Санкт-Петербургский политехнический университет Петра Великого

Кафедра компьютерных систем и программных технологий

**Отчёт по лабораторной работе №2**

**Дисциплина**: Базы данных

**GUI-приложение**

Выполнил студент гр. 43501/3 Муравьев Ф.Э.

Преподаватель: Мяснов А. В.

Санкт-Петербург

2016

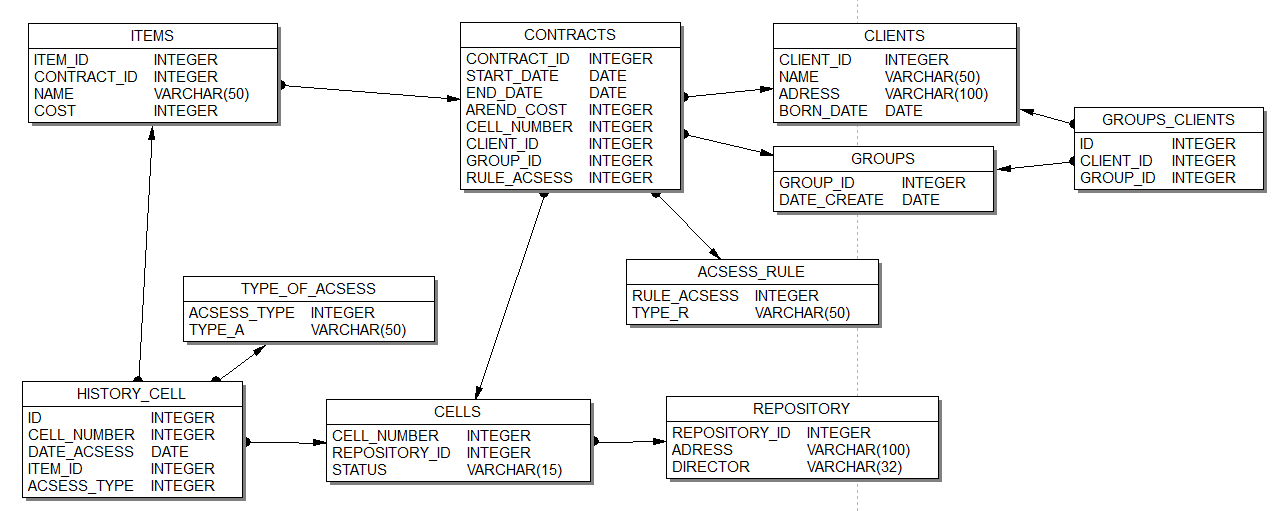
**Программа работы**

Варианты приложения:

1. Консольное приложение
   * импорт и экспорт в xml нескольких таблиц
   * проверка с помощью схемы
2. **Графическое приложение с CRUD (технология Swing) для 3-4 таблиц**
   * **master-detail с редактированием**
   * **отчет с использованием запроса**
   * **запуск процедур и получение результатов**
3. Веб-приложение на JSP
   * master-detail с редактированием
   * отчет с использованием запроса
   * запуск процедур и получение результатов
4. Веб-приложение на XML + XSLT
   * master-detail
   * отчет с использованием запроса
   * запуск процедур и получение результатов

**Ход работы**

Ниже приведена схема БД для которой выполнялось GUI-приложение



Подключение к БД осуществляется следующим образом:

|  |
| --- |
| public String DB\_URL = "jdbc:firebirdsql://localhost:3050/C:/DataBase/Test.FDB";  public String DB\_DEFAULT\_USER = "SYSDBA";  public String DB\_DEFAULT\_PASSWORD = "masterkey";  public String DB\_DEFAULT\_ENCODING = "win1251";  public Connection conn = null;  Properties props = null;  …  Class.forName("org.firebirdsql.jdbc.FBDriver");  props = new Properties();  props.setProperty("user", DB\_DEFAULT\_USER);  props.setProperty("password", DB\_DEFAULT\_PASSWORD);  props.setProperty("encoding", DB\_DEFAULT\_ENCODING);  …  private void DBConnect()  {  try {  conn = DriverManager.getConnection(DB\_URL, props);  } catch (SQLException e) {  e.printStackTrace();  }  } |

