содержимого таблицы
        ResultSet res; // Класс для хранения результатов SQL запроса
        Document doc = null; //Класс для хранения XML

        // Инициализация драйвера
        Class.forName("org.firebirdsql.jdbc.FBDriver").newInstance();
        //Указание пути к БД
        String strPath = "jdbc:firebirdsql://localhost/D:/BS.FDB";

        Class.forName("org.firebirdsql.jdbc.FBDriver").newInstance();
        //Подключение к БД
        connect = DriverManager.getConnection(strPath, "SYSDBA", "masterkey");
        if (connect == null) {
            System.err.println("Невозможно подключиться к БД.");}
        //Создание класса для выполнения SQL запросов
        rqst = connect.createStatement();
        System.out.println("Подключение к БД успешно выполнено.");
        // Получение списка 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);
        }
        // Меню
        while (menu_select != 5) {
            // ВЫВОД МЕНЮ

```

```

System.out.println("-----");
System.out.println("|                LIBRARY_                |");
System.out.println("-----");
System.out.println("Функции:");
System.out.println("1.Вывод списка всех таблиц");
System.out.println("2.Добавление записи в таблицу LAB8");
// System.out.println("3.Выполнение хранимой процедуры ");
System.out.println("3.Импорт из JSON данных в LISTBOOK и AUTHOR");
System.out.println("4.Экспорт содержимого таблицы в XML");
System.out.println("5.Выход");
// считывание номера пункта меню
System.out.println("Выберите пункт меню:");
try{
    menu_select = Integer.parseInt(sc.nextLine());
}catch(NumberFormatException e){
    System.err.println("Ошибка! Вводите только цифры.");
    continue;
}
if (menu_select > 5)
    System.err.println("Ошибка! Пункт меню с таким номером
отсутствует.");

if (menu_select == 1)
{
    System.out.println("Список таблиц:");
    for(int i=1;i<=vec_tab.size();i++)
    {
        System.out.printf("%d. %s\n",i,vec_tab.elementAt(i-1));
    }
    System.out.println("Введите номер таблицы для
отображения ее содержимого или "
+ "\n0 для возврата в основное меню:");
    try{
        table_number=Integer.parseInt(sc.nextLine());
    }catch(NumberFormatException e){
        System.err.println("Ошибка! Номер должен быть числом!");
        continue;
    }
    if((table_number > vec_tab.size()) || (table_number < 0)){
        System.err.println("Ошибка! Таблица с таким номером
отсутствует.");
        continue;
    }
    if(table_number == 0){
        continue;
    }
    System.out.println();

```

```

        //Выполнение SQL запроса
        res = rqst.executeQuery("SELECT * from "+
vec_tab.elementAt(table_number-1));
        // Вывод результата
        cnt_col = res.getMetaData().getColumnCount();
        // Вывод содержимого таблицы
        // Сначала имена столбцов:
        for(int i = 1; i < cnt_col + 1; i++){
            System.out.print(res.getMetaData().getColumnName(i)+
                " | ");
        }
        // Затем сами записи в таблице:
        while(res.next())
        {
            System.out.println();
            for (int i = 1;i < cnt_col + 1;i++)
            {
                Object obj = res.getObject(i);
                if (obj!=null)
                {
                    System.out.print(obj+" \t ");
                }
            }
            System.out.println();

            continue;
        }

        if (menu_select == 2 )
        {
            if(connect == null) {
                System.err.println("Соединение с БД не установлено.");
                continue;
            }
            // ВВОД АРГУМЕНТОВ ДЛЯ ОПЕРАЦИИ INSERT В ТАБЛИЦУ LAB8
            System.out.println("Введите ID_LAB:");
            try{
                argz=Integer.parseInt(sc.nextLine());
            }catch(NumberFormatException e){
                System.err.println("Ошибка! ID не является числом или превышает 9
СИМВОЛОВ.");
            }
            continue;
        }
        if (argz <= 0)
        {
            System.err.println("Ошибка! ID не может быть отрицательным
или равным нулю.");
            continue;
        }
    }
}

```

