Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет» Кафедра ИИТ

Лабораторная работа №10

По дисциплине «СПП» за 6 семестр

Выполнил:

Студент группы ПО-3

Ковалёва А. И.

Проверил:

Крощенко А. А.

Брест 2021

**Вариант 12**

**Цель**: приобрести практические навыки разработки многооконных приложений на JavaFX для работы с базами данных.

**Задание**

На основе БД, разработанной в лабораторной работе No9, реализовать многооконное приложение- клиент, позволяющее выполнять основные операции над таблицей в БД (добавление, удаление, мо- дификацию данных).12) База данных «Европейские футбольные чемпионаты»

**Текст программы:**

**Main**  
  
**package** sample;  
  
**import** javafx.application.Application;  
**import** javafx.fxml.FXMLLoader;  
**import** javafx.scene.Parent;  
**import** javafx.scene.Scene;  
**import** javafx.stage.Stage;  
  
**public class** Main **extends** Application {  
  
 @Override  
 **public void** start(Stage primaryStage) **throws** Exception{  
 Parent root = FXMLLoader.*load*(Main.**class**.getClassLoader().getResource(**"home.fxml"**));  
 primaryStage.setTitle(**"Lab 10"**);  
 primaryStage.setScene(**new** Scene(root, 1000, 450));  
 primaryStage.show();  
 }  
  
  
 **public static void** main(String[] args) {  
 launch(args);  
 }  
}

**DBConnection**

**package** sample;  
  
**import** java.sql.Connection;  
**import** java.sql.DriverManager;  
**import** java.sql.SQLException;  
  
**public class** DBConnection {  
  
 **static final** String ***URL*** = **"jdbc:h2:mem:test;INIT=runscript from 'src/main/resources/sql/schema1.sql'\\;"** +  
 **"runscript from 'src/main/resources/sql/data.sql'"**;  
  
 **static final** String ***USER*** = **"sa"**;  
 **static final** String ***PASS*** = **""**;  
  
 **public static** Connection *connection* = **null**;  
  
 **public static void** getConnection() **throws** SQLException {  
  
 **if**(*connection* == **null**)  
 *connection* = DriverManager.*getConnection*(***URL***, **"sa"**, **""**);  
  
 }  
}

**MatchFootball**

**package** sample;  
  
**public class** MatchFootball {  
  
 **int id**;  
 String **match\_name**, **comand**, **referee**, **stadion**, **stadion\_size**;  
  
 **public** MatchFootball(**int** id, String match\_name, String comand, String referee, String stadion, String stadion\_size) {  
 **this**.**id** = id;  
 **this**.**match\_name** = match\_name;  
 **this**.**comand** = comand;  
 **this**.**referee** = referee;  
 **this**.**stadion** = stadion;  
 **this**.**stadion\_size** = stadion\_size;  
 }  
  
 **public** MatchFootball(**int** id, String match\_name, String comand, String referee, String stadion) {  
 **this**.**id** = id;  
 **this**.**match\_name** = match\_name;  
 **this**.**comand** = comand;  
 **this**.**referee** = referee;  
 **this**.**stadion** = stadion;  
 }  
  
 @Override  
 **public** String toString() {  
 **return "MatchFootball{"** +  
 **"id="** + **id** +  
 **", match\_name='"** + **match\_name** + **'\''** +  
 **", comand='"** + **comand** + **'\''** +  
 **", referee='"** + **referee** + **'\''** +  
 **", stadion="** + **stadion** +  
 **'}'**;  
 }  
  
 **public int** getId() {  
 **return id**;  
 }  
  
 **public void** setId(**int** id) {  
 **this**.**id** = id;  
 }  
  
 **public** String getMatch\_name() {  
 **return match\_name**;  
 }  
  
 **public void** setMatch\_name(String match\_name) {  
 **this**.**match\_name** = match\_name;  
 }  
  
 **public** String getComand() {  
 **return comand**;  
 }  
  
 **public void** setComand(String comand) {  
 **this**.**comand** = comand;  
 }  
  
 **public** String getReferee() {  
 **return referee**;  
 }  
  
 **public void** setReferee(String referee) {  
 **this**.**referee** = referee;  
 }  
  
 **public** String getStadion() {  
 **return stadion**;  
 }  
  
 **public void** setStadion(String stadion) {  
 **this**.**stadion** = stadion;  
 }  
  
 **public** String getStadion\_size() {  
 **return stadion\_size**;  
 }  
  
 **public void** setStadion\_size(String stadion\_size) {  
 **this**.**stadion\_size** = stadion\_size;  
 }  
}

**MatchFootballDAO**

**package** sample;  
  
**import** java.sql.PreparedStatement;  
**import** java.sql.ResultSet;  
**import** java.sql.SQLException;  
**import** java.sql.Statement;  
**import** java.util.ArrayList;  
**import** java.util.List;  
  