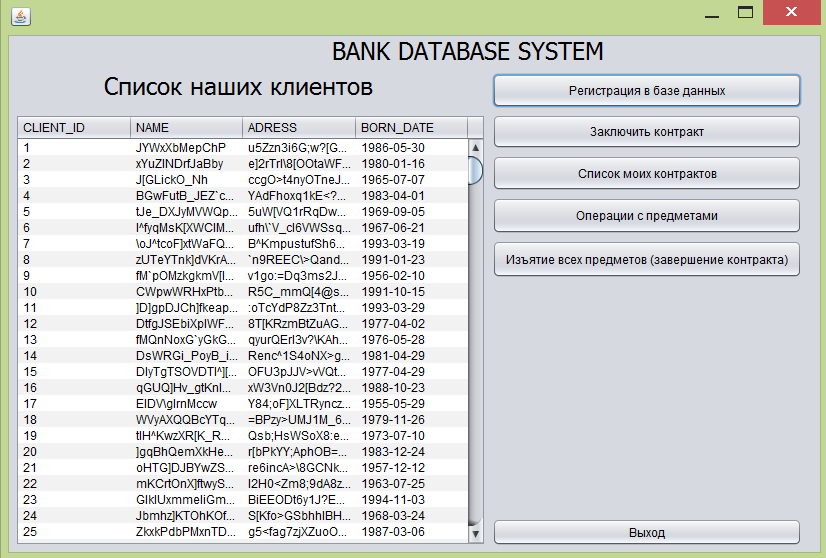


Рис. 1 Вид оконного приложения (Начальное окно)

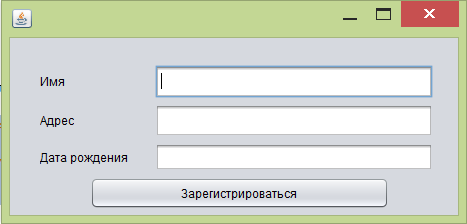


Рис. 2 Регистрация нового клиента

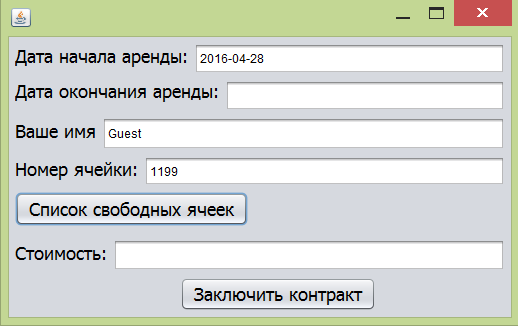


Рис. 3 Заключение нового контракта

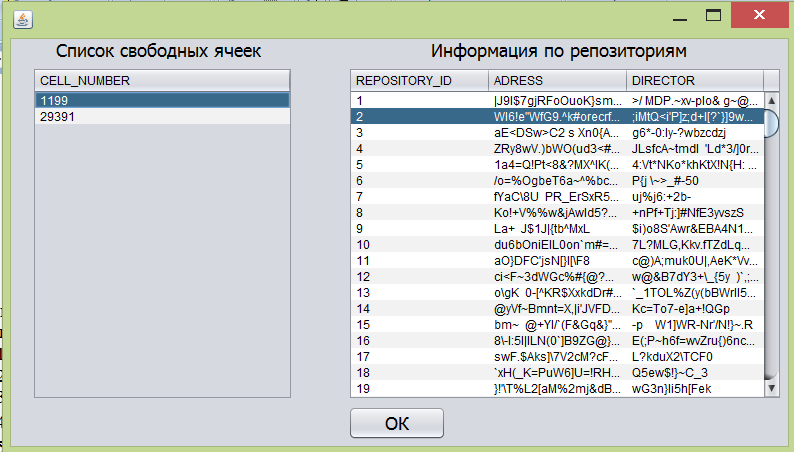


Рис. 4 Информация о свободных ячейках

Таблицы CELLS и HISTORY\_CELL связаны по принципу master-detail

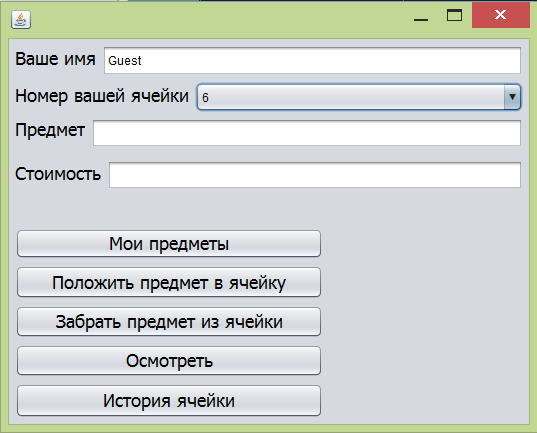


Рис. 5 Операции с предметами в ячейке

Данное приложение написано на java с использованием JDBC библиотек. Реализована следующая функциональность:

1. Подключение к БД.
2. Возможность просматривать список клиентов
3. Возможность регистрации нового клиента
4. Возможность просмотра списка контрактов для заданного клиента
5. Возможность завершения контракта(изъятие всех предметов из ячейки)
6. Возможность производить осмотр, добавление, изъятие предмета из ячейки
7. Возможность просмотра содержимого ячейки, а также ее истории по контракту
8. Возможность просмотра списка не занятых ячеек

Процесс работы клиента начинается с того, что он находит себя в списке клиентов. После этого клиент может проводит все доступные операции: заключение договора аренды, его завершение, различные операции с предметами в ячейке.

