Министерство образования и науки Российской Федерации

**Федеральное государственное бюджетное образовательное учреждение науки и высшего образования**

«Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых»

(ВлГУ)

**Институт информационных технологий и радиотехники**

**(ИИТР)**

Кафедра информационных систем и программной инженерии

**Лабораторная работа № 02**

**по дисциплине**

**«Распределённые программные системы»**

**Тема «Реализация**

**шаблона проектирования Data Access Object.»**

Выполнил:

ст. гр. ПРИ-117

Хлызова В.Г.

Принял:

Трифонов Д.А.

Владимир, 2019

ЦЕЛЬ РАБОТЫ

Получить навыки практического использования средства JDBC API для реализации объектов доступа к данным (DAO).

ВЫПОЛНЕНИЕ РАБОТЫ

КЛАСС GROOUP:

**package** modelDao;  
  
**import** java.util.Date;  
  
**public class** Grooup {  
 **private int id**;  
 **private** String **groupName**;  
 **private** Date **creationDate**;  
 **private** String **curatorName**;  
  
 **public** Grooup(){}  
  
 **public** Grooup(**int** id, String groupName, Date creationDate, String curatorName){  
 **this**.**id** = id;  
 **this**.**groupName** = groupName;  
 **this**.**creationDate** = creationDate;  
 **this**.**curatorName** = curatorName;  
 }  
 **public** Grooup(String groupName, Date creationDate, String curatorName){  
 **this**.**groupName** = groupName;  
 **this**.**creationDate** = creationDate;  
 **this**.**curatorName** = curatorName;  
 }  
  
 **public int** getId() {  
 **return id**;  
 }  
  
 **public void** setId(**int** id) {  
 **this**.**id** = id;  
 }  
  
 **public** String getGroupName() {  
 **return groupName**;  
 }  
  
 **public void** setGroupName(String groupName) {  
 **this**.**groupName** = groupName;  
 }  
  
 **public** Date getCreationDate() {  
 **return creationDate**;  
 }  
  
 **public void** setCreationDate(Date creationDate) {  
 **this**.**creationDate** = creationDate;  
 }  
  
 **public** String getCuratorName() {  
 **return curatorName**;  
 }  
  
 **public void** setCuratorName(String curatorName) {  
 **this**.**curatorName** = curatorName;  
 }  
}

КЛАСС STUDENT:

**package** modelDao;  
  
**import** java.util.Date;  
  
**public class** Student{  
 **private int id**;  
 **private int grooupeId**;  
 **private** String **lastName**;  
 **private** String **firstName**;  
 **private** Date **birthDate**;  
 **private int age**;  
  
 **public** Student(){};  
  
 **public** Student(**int** id, **int** grooupeId, String lastName,String firstName, Date birthDate, **int** age){  
 **this**.**id**= id;  
 **this**.**grooupeId** = grooupeId;  
 **this**.**lastName** = lastName;  
 **this**.**firstName** = firstName;  
 **this**.**birthDate** = birthDate;  
 **this**.**age** = age;  
 }  
  
 **public** Student(**int** grooupeId, String lastName,String firstName, Date birthDate, **int** age){  
 **this**.**grooupeId** = grooupeId;  
 **this**.**lastName** = lastName;  
 **this**.**firstName** = firstName;  
 **this**.**birthDate** = birthDate;  
 **this**.**age** = age;  
 }  
  
 **public int** getId() {  
 **return id**;  
 }  
  
 **public void** setId(**int** id) {  
 **this**.**id** = id;  
 }  
  
 **public int** getGrooupeId() {  
 **return grooupeId**;  
 }  
  
 **public void** setGrooupeId(**int** grooupeId) {  
 **this**.**grooupeId** = grooupeId;  
 }  
  
 **public** String getLastName() {  
 **return lastName**;  
 }  
  
 **public void** setLastName(String lastName) {  
 **this**.**lastName** = lastName;  
 }  
  
 **public** String getFirstName() {  
 **return firstName**;  
 }  
  
 **public void** setFirstName(String firstName) {  
 **this**.**firstName** = firstName;  
 }  
  
 **public** Date getBirthDate() {  
 **return birthDate**;  
 }  
  
 **public void** setBirthDate(Date birthDate) {  
 **this**.**birthDate** = birthDate;  
 }  
  
 **public int** getAge() {  
 **return age**;  
 }  
  
 **public void** setAge(**int** age) {  
 **this**.**age** = age;  
 }  
}

ИНТЕРФЕЙС DAO:

**package** dao;  
  
**import** java.util.ArrayList;  
  
**public interface** DAO<T> {  
 **public abstract boolean** insert(T obj);  
 **public abstract** T getById(**int** id);  
 **public abstract boolean** update(T obj);  
 **public abstract boolean** delete(**int** id);  
 **public abstract** ArrayList<T> getAll();  
}

КЛАСС GROOUPEDAO:

**package** dao;  
  
**import** modelDao.Grooup;  
  
**import** java.sql.\*;  
**import** java.util.ArrayList;  
  
**public class** GrooupeDAO **implements** DAO<Grooup>{  
  
 **private static** String *url* = **"jdbc:mysql://localhost:3306/lab02?useSSL=false&useUnicode=true&"** +  
 **"useJDBCCompliantTimezoneShift=true&useLegacyDatetimeCode=false&serverTimezone=UTC"**;  
 **private static** String *username* = **"root"**;  
 **private static** String *password* = **"root"**;  
 **private static** String *query*;  
  
 @Override  
 **public boolean** insert(Grooup obj) {  
 *query* = **"insert into Grooup (groupName, creationDate, curatorName) "** +  
 **"values (?, ?, ?);"**;  
 **try**(Connection con = DriverManager.*getConnection*(*url*, *username*, *password*)){  
 PreparedStatement pStmt = con.prepareStatement(*query*);  
 pStmt.setString(1, obj.getGroupName());  
 pStmt.setDate(2,**new** java.sql.Date(obj.getCreationDate().getTime()));  
 pStmt.setString(3,obj.getCuratorName());  
 **if** (pStmt.executeUpdate()>0) **return true**;  
 }**catch** (SQLException sqlEx){  
 sqlEx.printStackTrace();  
 }  
 **return false**;  
 }  
  
 @Override  
 **public** Grooup getById(**int** id) {  
 *query* = **"select \* from grooup where id="**+id+**";"**;  
  
 **try**(Connection con = DriverManager.*getConnection*(*url*, *username*, *password*);  
 Statement stmt = con.createStatement(); ResultSet rs = stmt.executeQuery(*query*);){  
 **while** (rs.next()){  
 **return new** Grooup(id, rs.getString(**"groupName"**),  
 rs.getDate(**"creationDate"**), rs.getString(**"curatorName"**));  
 }  
 }**catch** (SQLException e) {  
 e.printStackTrace();  
 }  
 **return null**;  
 }  
  
 @Override  
 **public boolean** update(Grooup obj) {  
 *query* = **"update grooup set CuratorName='"**+ obj.getCuratorName() + **"'where id="** + obj.getId() + **";"**;  
 **try** (Connection con = DriverManager.*getConnection*(*url*, *username*, *password*)) {  
 **if**(con.prepareStatement(*query*).executeUpdate()>0)**return true**;  
 }**catch** (SQLException sqlEx){  
 sqlEx.printStackTrace();  
 }  
 **return false**;  
 }  
  
 @Override  
 **public boolean** delete(**int** id) {  
 *query* = **"delete from grooup where id="** + id +**";"**;  
 **try** (Connection con = DriverManager.*getConnection*(*url*, *username*, *password*)) {  
 **if**(con.prepareStatement(*query*).executeUpdate()>0) **return true**;  
 }**catch** (SQLException sqlEx){  
 sqlEx.printStackTrace();  
 }  
 **return false**;  
 }  
  
 @Override  
 **public** ArrayList<Grooup> getAll() {  
 ArrayList<Grooup> grooupList = **new** ArrayList<>();  
 *query* = **"select\*from grooup;"**;  
  