**public class** MatchFootballDAO {  
  
 **public** List<MatchFootball> getAll(){  
 String sql = **"SELECT MF.ID, MF.MATCH\_NAME, R.NAME as TYPE, C.COMAND\_NAME, C.CITY, S.STADION\_NAME AS STADION, S.STADION\_SIZE, "** +  
 **"FROM MATCHFOOTBALL MF "** +  
 **"INNER JOIN REFEREE R ON MF.REFEREE = R.ID "** +  
 **"INNER JOIN STADION S ON MF.STADION = S.ID "** +  
 **"INNER JOIN COMAND C ON MF.COMAND = C.ID"**;  
  
 List<MatchFootball> list = **new** ArrayList<>();  
  
 **try** {  
 DBConnection.*getConnection*();  
 Statement statement = DBConnection.*connection*.createStatement();  
 ResultSet resultSet = statement.executeQuery(sql);  
  
 **while** (resultSet.next()){  
 System.***out***.println(resultSet.isLast());  
 list.add(**new** MatchFootball(  
 resultSet.getInt(**"ID"**),  
 resultSet.getString(**"MATCH\_NAME"**),  
 resultSet.getString(**"COMAND\_NAME"**),  
 resultSet.getString(**"TYPE"**),  
 resultSet.getString(**"STADION"**),  
 resultSet.getString(**"STADION\_SIZE"**)  
 ));  
 }  
 } **catch** (SQLException throwables) {  
 throwables.printStackTrace();  
 }  
  
 **return** list;  
 }  
  
 **public void** insertMatchFootball(MatchFootball mf){  
 String sql = **"insert into MatchFootball (match\_name, referee, "** +  
 **"stadion, comand) values ( ?, (SELECT id FROM REFEREE where NAME = ?),"** +  
 **"SELECT id FROM STADION where STADION\_NAME = ?,"** +  
 **"SELECT id FROM COMAND where COMAND\_NAME = ?)"**;  
  
 **try** {  
 DBConnection.*getConnection*();  
 PreparedStatement preparedStatement = DBConnection.*connection*.prepareStatement(sql);  
 preparedStatement.setString(1, mf.getMatch\_name());  
 preparedStatement.setString(2, mf.getComand());  
 preparedStatement.setString(3, mf.getReferee());  
 preparedStatement.setString(4, mf.getStadion());  
  
 System.***out***.println(**"1 - "**+mf.getMatch\_name()+ **" 2 - "**+mf.getReferee()+  
 **" 3 - "**+ mf.getStadion()+ **" 4 - "** + mf.getComand());  
 **int** rows = preparedStatement.executeUpdate();  
 System.***out***.printf(**"%d rows added"**, rows);  
  
 List<MatchFootball> list = getAll();  
 **for** (MatchFootball matchFootball :list) {  
 System.***out***.println(matchFootball.getComand());  
 }  
  
 } **catch** (SQLException throwables) {  
 throwables.printStackTrace();  
 }  
 }  
  
 **public void** delete(**int** id){  
 String sql = **"Delete from MATCHFOOTBALL where id = ?"**;  
 **try** {  
 DBConnection.*getConnection*();  
 PreparedStatement preparedStatement = DBConnection.*connection*.prepareStatement(sql);  
 preparedStatement.setInt(1, id);  
 preparedStatement.executeUpdate();  
  
 } **catch** (SQLException throwables) {  
 throwables.printStackTrace();  
 }  
 }  
  
 **public void** update(MatchFootball matchFootball, String str){  
 String sql = **"UPDATE MATCHFOOTBALL SET MATCH\_NAME = ? WHERE ID = ?"**;  
 **try** {  
 DBConnection.*getConnection*();  
 PreparedStatement preparedStatement = DBConnection.*connection*.prepareStatement(sql);  
 preparedStatement.setString(1, str);  
 preparedStatement.setString(2, Integer.*toString*(matchFootball.getId()));  
  
 **int** rows = preparedStatement.executeUpdate();  
 System.***out***.printf(**"%d rows upDate added"**, rows);  
  
 List<MatchFootball> list = getAll();  
  
 } **catch** (SQLException throwables) {  
 throwables.printStackTrace();  
 }  
  
 }  
  
 **public** List<String> getReferee(){  
 String sql = **"SELECT** *\** **FROM REFEREE"**;  
  
 List<String> list = **new** ArrayList<>();  
  
 **try** {  
 DBConnection.*getConnection*();  
 Statement statement = DBConnection.*connection*.createStatement();  
 ResultSet resultSet = statement.executeQuery(sql);  
  
 **while** (resultSet.next()){  
 list.add(resultSet.getString(**"name"**));  
 }  
 } **catch** (SQLException throwables) {  
 throwables.printStackTrace();  
 }  
  
