|  |
| --- |
|  |
| МИНОБРНАУКИ РОССИИ |
| *Федеральное государственное бюджетное образовательное учреждение высшего образования* ***«МИРЭА – Российский технологический университет»***  **РТУ МИРЭА** |

**Институт информационных технологий (ИТ)**

**Кафедра инструментального и прикладного программного обеспечения (ИиППО)**

**Дисциплина «Программирование на языке Джава»**

**ОТЧЕТ**

**ПО ПРАКТИЧЕСКОЙ РАБОТЕ №4**

Выполнил студент группы ИНБО-02-20 Бойко К.В.

Принял Степанов П.В.

Практические работы выполнены «\_\_\_»\_\_\_\_\_\_\_2021г.

«\_\_\_\_\_\_\_\_\_\_\_\_\_\_» «\_\_\_»\_\_\_\_\_\_\_2021г.

Отметка о выполнении

**Москва – 2021 г.**

1. **Задание**

Напишите интерактивную программу с использованием GUI имитирует таблицу результатов матчей между командами Милан и Мадрид. Создайте JFrame приложение у которого есть следующие компоненты GUI:

* одна кнопка JButton labeled “AC Milan”
* другая JButton подписана “Real Madrid”
* надпись JLabel содержит текст “Result: 0 X 0”
* надпись JLabel содержит текст “Last Scorer: N/A”
* надпись Label содержит текст “Winner: DRAW”;

Всякий раз, когда пользователь нажимает на кнопку AC Milan, результат будет увеличиваться для Милана, сначала 1 X 0, затем 2 X 0 и так далее. Last Scorer означает последнюю забившую команду. В этом случае: AC Milan. Если пользователь нажимает кнопку для команды Мадрид, то счет приписывается ей. Победителем становится команда, которая имеет больше кликов кнопку на соответствующую, чем другая.

1. **Ход Работы**

(основной код доступен по ссылке <https://github.com/Zyablikitsme/Java_practice>)

В ходе выполнения работы были получены следующие исходные коды:

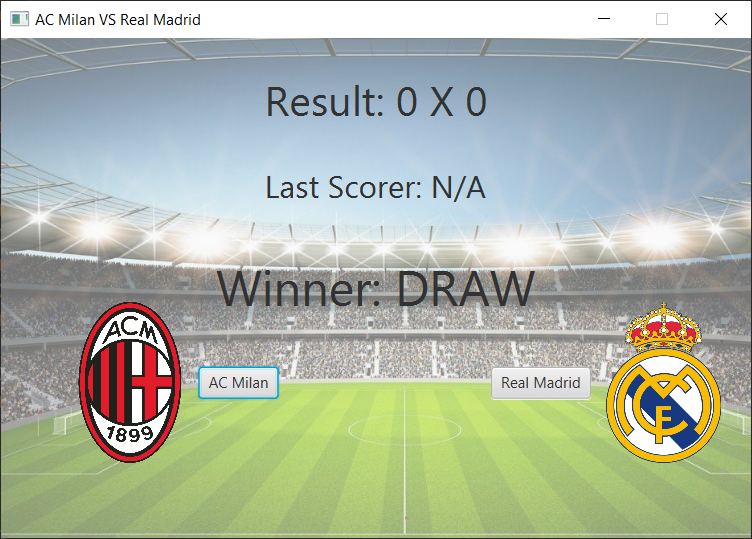
**HelloController.java**

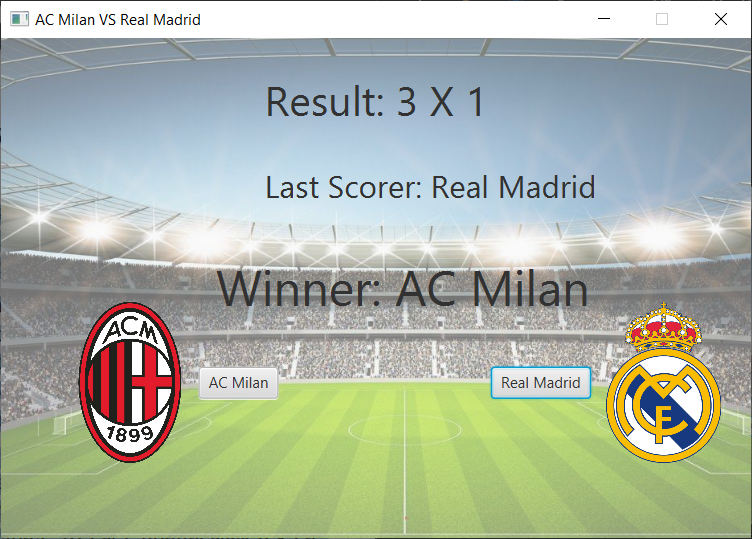
package com.example.test01;  
  
import javafx.fxml.FXML;  
import javafx.scene.control.Button;  
import javafx.scene.control.Label;  
  
public class HelloController {  
  
 private int milanScore = 0, madridScore = 0;  
  
 @FXML  
 private Label label\_result;  
 @FXML  
 private Label label\_lastScorer;  
 @FXML  
 private Label label\_winner;  
  
 @FXML  
 protected void onMilan() {  
 milanScore++;  
 label\_result.setText("Result: " + milanScore + " X " + madridScore);  
 label\_lastScorer.setText("Last Scorer: AC Milan");  
 whoIsWinner();  
 }  
  
 @FXML  
 protected void onMadrid() {  
 madridScore++;  
 label\_result.setText("Result: " + milanScore + " X " + madridScore);  
 label\_lastScorer.setText("Last Scorer: Real Madrid");  
 whoIsWinner();  
 }  
  
 protected void whoIsWinner(){  
 if (madridScore > milanScore) label\_winner.setText("Winner: Real Madrid");  
 else if (madridScore < milanScore) label\_winner.setText("Winner: AC Milan");  
 else label\_winner.setText("Winner: DRAW");  
 }  
  
}

**HelloApplication.java**

package com.example.test01;  
  
import javafx.application.Application;  
import javafx.fxml.FXMLLoader;  
import javafx.scene.Scene;  
import javafx.scene.control.Button;  
import javafx.stage.Stage;  
  
import java.io.IOException;  
  
public class HelloApplication extends Application {  
 @Override  
 public void start(Stage stage) throws IOException {  
 FXMLLoader fxmlLoader = new FXMLLoader(HelloApplication.class.getResource("hello-view.fxml"));  
 Scene scene = new Scene(fxmlLoader.load(), 600, 400);  
 stage.setTitle("AC Milan VS Real Madrid");  
 stage.setScene(scene);  
 stage.setResizable(false);  
 stage.show();  
  
 }  
  
 public static void main(String[] args) {  
 launch();  
 }  
}

**Внешний вид и работа программы:**





1. **Вывод**

В ходе данной работы мной были получены знания по событийному программированию и разработке GUI приложений на Java.