



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 ПРАКТИЧЕСКОМУ ЗАНЯТИЮ №4**

Выполнил студент группы ИНБО-02-20

Лукияненко Д.В.

Принял

Степанов П.В.

Практическая работа выполнена «\_\_» \_\_\_\_\_ 2021 г.

«\_\_\_\_\_»  
Отметка о выполнении

«\_\_» \_\_\_\_\_ 2021 г.

**Москва – 2021 г.**

## СОДЕРЖАНИЕ

Цель работы .....	3
Задание .....	3
Репозиторий .....	3
Выполнение работы .....	3
Код выполненной работы .....	5
Вывод.....	7

## Цель работы

Цель данной практической работы – введение в событийное программирование.

## Задание

Создайте JFrame приложение у которо 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”;

Теперь всякий раз, когда вы нажимаете на кнопку AC Milan, результат будет увеличиваться для Милана, чтобы стать сначала 1 X 0, затем 2 X 0. Last Scorer означает последнюю забившую команду. В этом случае: AC Milan. И победителем становится команда, которая имеет больше кликов кнопку на соответствующую, чем другая

## Репозиторий

Ссылка: [https://github.com/neluckoff/mirea-java-lessons/tree/master/src/ru/luckoff/mirea/practice\\_4](https://github.com/neluckoff/mirea-java-lessons/tree/master/src/ru/luckoff/mirea/practice_4)

## Выполнение работы

Первым делом мною был создан класс App, который наследует JFrame. Внутри этого класса были созданы label и button (метки и кнопки), и две переменные score1 и score2, которые отвечали за голы, забитые командами.

В конструкторе класса App я создал контейнер “pane”, где и была произведена полная настройка окна приложения (добавлены кнопки и метки, выбран цвет фона и кнопок, указан размер окна приложения и расположение кнопок). К кнопкам были привязаны классы действий, имплементирующие ActionListener. В обоих классах были проработаны методы, которые срабатывают после нажатия на кнопку.

Ниже мною будет представлена UML Диаграмма, которая была создана после написании программы (рис. 1) и само окно приложения (рис. 2).