 **return** list;  
 }  
  
 **public** List<String> getComand(){  
 String sql = **"SELECT** *\** **FROM COMAND"**;  
  
 List<String> list = **new** ArrayList<>();  
  
 **try** {  
 DBConnection.*getConnection*();  
 Statement statement = DBConnection.*connection*.createStatement();  
 ResultSet resultSet = statement.executeQuery(sql);  
  
 **while** (resultSet.next()){  
 list.add(resultSet.getString(**"comand\_name"**));  
 }  
 } **catch** (SQLException throwables) {  
 throwables.printStackTrace();  
 }  
  
 **return** list;  
 }  
  
 **public** List<String> getStadion(){  
 String sql = **"SELECT** *\** **FROM STADION"**;  
  
 List<String> list = **new** ArrayList<>();  
  
 **try** {  
 DBConnection.*getConnection*();  
 Statement statement = DBConnection.*connection*.createStatement();  
 ResultSet resultSet = statement.executeQuery(sql);  
  
 **while** (resultSet.next()){  
 list.add(resultSet.getString(**"stadion\_name"**));  
 }  
 } **catch** (SQLException throwables) {  
 throwables.printStackTrace();  
 }  
  
 **return** list;  
 }  
}

**Stadion**

**package** sample;  
  
**public class** Stadion {  
 **int id**;  
 String **stadion\_name**;  
 String **stadion\_size**;  
  
 **public** Stadion(String stadion\_name, String stadion\_size) {  
 **this**.**id** = **id**;  
 **this**.**stadion\_name** = stadion\_name;  
 **this**.**stadion\_size** = stadion\_size;  
 }  
  
 **public int** getId() {  
 **return id**;  
 }  
  
 **public void** setId(**int** id) {  
 **this**.**id** = id;  
 }  
  
 **public** String getStadion\_name() {  
 **return stadion\_name**;  
 }  
  
 **public void** setStadion\_name(String stadion\_name) {  
 **this**.**stadion\_name** = stadion\_name;  
 }  
  
 **public** String getStadion\_size() {  
 **return stadion\_size**;  
 }  
  
 **public void** setStadion\_size(String stadion\_size) {  
 **this**.**stadion\_size** = stadion\_size;  
 }  
}

**FormController**

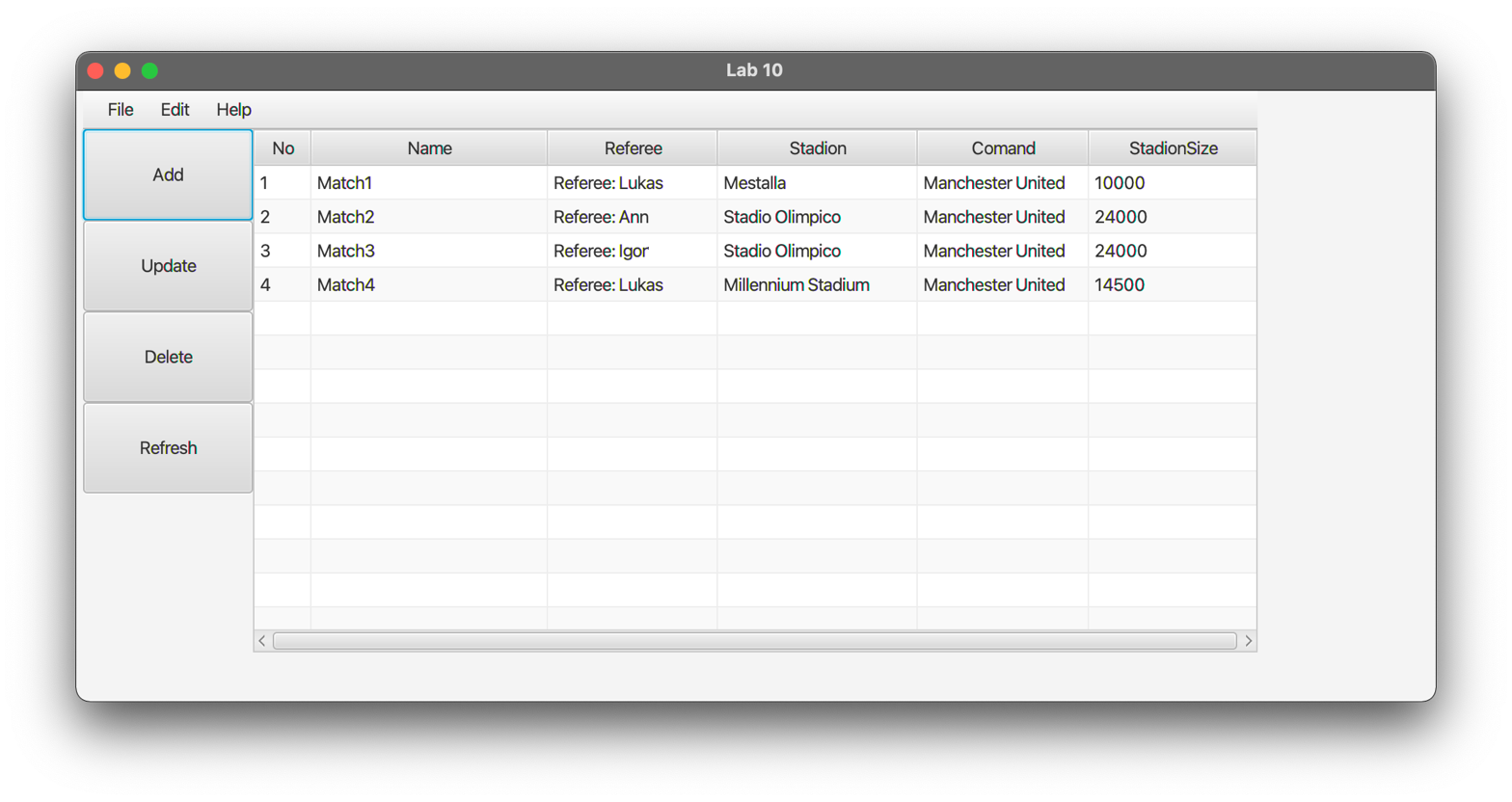
**package** sample.controller;  
  