 **try** (Connection con = DriverManager.*getConnection*(*url*, *username*, *password*);  
 Statement stmt = con.createStatement(); ResultSet rs = stmt.executeQuery(*query*);) {  
 **while** (rs.next()) {  
 Grooup grooup = **new** Grooup(rs.getInt(**"id"**), rs.getString(**"groupName"**),  
 rs.getDate(**"creationDate"**), rs.getString(**"curatorName"**));  
 grooupList.add(grooup);  
 }  
 } **catch** (SQLException e) {  
 e.printStackTrace();  
 }  
 **return** grooupList;  
 }  
  
}

КЛАСС STUDENTDAO:

**package** dao;  
  
**import** modelDao.Student;  
  
**import** java.sql.\*;  
**import** java.util.ArrayList;  
  
  
**public class** StudentDAO **implements** DAO<Student>{  
  
 **private static** String *url* = **"jdbc:mysql://localhost:3306/lab02?useSSL=false&useUnicode=true&"** +  
 **"useJDBCCompliantTimezoneShift=true&useLegacyDatetimeCode=false&serverTimezone=UTC"**;  
 **private static** String *username* = **"root"**;  
 **private static** String *password* = **"root"**;  
 **private static** String *query*;  
  
 @Override  
 **public boolean** insert(Student obj) {  
 *query* = **"insert into Student (grooupId, lastName, firstName, birthDate, age) "** +  
 **"values (?, ?, ?, ?, ?)"**;  
 **try**(Connection con = DriverManager.*getConnection*(*url*, *username*, *password*)){  
 PreparedStatement pStmt = con.prepareStatement(*query*);  
 pStmt.setInt(1, obj.getGrooupeId());  
 pStmt.setString(2,obj.getLastName());  
 pStmt.setString(3,obj.getFirstName());  
 pStmt.setDate(4,**new** java.sql.Date(obj.getBirthDate().getTime()));  
 pStmt.setInt(5,obj.getAge());  
 **if**(pStmt.executeUpdate()>0) **return true**;  
 }**catch** (SQLException sqlEx){  
 sqlEx.printStackTrace();  
 }  
 **return false**;  
 }  
  
 @Override  
 **public** Student getById(**int** id) {  
 *query* = **"select \* from student where id="**+id+**";"**;  
  
 **try**(Connection con = DriverManager.*getConnection*(*url*, *username*, *password*);  
 Statement stmt = con.createStatement(); ResultSet rs = stmt.executeQuery(*query*);){  
 **while** (rs.next()){  
 **return new** Student(id, rs.getInt(**"grooupId"**), rs.getString(**"lastname"**),  
 rs.getString(**"firstname"**), rs.getDate(**"birthDate"**), rs.getInt(**"age"**));  
 }  
 } **catch** (SQLException sqlEx){  
 sqlEx.printStackTrace();  
 }  
 **return null**;  
 }  
  
 @Override  
 **public boolean** update(Student obj) {  
 *query* = **"update student set grooupID="**+ obj.getGrooupeId()+**" where id="** + obj.getId()+**";"**;  
 **try** (Connection con = DriverManager.*getConnection*(*url*, *username*, *password*)) {  
 **if**(con.prepareStatement(*query*).executeUpdate()>0) **return true**;  
 }**catch** (SQLException sqlEx){  
 sqlEx.printStackTrace();  
 }  
 **return false**;  
 }  
  
 @Override  
 **public boolean** delete(**int** id) {  
 *query* = **"delete from student where id="**+id;  
 **try** (Connection con = DriverManager.*getConnection*(*url*, *username*, *password*)) {  
 **if**(con.prepareStatement(*query*).executeUpdate()>0) **return true**;  
 }**catch** (SQLException sqlEx){  
 sqlEx.printStackTrace();  
 }  
 **return false**;  
 }  
  
 @Override  
 **public** ArrayList<Student> getAll() {  
 ArrayList<Student> studentList= **new** ArrayList<>();  
 *query* = **"select\*from student;"**;  
  