```

        System.out.println("Введите название LAB8:");
        String argz2 = sc.nextLine();
        if (argz2.length() > 25 || argz2.isEmpty())
        {
            System.err.println("Ошибка! Название типа не может быть
пустым или больше 25 символов.");
            continue;
        }

        try{
            rqst.executeUpdate("insert into LAB8 values
('"+argz+"','"+argz2+"')");
            System.out.println("Запись добавлена в таблицу.");
        }catch (SQLException se){
            System.out.println(se.getMessage());
        }

        continue;
    }

    if (menu_select == 3)
    {

        JSONParser parser = new JSONParser(); //создание объекта для парсинга
        String textjson =
filework.read("C:\\Users\\dangk\\Documents\\NetBeansProjects\\book_store\\src\\newpac
kage\\LIST_BOOKS.json");
        Object obj = parser.parse(textjson);
        JSONObject jsonObj = (JSONObject) obj;
        JSONArray jo = (JSONArray) jsonObj.get("LIST_BOOKS");

        //Добавление данных в таблицу list_book
        for (int i = 0; i < jo.size(); i++){
            JSONObject element = (JSONObject) jo.get(i);
            try{
                rqst.executeUpdate("insert into list_books values
('"+element.get("ID_LISTBOOKS")+"','"+element.get("NAME")+"')");
            }catch (SQLException se){
                System.out.println(se.getMessage());
            }
        }

        textjson =
filework.read("C:\\Users\\dangk\\Documents\\NetBeansProjects\\book_store\\src\\newpac
kage\\AUTHOR.json");
        obj = parser.parse(textjson);
        jsonObj = (JSONObject) obj;
        jo = (JSONArray) jsonObj.get("AUTHOR");

        for (int ii=0; ii < jo.size(); ii++){
            JSONObject element = (JSONObject) jo.get(ii);
            try{
                rqst.executeUpdate("insert into AUTHOR values

```

```

('"+element.get("ID_AUTHOR")+"', '"+element.get("NAME_AUTHOR")+"', '"+element.get("RATI
NG_AUTHOR")+"'");
        }catch (SQLException se){
            System.out.println(se.getMessage());
        }
    }

    System.out.println("\nИмпорт данных из JSON файлов в таблицы
LIST_BOOK и AUTHOR выполнен.");

    continue;
}

if (menu_select ==4)
{
    System.out.println("Список таблиц:");
    for(int i=1;i<=vec_tab.size();i++)
    {
        System.out.printf("%d. %s\n",i,vec_tab.elementAt(i-1));
    }

    System.out.println("Введите номер таблицы для
экспорта в XML:");

    try{
        table_number=Integer.parseInt(sc.nextLine());
    }catch(NumberFormatException e){
        System.err.println("Ошибка! Номер должен быть числом!");
        continue;
    }
    if((table_number > vec_tab.size()) || (table_number < 0)){
        System.err.println("Ошибка! Таблица с таким номером
отсутствует.");
        continue;
    }
    if(table_number == 0){
        continue;
    }
    System.out.println();
    //Выполнение SQL запроса
    res = rqst.executeQuery("SELECT * from "+
vec_tab.elementAt(table_number-1));

    Document xsd = RS2DOM.ResultSet2XSDDOM(res);
    Document d = RS2DOM.ResultSet2DOM(res);
    try {
        Transformer myTransformer =
            (TransformerFactory.newInstance()).newTransformer();
        System.out.println(
            "Схема, описывающая XML, экспортирована в файл
Description.xml");
        myTransformer.transform(
            new DOMSource(xsd),

```

```

        new StreamResult(new
FileOutputStream("C:\\Users\\dangk\\Documents\\NetBeansProjects\\book_store\\src\\new
package\\BOOK.xml"))));
        System.out.println(
            "\n\nСодержимое таблицы экспортировано в XML файл Data.xml");
        myTransformer.transform(
            new DOMSource(d),
            new StreamResult(new
FileOutputStream("C:\\Users\\dangk\\Documents\\NetBeansProjects\\book_store\\src\\new
package\\BOOK1.xml"))));
    } catch (Exception e) {
        e.printStackTrace();
    }
    continue;
}

if (menu_select == 5)
{
    System.out.println("ББ");
    continue;
}
}
System.exit(0);