**import** javafx.collections.FXCollections;  
**import** javafx.collections.ObservableList;  
**import** javafx.event.ActionEvent;  
**import** javafx.fxml.FXML;  
**import** javafx.scene.control.Button;  
**import** javafx.scene.control.ComboBox;  
**import** javafx.scene.control.TextField;  
**import** javafx.stage.Stage;  
**import** sample.MatchFootballDAO;  
**import** sample.MatchFootball;  
  
**public class** FormController{  
  
 @FXML  
 **private** TextField **id**;  
  
 @FXML  
 **private** TextField **match\_name**;  
  
 @FXML  
 **private** ComboBox<String> **comand**;  
  
 @FXML  
 **private** ComboBox<String> **stadion**;  
  
 @FXML  
 **private** ComboBox<String> **referee**;  
  
 @FXML  
 **private** Button **bottonOk**;  
  
 **private** MatchFootballDAO **dao** = **new** MatchFootballDAO();  
  
 **private static** HomeController *homeController*;  
  
 MatchFootball **matchFootball**;  
  
  
 @FXML  
 **void** initialize() {  
 **comand**.setItems(getComand());  
 **stadion**.setItems(getStadion());  
 **referee**.setItems(getReferee());  
 }  
  
 @FXML  
 **void** insert(ActionEvent event) {  
  
 **matchFootball** = **new** MatchFootball(  
 Integer.*parseInt*(**id**.getText()),  
 **match\_name**.getText(),  
 **referee**.getValue(),  
 **stadion**.getValue(),  
 **comand**.getValue()  
 );  
 **dao**.insertMatchFootball(**matchFootball**);  
 *homeController*.refresh();  
 Stage stage = (Stage) **bottonOk**.getScene().getWindow();  
 stage.close();  
  
 }  
  
 **public static void** setHomeController(HomeController home){  
 *homeController* = home;  
 }  
  
 ObservableList<String> getComand(){  
 ObservableList<String> disks = FXCollections.*observableArrayList*();  
 disks.addAll(**dao**.getComand());  
 **return** disks;  
 }  
  
 ObservableList<String> getStadion(){  
 ObservableList<String> disks = FXCollections.*observableArrayList*();  
 disks.addAll(**dao**.getStadion());  
 **return** disks;  
 }  
  
 ObservableList<String> getReferee(){  
 ObservableList<String> disks = FXCollections.*observableArrayList*();  
 disks.addAll(**dao**.getReferee());  
 **return** disks;  
 }  
}