 **try**(Connection con = DriverManager.*getConnection*(*url*, *username*, *password*);  
 Statement stmt = con.createStatement(); ResultSet rs = stmt.executeQuery(*query*);){  
 **while** (rs.next()){  
 Student student = **new** Student(rs.getInt(**"id"**), rs.getInt(**"grooupId"**), rs.getString(**"lastname"**),  
 rs.getString(**"firstname"**), rs.getDate(**"birthDate"**), rs.getInt(**"age"**));  
 studentList.add(student);  
 }  
 } **catch** (SQLException sqlEx){  
 sqlEx.printStackTrace();  
 }  
  
 **return** studentList;  
 }  
}

ТЕСТ GROOUPEDAOTEST:

**package** dao;  
  
**import** modelDao.Grooup;  
**import** modelDao.Student;  
**import** org.junit.Test;  
**import** org.junit.Assert;  
  
**import** java.text.ParseException;  
**import** java.text.SimpleDateFormat;  
**import** java.util.ArrayList;  
  
**import static** org.junit.Assert.\*;  
  
**public class** GrooupeDAOTest {  
  
 @Test  
 **public void** insert() **throws** ParseException {  
 System.***out***.println(**"Test insert"**);  
 String date = **"2018-01-09"**;  
 SimpleDateFormat format = **new** SimpleDateFormat(**"yyyy-MM-dd"**);  
 Grooup grooup = **new** Grooup(**"ВТ-118"**, format.parse(date), **"Привалов М.Е."**);  
  
 **boolean** expectation = **true**;  
 GrooupeDAO grooupeDAO = **new** GrooupeDAO();  
 **boolean** reality = grooupeDAO.insert(grooup);  
  
 *assertEquals*(**"Добавление записи не произошло"**,expectation,reality);  
 }  
  
 @Test  
 **public void** getById() {  
 System.***out***.println(**"Test getById"**);  
 **int** id = 1;  
 GrooupeDAO grooupeDAO = **new** GrooupeDAO();  
 Grooup grooup = grooupeDAO.getById(id);  
  
 *assertEquals*(**"Поиск записи по id провален"**,grooup.getGroupName(),**"ПРИ-117"**);  
 }  
  
 @Test  
 **public void** update() {  
 System.***out***.println(**"Test update"**);  
 Grooup grooup = **new** Grooup();  
 grooup.setCuratorName(**"Синичкин В.И."**);  
 grooup.setId(2);  
  
 **boolean** expectation = **true**;  
 GrooupeDAO grooupeDAO = **new** GrooupeDAO();  
 **boolean** reality = grooupeDAO.update(grooup);  
  
 *assertEquals*(**"Обновление записи не произошло"**,expectation,reality);  
 }  
  
 @Test  
 **public void** delete() {  
 System.***out***.println(**"Test delete"**);  
 **int** id = 4;  
 **boolean** expectation = **true**;  
 GrooupeDAO grooupeDAO = **new** GrooupeDAO();  
 **boolean** reality = grooupeDAO.delete(id);  
  
 *assertEquals*(**"Удаление записи не произошло"**,expectation,reality);  
  
 }  
  
 @Test  
 **public void** getAll() {  
 System.***out***.println(**"Test getAll"**);  
 GrooupeDAO grooupeDAO = **new** GrooupeDAO();  
 ArrayList<Grooup> grooupList= grooupeDAO.getAll();  
  
 **int** expectation = 3;  
 **int** reality = grooupList.size();  
  
 *assertEquals*(**"Все записи не были получениы"**,expectation,reality);  
  
 **for** (Grooup i : grooupList){  
 System.***out***.print(i.getId()+ **" "** +i.getGroupName()+ **" "** + i.getCuratorName() + **" "** + i.getCreationDate() + **" "** + i.getCreationDate());  
 System.***out***.println();  
 }  
 }  
}

ТЕСТ STUDENTDAOTEST:

**package** dao;  
  
**import** modelDao.Student;  
**import** org.junit.Test;  
  
**import** java.text.ParseException;  
**import** java.text.SimpleDateFormat;  
**import** java.util.ArrayList;  
  
**import static** org.junit.Assert.\*;  
  
**public class** StudentDAOTest {  
  
 @Test  
 **public void** insert() **throws** ParseException {  
 System.***out***.println(**"Test insert"**);  
 String date = **"1996-03-01"**;  
 SimpleDateFormat format = **new** SimpleDateFormat(**"yyyy-MM-dd"**);  
 Student student = **new** Student(3,**"Филипов"**,**"Илья"**, format.parse(date),21);  
  
 **boolean** expectation = **true**;  
 StudentDAO studentDAO = **new** StudentDAO();  
 **boolean** reality = studentDAO.insert(student);  
  
 *assertEquals*(**"Добавление записи не произошло"**,expectation,reality);  
 }  
  
 @Test  
 **public void** getById() {  
 System.***out***.println(**"Test getById"**);  
 **int** id = 1;  
 StudentDAO studentDAO = **new** StudentDAO();  
 Student student = studentDAO.getById(id);  
  
 *assertEquals*(**"Поиск записи по id провален"**,student.getLastName(),**"Михайлов"**);  
 System.***out***.println(student.getLastName()+**" "**+student.getFirstName()  
 +**" "**+student.getBirthDate()+**" "**+student.getAge()+ **" лет"**);  
 }  
  
 @Test  
 **public void** update() {  
 System.***out***.println(**"Test update"**);  
 Student student = **new** Student();  
 student.setGrooupeId(3);  
 student.setId(1);  
  
 **boolean** expectation = **true**;  
 StudentDAO studentDAO = **new** StudentDAO();  
 **boolean** reality = studentDAO.update(student);  
  
 *assertEquals*(**"Обновление записи не произошло"**,expectation,reality);  
 }  
  
 @Test  
 **public void** delete() {  
 System.***out***.println(**"Test delete"**);  
 **int** id = 4;  
 **boolean** expectation = **true**;  
 StudentDAO studentDAO = **new** StudentDAO();  
 **boolean** reality = studentDAO.delete(id);  
  
 *assertEquals*(**"Удаление записи не произошло"**,expectation,reality);  
 }  
  
 @Test  
 **public void** getAll() {  
 System.***out***.println(**"Test getAll"**);  
 StudentDAO studentDAO = **new** StudentDAO();  
 ArrayList<Student> studentList= studentDAO.getAll();  
  
 **int** expectation = 5;  
 **int** reality = studentList.size();  
  
 *assertEquals*(**"Все записи не были получениы"**,expectation,reality);  
  
 **for** (Student i : studentList){  
 System.***out***.print(i.getId()+ **" "** +i.getGrooupeId()+ **" "** + i.getLastName() + **" "** + i.getFirstName() + **" "** + i.getBirthDate());  
 System.***out***.println();  
 }  
 }  
}

ТЕСТИРОВАНИЕ

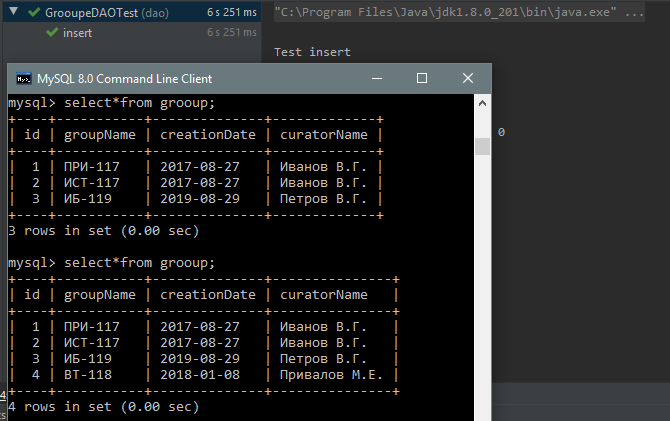


Рисунок 1. Test insert.