}

}

```

## filework.java

```

/*
 * To change this license header, choose License Headers in Project
Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package libraryfilm;
import java.io.BufferedReader;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;
/**
 *
 * @author quyettran

```



```

*/
public class filework {
public static String read(String fileName) throws FileNotFoundException,
IOException {
    //Этот спец. объект для построения строки
    //Определяем файл
    File file = new File(fileName);

    StringBuilder sb = new StringBuilder();

    exists(fileName);

    try {
        //Объект для чтения файла в буфер
        BufferedReader in = new BufferedReader(new FileReader(
file.getAbsolutePath()));
        try {
            //В цикле построчно считываем файл
            String s;
            while ((s = in.readLine()) != null) {
                sb.append(s);
                sb.append("\n");
            }
        } finally {
            //Также не забываем закрыть файл
            in.close();
        }
    } catch (IOException e) {
        throw new RuntimeException(e);
    }

    //Возвращаем полученный текст с файла
    return sb.toString();
}

private static void exists(String fileName) throws FileNotFoundException
{
    File file = new File(fileName);
    if (!file.exists()){
        throw new FileNotFoundException(file.getName());
    }
}

private static String fileName;
}

```

Для импорта данных из файлов JSON использовалась библиотека json-simple.  
 Для экспорта данных в XML использовался класс RS2DOM.

### **Примеры работы программы:**

Запуск:

```
Output - book_store (run) #2 X
run:
Подключение к БД успешно выполнено.
-----
|                LIBRARY_                |
-----
функции:
1. Вывод списка всех таблиц
2. Добавление записи в таблицу LISTBOOKS
3. Импорт из CSV данных в LISTBOOK и LAB8
4. Экспорт содержимого таблицы в XML
5. Выход
Выберите пункт меню:
```

*1. Выполним первый пункт меню: Вывод списка всех таблиц.*

```
Output - book_store (run) #2 X
run:
Подключение к БД успешно выполнено.
-----
|                LIBRARY_                |
-----
функции:
1. Вывод списка всех таблиц
2. Добавление записи в таблицу LISTBOOKS
3. Импорт из CSV данных в LISTBOOK и LAB8
4. Экспорт содержимого таблицы в XML
5. Выход
Выберите пункт меню:
1
Список таблиц:
1. AUTHOR
2. BOOK
3. BUYER
4. CART
5. LAB7
6. LAB8
7. LIST_AUTHORS
8. LIST_BOOKS
9. LIST_CART
10. PUBLISHER
Введите номер таблицы для отображения ее содержимого или
0 для возврата в основное меню:
|
```

## Выберем для вывода содержимого таблицу 2

Output-book\_store (run) #2 X

10. PUBLISHER

Введите номер таблицы для отображения ее содержимого или  
0 для возврата в основное меню:

2

ID_BOOK	ID_PUBLISHER	NAME_BOOK	KOLICHESTVO	PRICE	YEAR_OF_PUBLISHER	ID_LISTBOOK	PAGE_NUMBER	RATING_BOOK			
B001	PUB001	MICROCONTROLLER	500	1500.0	1991-08-09	LB001	300	10			
B002	PUB001	MATH	500	1200.0	2011-10-09	LB001	200	8			
B003	PUB002	HISTORY IN VIET NAM	200	600.0	2011-10-06	LB003	100	7			
B004	PUB003	HARRY AND POSTER	100	1200.0	2015-05-10	LB002	500	9			
B005	PUB005	TOM AND JERRY	1000	2000.0	2016-01-07	LB002	600	6			
B006	PUB005	LANGUAGE	1000	800.0	2015-10-10	LB010	500	5			
B007	PUB005	ST.PETERSBURG CITY	800	400.0	2014-12-12	LB008	300	7			
B008	PUB003	TAT DEN	600	1200.0	2009-09-18	LB008	800	8			
B009	PUB003	DE MEN PHIEU LUU KY	1500	1300.0	2007-12-20	LB002	700	2			
B010	PUB004	WE ARE ONE	200	400.0	2011-04-09	LB002	300	4			
B011	PUB006	EVERY DAY I LOVE U	350	600.0	2014-02-25	LB006	20	7			
B012	PUB006	GIAI TOAN VAT LY 10	900	600.0	2012-12-22	LB001	260	9			
B013	PUB004	ECONOMIC RUSSIA	1500	320.0	2010-04-20	LB004	90	3			

-----

| LIBRARY |

-----

#функции:

1. Вывод списка всех таблиц
2. Добавление записи в таблицу LISTBOOKS
3. Импорт из CSV данных в LISTBOOK и LABS
4. Экспорт содержимого таблицы в XML
5. Выход

Выберите пункт меню:

## 2. Добавление записи в таблицу LAB8:

0

-----

| LIBRARY |

-----

#функции:

1. Вывод списка всех таблиц
2. Добавление записи в таблицу LAB8
3. Импорт из JSON данных в LISTBOOK и LABS
4. Экспорт содержимого таблицы в XML
5. Выход

Выберите пункт меню:

2

Введите ID\_LAB:

1001

Введите название LAB8:

LA8888888888

Запись добавлена в таблицу.

-----

## Отобразим содержимое таблицы LAB8:

```

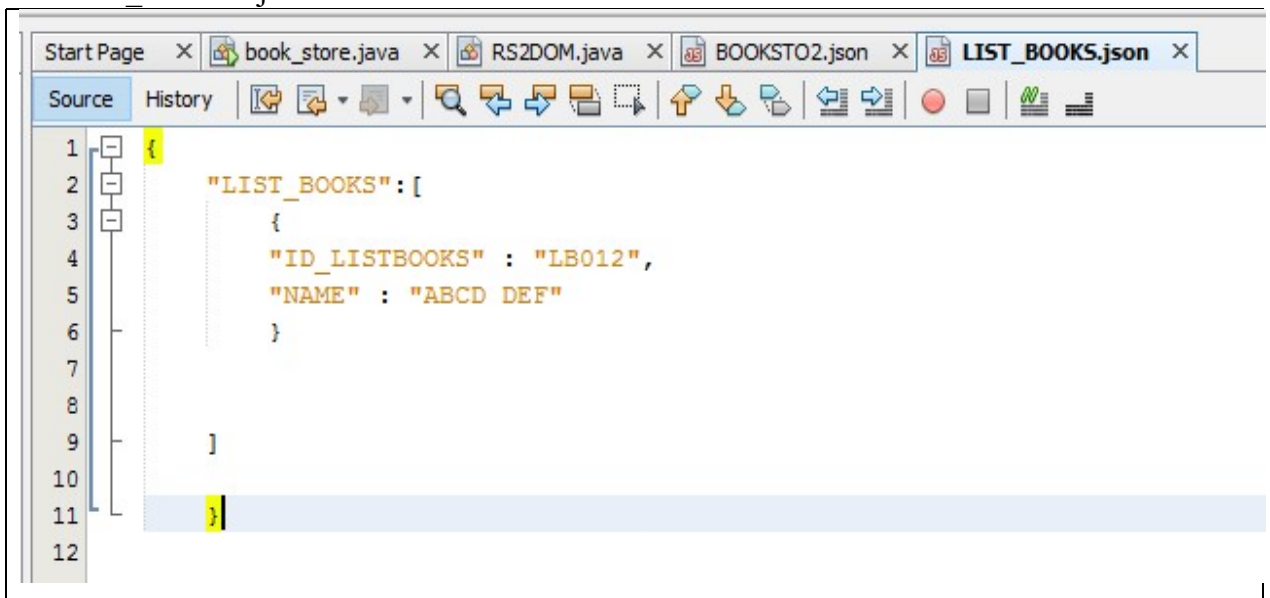
-----
|               LIBRARY_               |
-----
#функции:
1. Вывод списка всех таблиц
2. Добавление записи в таблицу LAB8
3. Импорт из JSON данных в LISTBOOK и LAB8
4. Экспорт содержимого таблицы в XML
5. Выход
Выберите пункт меню:
1
Список таблиц:
1. AUTHOR
2. BOOK
3. BUYER
4. CART
5. LAB7
6. LAB8
7. LIST_AUTHORS
8. LIST_BOOKS
9. LIST_CART
10. PUBLISHER
Введите номер таблицы для отображения ее содержимого или
0 для возврата в основное меню:
6