**HomeController**

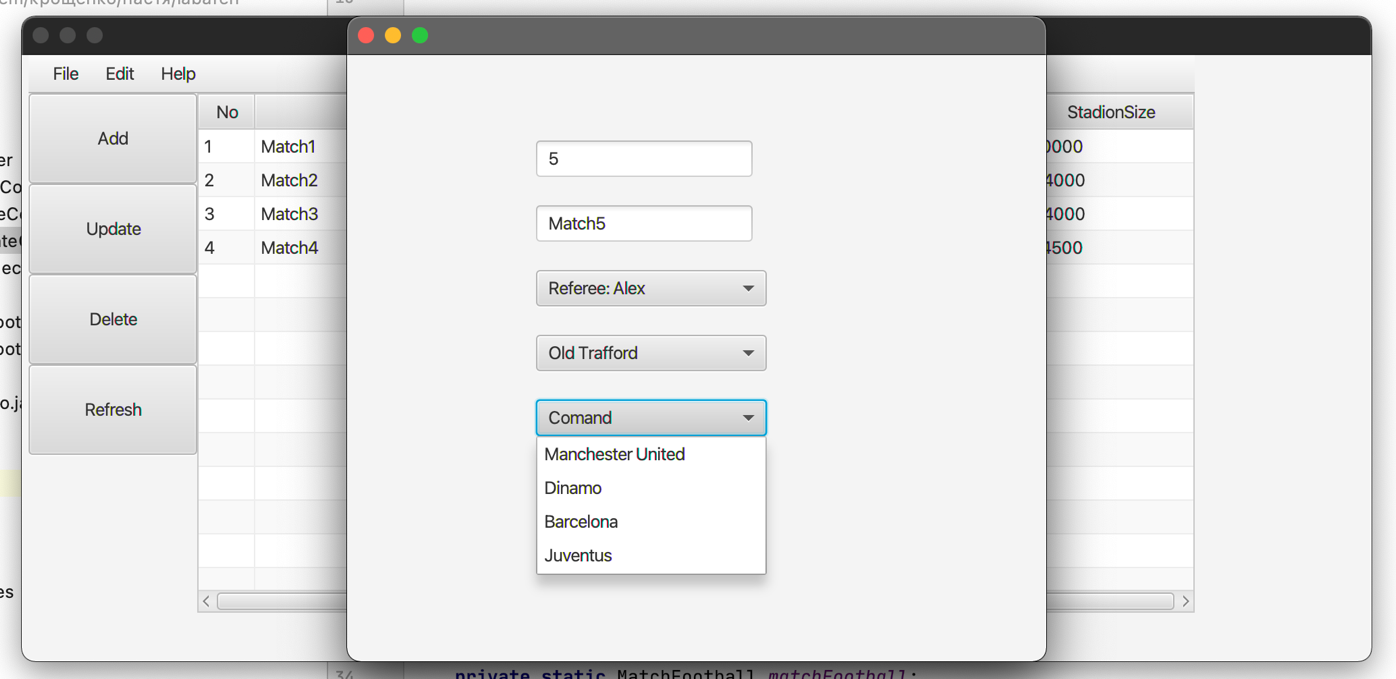
**package** sample.controller;  
  
**import** javafx.collections.FXCollections;  
**import** javafx.collections.ObservableList;  
**import** javafx.event.ActionEvent;  
**import** javafx.fxml.FXML;  
**import** javafx.fxml.FXMLLoader;  
**import** javafx.scene.Parent;  
**import** javafx.scene.Scene;  
**import** javafx.scene.control.TableColumn;  
**import** javafx.scene.control.TableView;  
**import** javafx.scene.control.cell.PropertyValueFactory;  
**import** javafx.stage.Stage;  
**import** sample.MatchFootballDAO;  
**import** sample.MatchFootball;  
  
**import** java.io.IOException;  
  
**public class** HomeController {  
  
 @FXML  
 **protected** TableView<MatchFootball> **table**;  
  
 @FXML  
 **private** TableColumn<MatchFootball, String> **id**;  
  
 @FXML  
 **private** TableColumn<MatchFootball, String> **match\_name**;  
  
 @FXML  
 **private** TableColumn<MatchFootball, String> **referee**;  
  
 @FXML  
 **private** TableColumn<MatchFootball, String> **stadion**;  
  
 @FXML  
 **private** TableColumn<MatchFootball, String> **comand**;  
  
 @FXML  
 **private** TableColumn<MatchFootball, String> **stadion\_size**;  
  
 ObservableList<MatchFootball> **listview** = FXCollections.*observableArrayList*();  
  
 **private** MatchFootballDAO **dao** = **new** MatchFootballDAO();  
  
 @FXML  
 **void** initialize() {  
  
 **id**.setCellValueFactory(**new** PropertyValueFactory<MatchFootball, String>(**"id"**));  
 **match\_name**.setCellValueFactory(**new** PropertyValueFactory<>(**"match\_name"**));  
 **referee**.setCellValueFactory(**new** PropertyValueFactory<>(**"referee"**));  
 **stadion**.setCellValueFactory(**new** PropertyValueFactory<>(**"stadion"**));  
 **comand**.setCellValueFactory(**new** PropertyValueFactory<>(**"comand"**));  
 **stadion\_size**.setCellValueFactory(**new** PropertyValueFactory<>(**"stadion\_size"**));  
  
