



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

Деревянкин Н.А.

Принял

Степанов П.В.

Практическая работа выполнена

«__»_____2021 г.

«_____»

«__»_____2021 г.

Отметка о выполнении

Москва – 2021 г.

СОДЕРЖАНИЕ

Цель работы	3
Задание	3
Выполнение работы	4
Код выполненной работы.....	6
Вывод.....	9

Цель 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

Выполнение работы

Приступив к выполнению, я начал знакомиться с библиотекой Swing.

Создал два файл, один отвечал за запуск приложения, второй – за элементы и действия окна. Сначала я создал класс GUI, который наследует JFrame. Внутри класса я прописал все объекты и переменные, нужные для работы приложения.

Дальше, внутри GUI, я создал конструктор, в котором я создал контейнер и к нему добавил ранее созданные объекты (кнопки, текстовые сообщения). К каждой кнопке я привязал классы действий, которые имплементируют ActionListener и внутри каждого класса создал методы, они будут срабатывать при нажатии на кнопку. И для работы показателя победителя (Winner: DRAW) я написал отдельный метод с условными операторами.

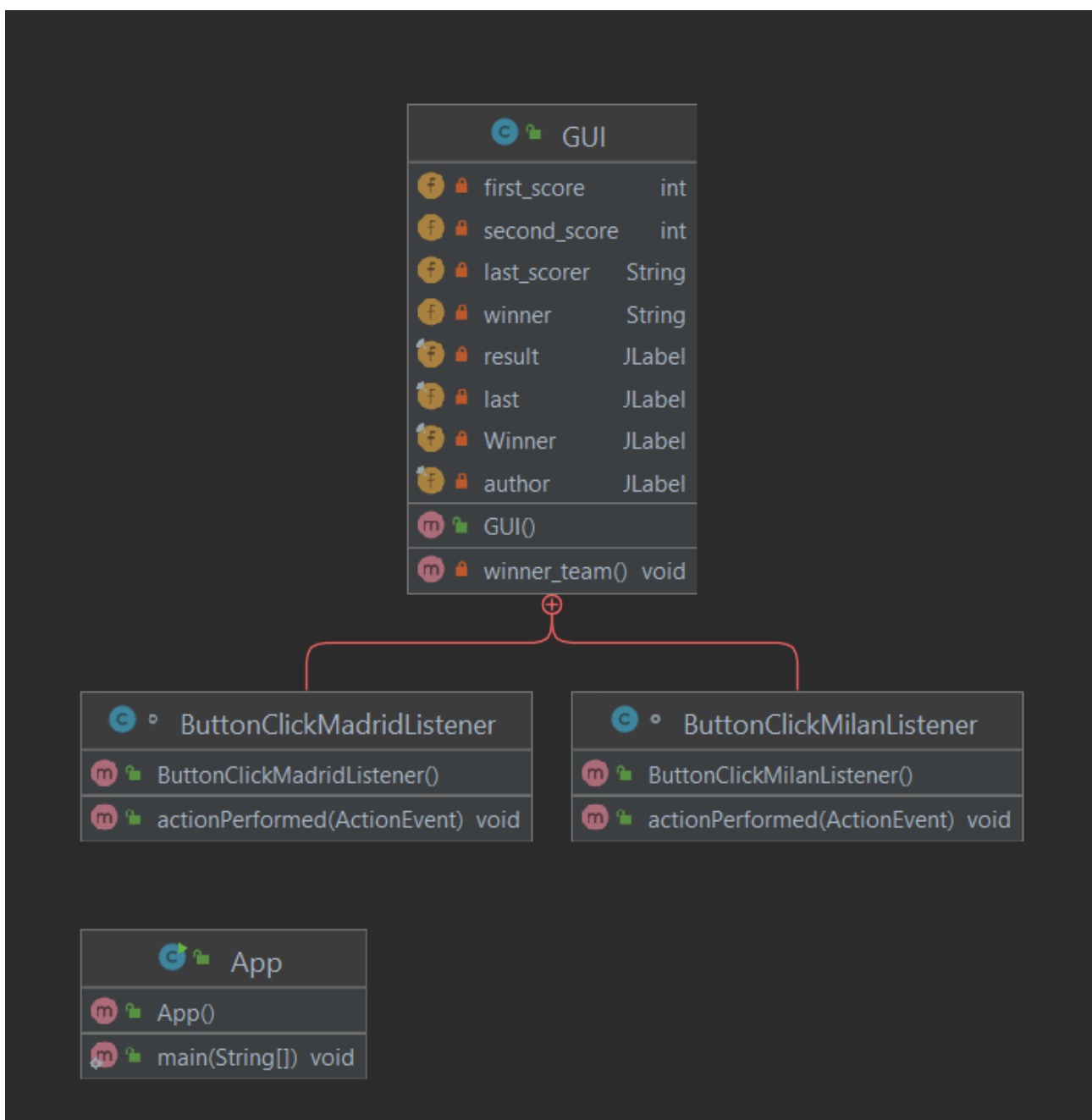


Рисунок 1 – UML Диаграмма

Код выполненной работы

Здесь в нескольких скриншотах можно увидеть, как выглядит код полученного задания и его вывод.

```
public class GUI extends JFrame {
    private int first_score;
    private int second_score;
    private String last_scorer;
    private String winner;
    private final JLabel result = new JLabel( text: "Result: " + first_score + "X" + second_score, SwingConstants.CENTER);
    private final JLabel last = new JLabel( text: "Last Scorer: N/A", SwingConstants.CENTER);
    private final JLabel Winner = new JLabel( text: "Winner: DRAW", SwingConstants.CENTER);
    private final JLabel author = new JLabel( text: "<html>Made by<br> Nikita Derevyankin <br>IN80-02-20", SwingConstants.CENTER);
```

Рисунок 2 – Созданные переменные

```
public GUI () {
    super( title: "Soccer");
    this.setBounds( x: 200, y: 200, width: 300, height: 300);