Пример функции отображения списка клиентов:

|  |
| --- |
| public void ClientTable()  {  try {  String strSQL = "select \* from CLIENTS";  if (conn==null)  {  JOptionPane.showMessageDialog(null, "Could not connect to database");  }  PreparedStatement statement = conn.prepareStatement(strSQL);  ResultSet resultSet = statement.executeQuery();  Client\_Table.setModel(DbUtils.resultSetToTableModel(resultSet));    }  catch(Exception e) {  JOptionPane.showMessageDialog(null, e);  }  } |

Пример функции отображения истории ячейки:

|  |
| --- |
| private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {  String Cell\_number, Contract\_id, Start\_date, End\_date;  try{  Contract\_id = Contract\_ID.getText();    String Query = "SELECT CELL\_NUMBER, CLIENT\_ID, START\_DATE, END\_DATE FROM CONTRACTS WHERE CONTRACT\_ID="+Contract\_id;  PreparedStatement statement = conn.prepareStatement(Query);  ResultSet resultSet = statement.executeQuery();  if(!resultSet.next())  {  JOptionPane.showMessageDialog(null, "Не существует данного контракта.");  return;  }  Cell\_number=resultSet.getString("CELL\_NUMBER");  Start\_date=resultSet.getString("START\_DATE");  End\_date=resultSet.getString("END\_DATE");  Query = "SELECT ID, CELL\_NUMBER, ACSESS\_TYPE, DATE\_ACSESS, HISTORY\_CELL.ITEM\_ID, NAME, COST FROM HISTORY\_CELL, ITEMS "  + "WHERE (CELL\_NUMBER="+Cell\_number+")AND(HISTORY\_CELL.ITEM\_ID=ITEMS.ITEM\_ID)AND(DATE\_ACSESS>\'"+Start\_date+"\')AND(DATE\_ACSESS<\'"+End\_date+"\')";  statement = conn.prepareStatement(Query);  resultSet = statement.executeQuery();  JTable t = new JTable(DbUtils.resultSetToTableModel(resultSet));  HistoryList list=new HistoryList(t);  list.setVisible(true);  } catch (SQLException e) {  JOptionPane.showMessageDialog(null, e);  }    } |

Пример функции создания нового контракта:

|  |
| --- |
| private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  String sql = "select max(Contract\_id)+1 as id from contracts";  try{  PreparedStatement pst = conn.prepareStatement(sql);  ResultSet rs = pst.executeQuery();  String h = null;  if(rs.next())  {  h = rs.getString("id").toString();  }    String strSQL = "Insert into Contracts (CONTRACT\_ID, \"START\_DATE\", \"END\_DATE\", AREND\_COST, CELL\_NUMBER, CLIENT\_ID, GROUP\_ID, RULE\_ACSESS) values (?,?,?,?,?,?,?,?)";  pst = conn.prepareStatement(strSQL);  pst.setString(1, h);  pst.setString(2, StartDate.getText());  pst.setString(3, End\_Date.getText());  pst.setString(4, Cost.getText());  pst.setString(5, Cell\_Number.getText());  pst.setString(6, Client\_id.getText());  pst.setString(7, "1");  pst.setString(8, "0");  pst.execute();    strSQL="UPDATE CELLS SET STATUS=\'arend\' WHERE CELL\_NUMBER="+Cell\_Number.getText();  pst = conn.prepareStatement(strSQL);  pst.execute();    this.setVisible(false);  JOptionPane.showMessageDialog(null, "Ваш уникальный идентификатор контракта: "+h);  }  catch(Exception e) {  JOptionPane.showMessageDialog(null, e);  }  } |

**Вывод**

В результате было реализовано GUI-приложение для работы с БД. Данное приложение позволяет добавлять новые данные, удалять и изменять уже имеющиеся данные (добавление, изъятие осмотр предметов, завершение контракта). Реализован удобный интерфейс для работы пользователя.

В ходе выполнения работы был приобретен навык подключения и непосредственной работы с БД, используя язык Java.