 **listview**.addAll(**dao**.getAll());  
 **table**.setItems(**listview**);  
 }  
  
 @FXML  
 **void** add(ActionEvent event) {  
 **try** {  
 FormController.*setHomeController*(**this**);  
 FXMLLoader fxmlLoader = **new** FXMLLoader(getClass().getResource(**"/insert.fxml"**));  
 Parent root1 = (Parent) fxmlLoader.load();  
 Stage stage = **new** Stage();  
 stage.setScene(**new** Scene(root1));  
 stage.show();  
 } **catch** (IOException e) {  
 e.printStackTrace();  
 }  
  
 }  
  
 @FXML  
 **void** update(ActionEvent event) {  
 **try** {  
 UpdateController.*setMatchFootball*(**table**.getSelectionModel().getSelectedItem());  
 UpdateController.*setHomeController*(**this**);  
 FXMLLoader fxmlLoader = **new** FXMLLoader(getClass().getResource(**"/update.fxml"**));  
 Parent root1 = (Parent) fxmlLoader.load();  
 Stage stage = **new** Stage();  
 stage.setScene(**new** Scene(root1));  
 stage.show();  
 } **catch** (IOException e) {  
 e.printStackTrace();  
 }  
 }  
  
 @FXML  
 **public void** delete(ActionEvent event) {  
 MatchFootball matchFootball = **table**.getSelectionModel().getSelectedItem();  
 **dao**.delete(matchFootball.getId());  
 refresh();  
 }  
  
 **public void** refresh() {  
 **listview**.clear();  
 **listview**.addAll(**dao**.getAll());  
 **table**.setItems(**listview**);  
 }  
}

**UpdateController**

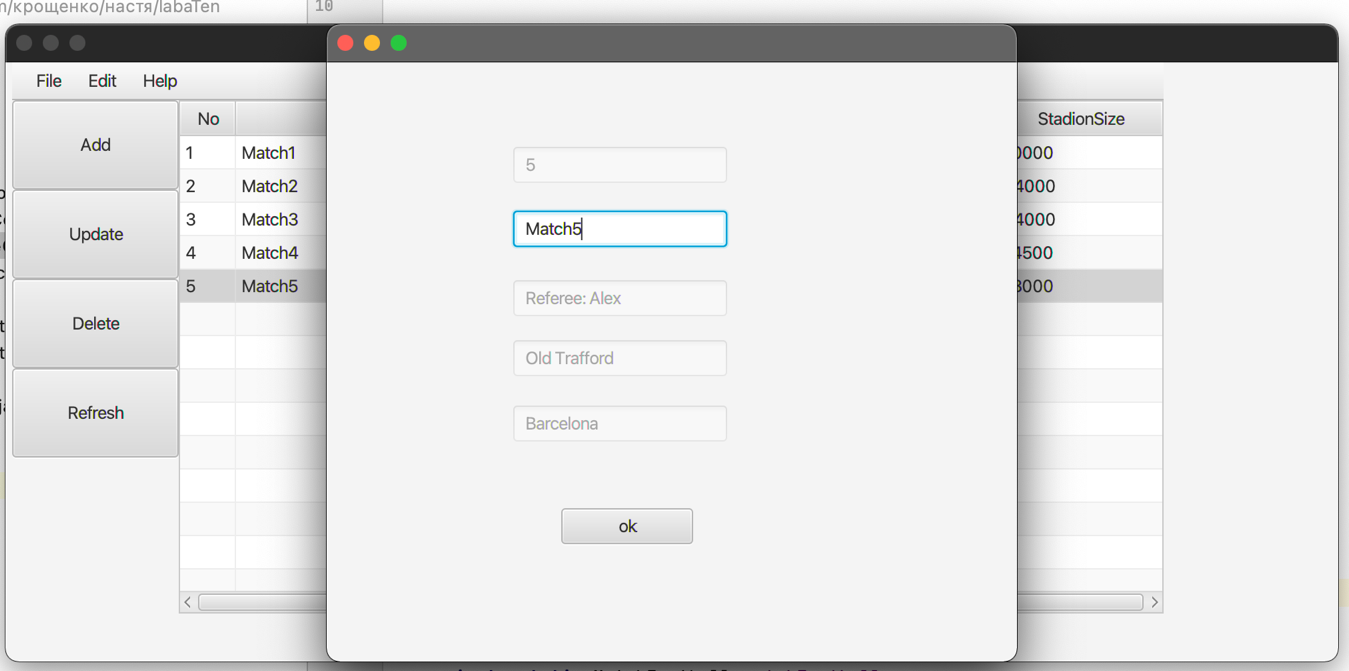
**package** sample.controller;  
  
**import** javafx.event.ActionEvent;  
**import** javafx.fxml.FXML;  
**import** javafx.scene.control.Button;  
**import** javafx.scene.control.TextField;  
**import** javafx.stage.Stage;  
**import** sample.MatchFootballDAO;  
**import** sample.MatchFootball;  
  