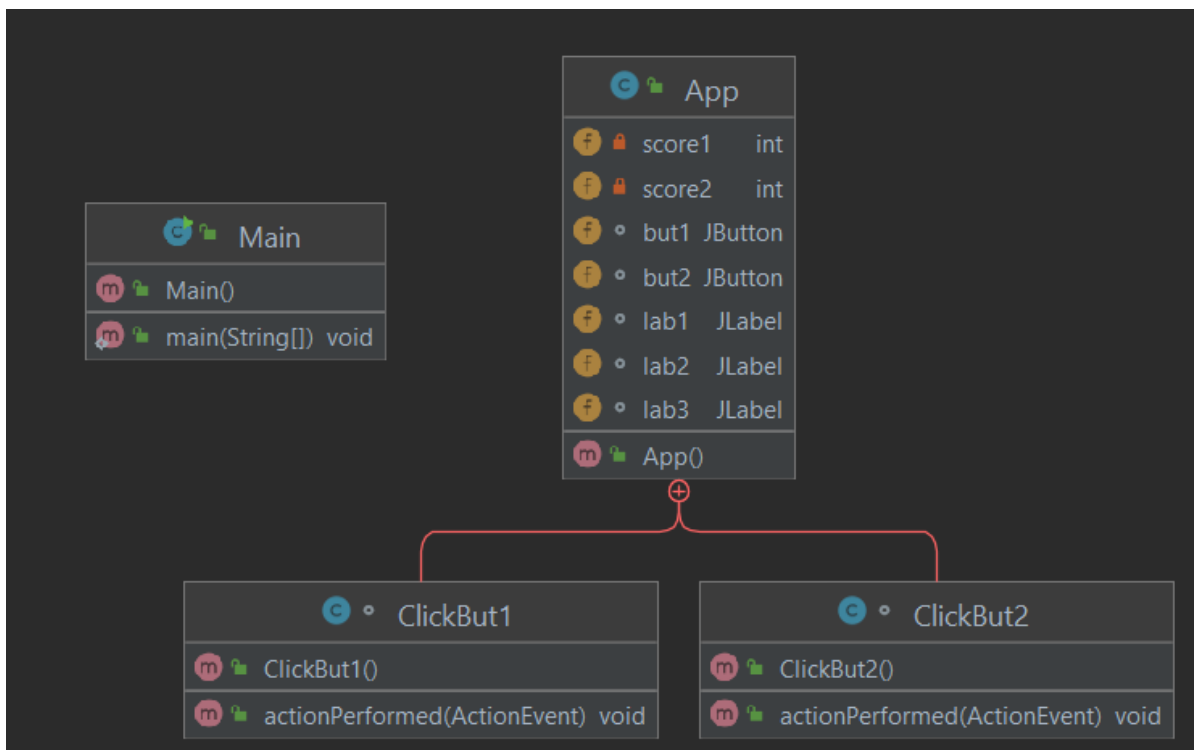


Рисунок 1 – Полученная UML Диаграмма

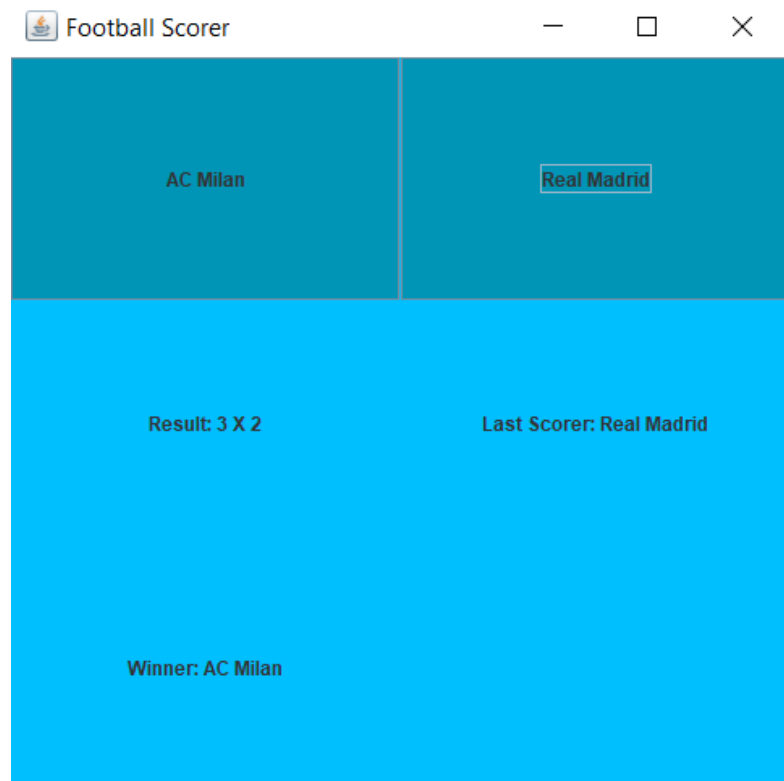


Рисунок 2 – Окно приложения

## Код выполненной работы

Здесь в нескольких скриншотах можно увидеть как выглядит код выполненного задания и результат его работы.

```
public class App extends JFrame {  
    private int score1 = 0;  
    private int score2 = 0;  
  
    JButton but1 = new JButton(text: "AC Milan");  
    JButton but2 = new JButton(text: "Real Madrid");  
  
    JLabel lab1 = new JLabel(text: "Result: " + score1 + " X " + score2, SwingConstants.CENTER);  
    JLabel lab2 = new JLabel(text: "Last Scorer: N/A", SwingConstants.CENTER);  
    JLabel lab3 = new JLabel(text: "Winner: DRAW", SwingConstants.CENTER);  
}
```

Рисунок 3 – Переменные

```
public App() {  
    super(title: "Football Scorer");  
    this.setBounds(x: 200, y: 200, width: 500, height: 500);  
    Container pane = this.getContentPane();  
    pane.setLayout(new GridLayout(rows: 3, cols: 3, hgap: 1, vgap: 1));  
    pane.setBackground(new Color(r: 0, g: 191, b: 255));  
    pane.add(but1);  
    pane.add(but2);  
    but1.addActionListener(new ClickBut1());  
    but2.addActionListener(new ClickBut2());  
    but1.setBackground(new Color(r: 0, g: 149, b: 182));  
    but2.setBackground(new Color(r: 0, g: 149, b: 182));  
  
    pane.add(but1);  
    pane.add(but2);  
    pane.add(lab1);  
    pane.add(lab2);  
    pane.add(lab3);  
}
```

Рисунок 4 – Конструктор App()

```

class ClickBut1 implements ActionListener{
    public void actionPerformed(ActionEvent event){
        score1+=1;
        lab1.setText("Result: " + score1 + " X " + score2);
        lab2.setText("Last Scorer: AC Milan");

        if (score1 > score2) {
            lab3.setText("Winner: AC Milan");
        } else if (score1 < score2) {
            lab3.setText("Winner: Real Madrid");
        } else if (score1 == score2) {
            lab3.setText("Winner: DRAW");
        }
    }
}

```

Рисунок 5 – Класс ClickBut1 (Для кнопки AC Millan)

```

class ClickBut2 implements ActionListener{
    public void actionPerformed(ActionEvent event){
        score2+=1;
        lab1.setText("Result: " + score1 + " X " + score2);
        lab2.setText("Last Scorer: Real Madrid");

        if (score1 > score2) {
            lab3.setText("Winner: AC Milan");
        } else if (score1 < score2) {
            lab3.setText("Winner: Real Madrid");
        } else if (score1 == score2) {
            lab3.setText("Winner: DRAW");
        }
    }
}

```

Рисунок 6 – Класс ClickBut2 (Для кнопки Real Madrid)

## **Вывод**

В результате выполнения данной практической работы я освоил событийное программирование и научился создавать программу с интерфейсом.