ID_LAB | ID_NAME |
12      GFDGG
113     labbbbb8
1001    LAAAAAB 8888888888
-----

```

#### 4. Импорт из JSON данных в LISTBOOKS и AUTHOR

File LIST\_BOOKS.json:

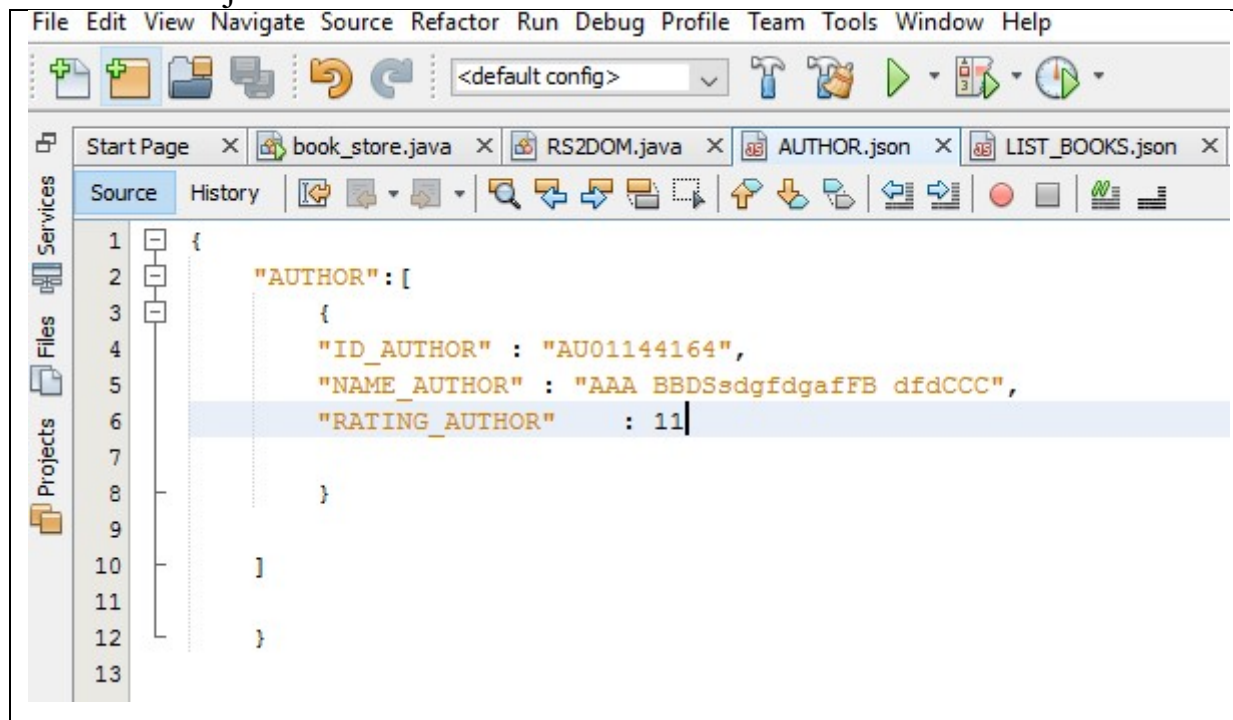


```

1 {
2   "LIST_BOOKS": [
3     {
4       "ID_LISTBOOKS" : "LB012",
5       "NAME" : "ABCD DEF"
6     }
7   ]
8 }
9
10
11
12

```

File AUTHOR.json:



Выполнение функции:

Перед добавлениями: LISTBOOKS