**public class** UpdateController {  
  
  
 @FXML  
 **private** TextField **id**;  
  
 @FXML  
 **private** TextField **match\_name**;  
  
 @FXML  
 **private** TextField **referee**;  
  
 @FXML  
 **private** TextField **stadion**;  
  
 @FXML  
 **private** TextField **comand**;  
  
 @FXML  
 **private** Button **buttonOk**;  
  
 **private** MatchFootballDAO **dao** = **new** MatchFootballDAO();  
  
 **private static** MatchFootball *matchFootball*;  
 **private static** HomeController *homeController*;  
  
 @FXML  
 **void** initialize() {  
  
 **id**.setText(Integer.*toString*(*matchFootball*.getId()));  
 **id**.setDisable(**true**);  
 **match\_name**.setText(*matchFootball*.getMatch\_name());  
 **referee**.setText(*matchFootball*.getReferee());  
 **referee**.setDisable(**true**);  
 **stadion**.setText(*matchFootball*.getStadion());  
 **stadion**.setDisable(**true**);  
 **comand**.setText(*matchFootball*.getComand());  
 **comand**.setDisable(**true**);  
 }  
  
 @FXML  
 **public void** update(ActionEvent event){  
 **dao**.update(*matchFootball*, **match\_name**.getText());  
 Stage stage = (Stage) **buttonOk**.getScene().getWindow();  
 *homeController*.refresh();  
 stage.close();  
 }  
  