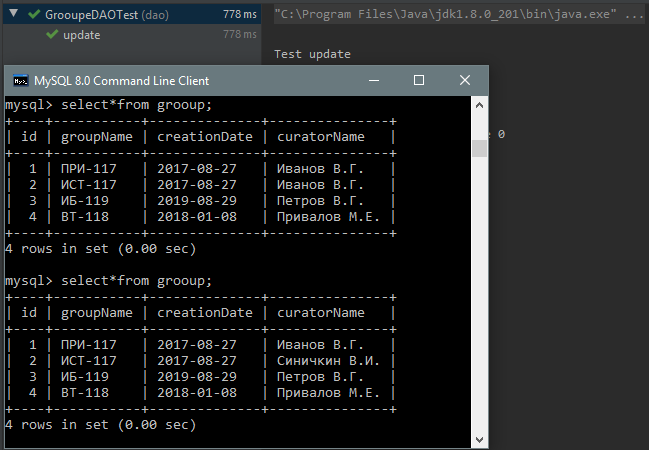


Рисунок 2. Test update.

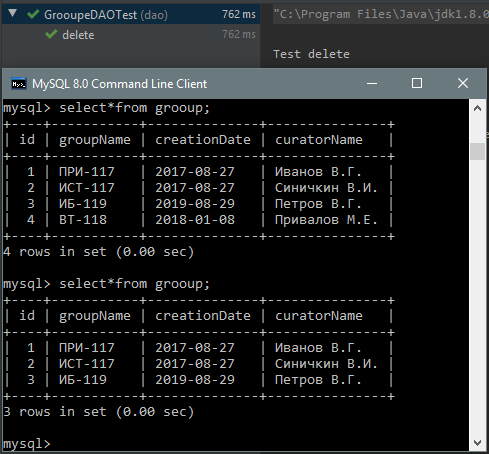


Рисунок 3. Test delete.

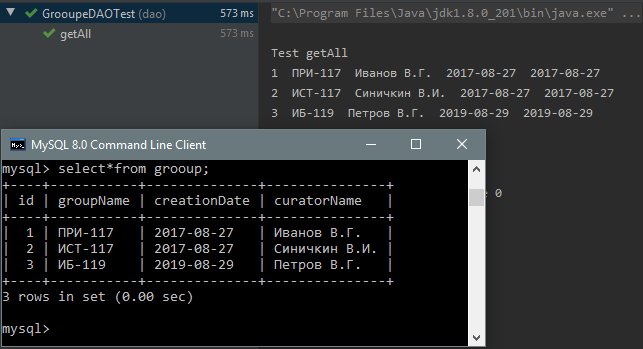


Рисунок 4. Test getAll.

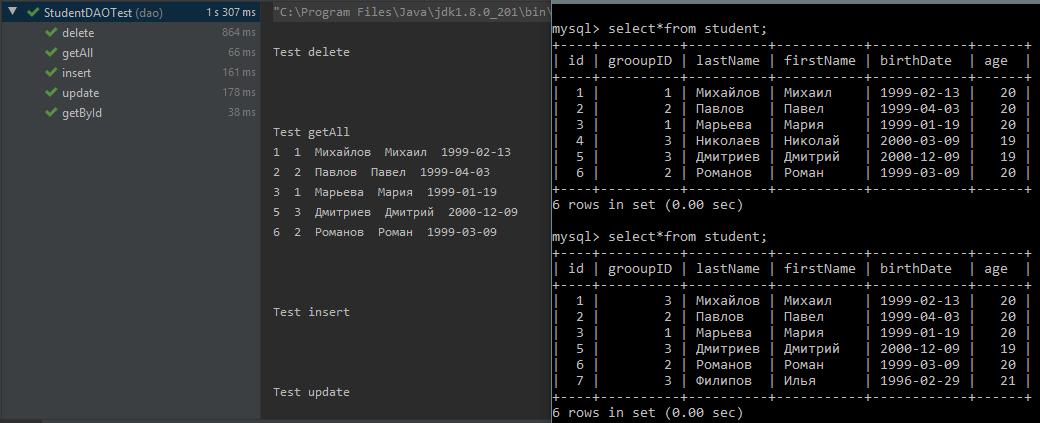


Рисунок 5. Выполнение всех тестов StudentDAOTest.

ВЫВОД

В ходе лабораторной работы были получены навыки практического использования средства JDBC API для реализации объектов доступа к данным (DAO).