```
Output - book_store (run) #2 X
| LIBRARY_ |
-----
функции:
1. Вывод списка всех таблиц
2. Добавление записи в таблицу LAB8
3. Импорт из JSON данных в LISTBOOK и LAB8
4. Экспорт содержимого таблицы в XML
5. Выход
Выберите пункт меню:
1
Список таблиц:
1. AUTHOR
2. BOOK
3. BUYER
4. CART
5. LAB7
6. LAB8
7. LIST_AUTHORS
8. LIST_BOOKS
9. LIST_CART
10. PUBLISHER
Введите номер таблицы для отображения ее содержимого или
0 для возврата в основное меню:
8

ID_LISTBOOKS | NAME |
LB001        TECHNOLOGY
LB002        ROMANCE
LB003        HISTORY
LB004        ECONOMIC
LB005        SPORT
LB006        MUSIC
LB007        FANTASY
LB008        LECTURE
LB009        KUNGFU
LB010        LANGUAGE
LB0111       ABCD
-----
| LIBRARY_ |
```

Выполнение функции:  
Перед добавлениями: AUTHOR

```
Output - book_store (run) #2 X
#функции:
1. Вывод списка всех таблиц
2. Добавление записи в таблицу LAB8
3. Импорт из JSON данных в LISTBOOK и LAB8
4. Экспорт содержимого таблицы в XML
5. Выход
Выберите пункт меню:
1
Список таблиц:
1. AUTHOR
2. BOOK
3. BUYER
4. CART
5. LAB7
6. LAB8
7. LIST_AUTHORS
8. LIST_BOOKS
9. LIST_CART
10. PUBLISHER
Введите номер таблицы для отображения ее содержимого или
0 для возврата в основное меню:
1

ID_AUTHOR | NAME_AUTHOR | RATING_AUTHOR |
AU001     | DANG KHANH  | 10
AU002     | NGUYEN TIEN VU | 4
AU003     | BUI ANH TUAN | 8
AU004     | DAO XUAN HOA | 6
AU005     | TRAN VAN QUYET | 5
AU006     | TRINH THANH NAM | 3
AU007     | PHAM CONG MINH | 4
AU008     | NGUYEN VAN THANG | 1
AU009     | NGUYEN VAN TUYEN | 2
AU010     | HO CHI MINH | 9
AU011     | NGO TAT TO | 9
AU012     | NGUYEN DU | 8
AU013     | HOANG HOA THAM | 9
-----
| LIBRARY_ |
```

Добавления:



```
Output ×
book_store (run) #2 × book_store (run) #3 ×

run:
Подключение к БД успешно выполнено.
-----
|              LIBRARY_              |
-----

#функции:
1.Вывод списка всех таблиц
2.Добавление записи в таблицу LAB8
3.Импорт из JSON данных в LISTBOOK и LAB8
4.Экспорт содержимого таблицы в XML
5.Выход
Выберите пункт меню:
3

Импорт данных из JSON файлов в таблицы LIST_BOOK и LAB8 выполнен.
-----
|              LIBRARY_              |
-----

#функции:
1.Вывод списка всех таблиц
2.Добавление записи в таблицу LAB8
3.Импорт из JSON данных в LISTBOOK и LAB8
4.Экспорт содержимого таблицы в XML
5.Выход
Выберите пункт меню:
|
```

После добавления:  
LISTBOOKS



```
Output X
book_store (run) #2 x book_store (run) #3 x
3.Импорт из JSON данных в LISTBOOK и LAB8
4.Экспорт содержимого таблицы в XML
5.Выход
Выберите пункт меню:
1
Список таблиц:
1. AUTHOR
2. BOOK
3. BUYER
4. CART
5. LAB7
6. LAB8
7. LIST_AUTHORS
8. LIST_BOOKS
9. LIST_CART
10. PUBLISHER
Введите номер таблицы для отображения ее содержимого или
0 для возврата в основное меню:
8