 **public static void** setMatchFootball(MatchFootball eq) {  
 *matchFootball* = eq;  
 }  
  
 **public static void** setHomeController(HomeController home){  
 *homeController* = home;  
 }  
}

**Результат выполнения: **

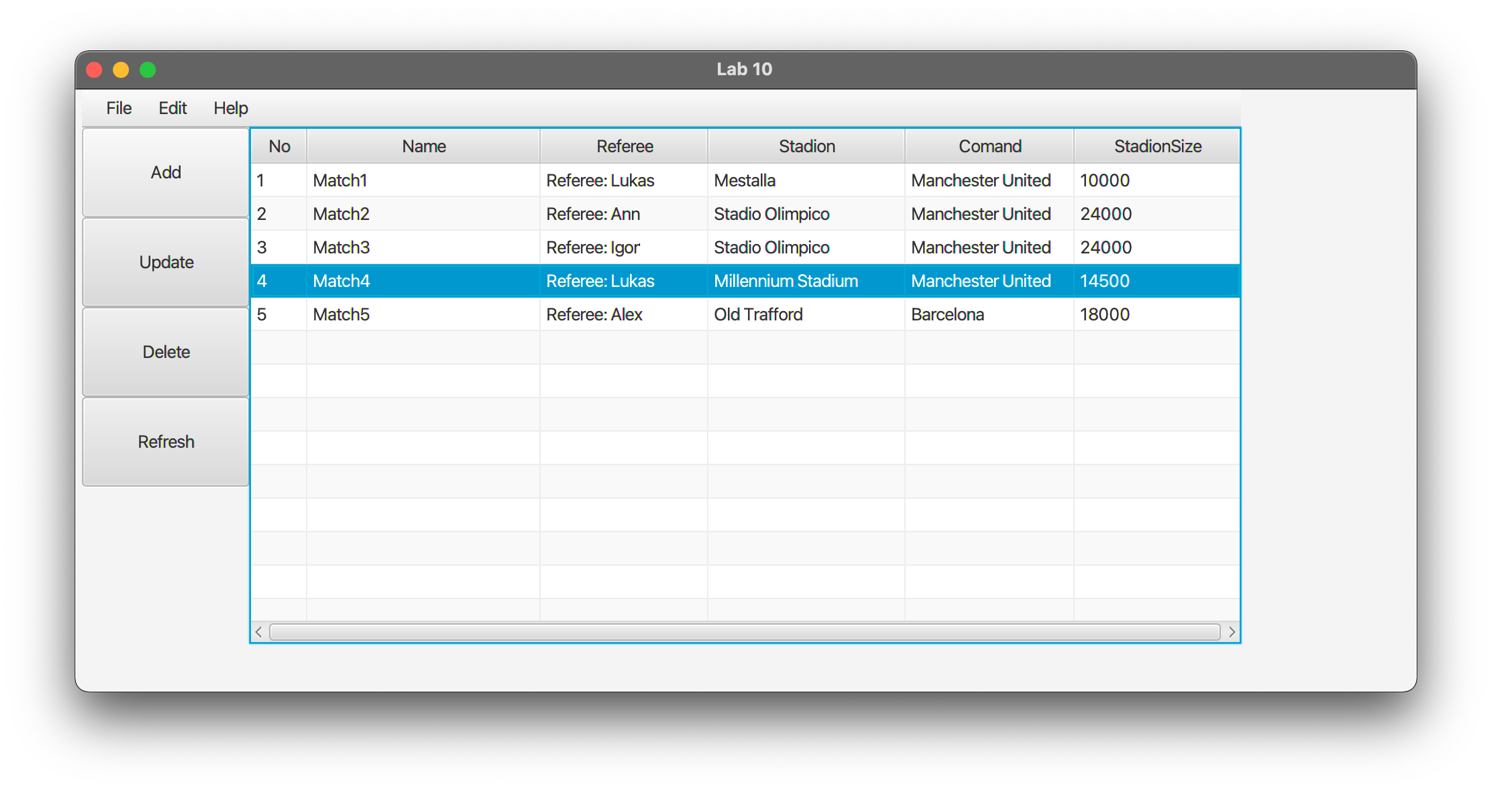
**Add:**

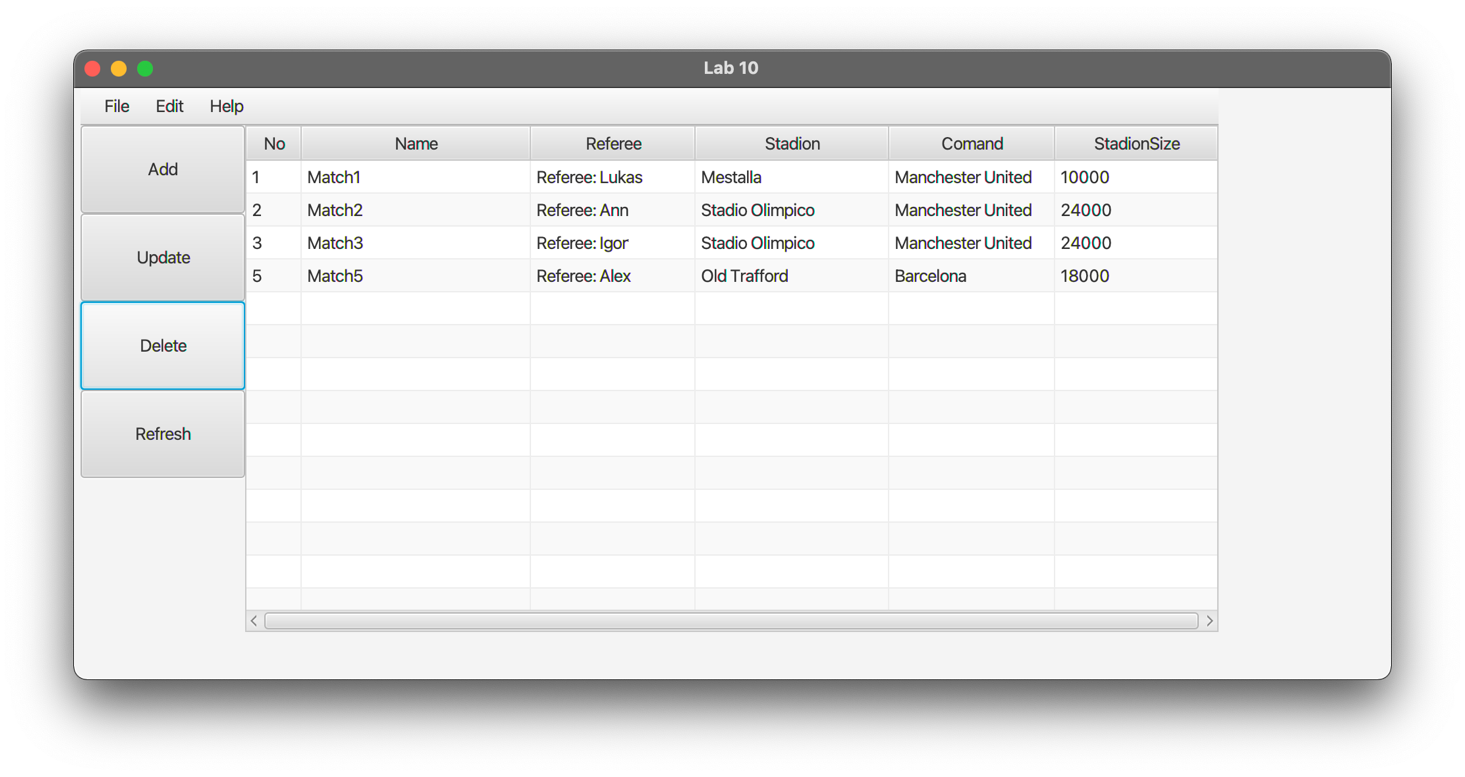
****

**Update:**

****

**Delete:**

****

****

**Вывод:** приобрела практические навыки разработки многооконных приложений на JavaFX для работы с базами данных.