    this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    Container container = this.getContentPane();
    container.setLayout(new GridLayout( rows: 3, cols: 2, hgap: 2, vgap: 2));
    JButton milan = new JButton( text: "AC Milan");
    container.add(milan);
    milan.addActionListener(new ButtonClickMilanListener ());
    JButton madrid = new JButton( text: "Real Madrid");
    container.add(madrid);
    madrid.addActionListener(new ButtonClickMadridListener ());
    container.add(result);
    container.add(last);
    container.add(Winner);
    container.add(author);
}
```

Рисунок 3 – Конструктор GUI

```
class ButtonClickMilanListener implements ActionListener {  
    public void actionPerformed(ActionEvent e) {  
        first_score++;  
        result.setText("Result: " + first_score + "X" + second_score);  
        last_scorer = "AC Milan";  
        last.setText("Last Scorer: " + last_scorer);  
        winner_team();  
    }  
}
```

Рисунок 4 – Класс ButtonClickMilanListener, для работы кнопки AC Milan

```
class ButtonClickMadridListener implements ActionListener {  
    public void actionPerformed(ActionEvent e) {  
        second_score++;  
        result.setText("Result: " + first_score + "X" + second_score);  
        last_scorer = "Real Madrid";  
        last.setText("Last Scorer: " + last_scorer);  
        winner_team();  
    }  
}
```

Рисунок 5 – Класс ButtonClickMadridListener для работы кнопки Real Madrid

```
private void winner_team() {
    if (first_score > second_score) {
        Winner.setText("Winner: AC Milan");
    }
    else if (first_score == second_score) {
        Winner.setText("Winner: DRAW");
    }
    else {
        Winner.setText("Winner: Real Madrid");
    }
}
}
```

Рисунок 6 – Метод для проверки победителя

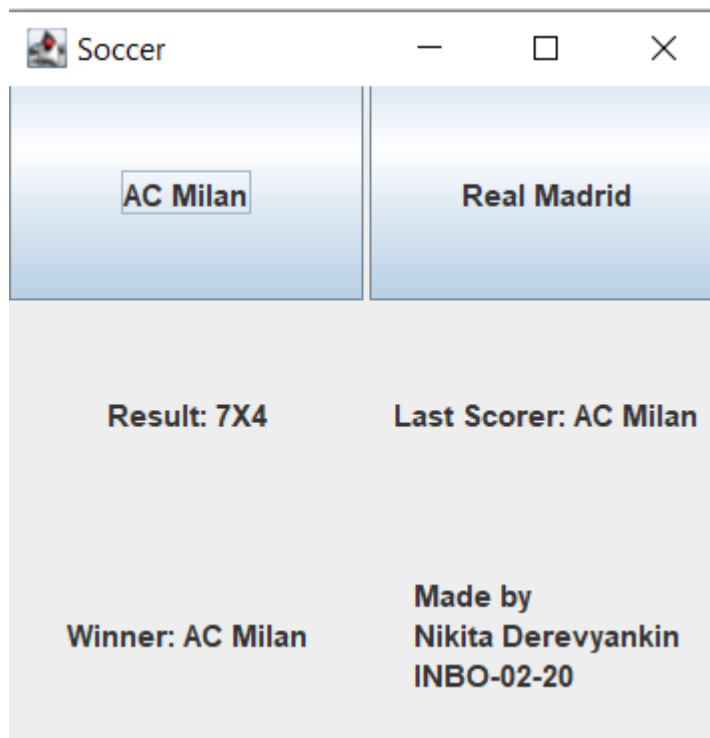


Рисунок 7 – Результат работы

Вывод

В результате выполнения данной практической работы я разобрался с событийным программированием, научился пользоваться библиотекой Swing, и смог создать программу с интерфейсом.