ID_LISTBOOKS | NAME |
LB001        TECHNOLOGY
LB002        ROMANCE
LB003        HISTORY
LB004        ECONOMIC
LB005        SPORT
LB006        MUSIC
LB007        FANTASY
LB008        LECTURE
LB009        KUNGFU
LB010        LANGUAGE
LB0111       ABCD
LB012        ABCD DEF
-----
|                LIBRARY_                |
-----
```

AUTHOR:

```

Output x
book_store (run) x book_store (run) #2 x book_store (run) #3 x
3.Импорт из JSON данных в LISTBOOK и AUTHOR
4.Экспорт содержимого таблицы в XML
5.Выход
Выберите пункт меню:
1
Список таблиц:
1. AUTHOR
2. BOOK
3. BUYER
4. CART
5. LAB7
6. LAB8
7. LIST_AUTHORS
8. LIST_BOOKS
9. LIST_CART
10. PUBLISHER
Введите номер таблицы для отображения ее содержимого или
0 для возврата в основное меню:
1

ID_AUTHOR | NAME_AUTHOR | RATING_AUTHOR |
AU001      DANG KHANH   10
AU002      NGUYEN TIEN VU 4
AU003      BUI ANH TUAN  8
AU004      DAO XUAN HOA   6
AU005      TRAN VAN QUYET 5
AU006      TRINH THANH NAM 3
AU007      PHAM CONG MINH 4
AU008      NGUYEN VAN THANG 1
AU009      NGUYEN VAN TUYEN 2
AU010      HO CHI MINH    9
AU011      NGO TAT TO     9
AU012      NGUYEN DU      8
AU013      HOANG HOA THAM 9
AU014464   AAA BBDSsdgfdgafFB fdCCC 9
AU0114464   AAA BBDSsdgfdgafFB dfdCCC 19
AU01144164   AAA BBDSsdgfdgafFB dfdCCC 11
-----
| LIBRARY_ |

```

## 5. Экспорт содержимого таблицы в XML

```
Output
book_store (run) x book_store (run) #2 x book_store (run) #3 x book_store (run) #4 x

run:
Подключение к БД успешно выполнено.
-----
|                LIBRARY_                |
-----

Функции:
1. Вывод списка всех таблиц
2. Добавление записи в таблицу LAB8
3. Импорт из JSON данных в LISTBOOK и AUTHOR
4. Экспорт содержимого таблицы в XML
5. Выход
Выберите пункт меню:
4
Список таблиц:
1. AUTHOR
2. BOOK
3. BUYER
4. CART
5. LAB7
6. LAB8
7. LIST_AUTHORS
8. LIST_BOOKS
9. LIST_CART
10. PUBLISHER
Введите номер таблицы для экспорта в XML:
|
```

6. Выход:

```

Введите номер таблицы для экспорта в XML:
1

Схема, описывающая XML, экспортирована в файл Description.xml

Содержимое таблицы экспортировано в XML файл Data.xml
-----
|                LIBRARY_                |
-----
функции:
1.Вывод списка всех таблиц
2.Добавление записи в таблицу LABS
3.Импорт из JSON данных в LISTBOOK и AUTHOR
4.Экспорт содержимого таблицы в XML
5.Выход
Выберите пункт меню:
5
ББ
BUILD SUCCESSFUL (total time: 1 minute 0 seconds)
|

```

#### 4. Вывод

В результате выполнения работы было разработано клиентское приложение, осуществляющее некоторые функции для работы с нашей БД.

JDBC (Java DataBase Connectivity) – это платформенно - независимый промышленный стандарт взаимодействия Java-приложений с различными СУБД, реализованный в виде пакета java.sql, входящего в состав Java SE.

Преимуществами JDBC являются:

- 1) Сочетание JAVA API и JDBC API делает создание приложений лёгким и эффективным.
- 2) Код приложения подвержен наименьшим изменениям в случае, если происходит смена базы данных.
- 3) Лёгкость подсоединения к базе через легко описываемый URL.
- 4) JDBC API полностью предоставляет доступ к метаданным, что позволяет писать сложные приложения
- 5) Нет необходимости установки специального программного обеспечения, как на стороне клиента, так и на стороне сервера.
- 6) Драйвера JDBC могут загружаться динамически.