



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
"МИРЭА - Российский технологический университет"

РТУ МИРЭА

Институт информационных технологий (ИТ)
Кафедра инструментального и прикладного программного обеспечения

ОТЧЁТ
ПО ПРАКТИЧЕСКОЙ РАБОТЕ №4
по дисциплине
«Программирование на языке Java»

Тема: Создание GUI. Событийное программирование в Java

Выполнил студент группы ИКБО-16-20

Пак С.А.

Принял ассистент кафедры ИиППО

Русляков А.А.

Практические работы выполнены «_____» 2021г.

«Зачтено» «_____» 2021г

Москва 2021

СОДЕРЖАНИЕ

ТЕОРЕТИЧЕСКОЕ ВВЕДЕНИЕ.....	3
1. Общие сведения.....	3
2. JTextField.....	3
3. JTextArea.....	3
4. Layout менеджеры.....	3
4.1. BorderLayout.....	3
4.2. GridLayout.....	3
4.3. Null Layout.....	3
5. Слушатель событий мыши MouseListener.....	4
6. Меню.....	4
РЕШЕНИЕ ЗАДАЧИ.....	5
1. Постановка задачи.....	5
2. Программный код.....	5
3. Вывод программы.....	7
ВЫВОД.....	9

ТЕОРЕТИЧЕСКОЕ ВВЕДЕНИЕ

1. Общие сведения

Для создания графического интерфейса пользователя можно использовать стандартную Java библиотеку Swing или AWT. В этих библиотеках имеются различные классы, позволяющие создавать окна, кнопки, текстовые поля, меню и другие объекты.

2. JTextField

JTextFields - текстовое поле или поля для ввода текста (можно ввести только одну строку). Примерами текстовых полей являются поля для ввода логина и пароля, например, используемые, при входе в электронную почту.

```
JtextField jta = new JTextField(10)
```

В параметре конструктора задано число 10, это количество символов, которые могут быть видны в текстовом поле.

3. JTextArea

Компонент TextAreas похож на TextFields, но в него можно вводить более одной строки. В конструкторе можно передать количество рядов и столбцов для поля.

4. Layout менеджеры

4.1. BorderLayout

Разделяет компонент на 5 областей (WEST, EAST, NOTH, SOUTH и Center).

Метод для добавления в контейнер, который есть у менеджера BorderLayout отличается и выглядит следующим образом:

```
add( comp , BorderLayout.EAST)
```

Обратите внимание, что мы можем например добавить панели JPanel в эти области и затем добавлять компоненты этих панелей. Мы можем установить расположение этих JPanel используя другие менеджеры.

4.2. GridLayout

С помощью менеджера GridLayout компонент может принимать форму таблицы, где можно задать число строк и столбцов.

4.3. Null Layout

Иногда бывает нужно изменить размер и расположение компонента в контейнере. Таким образом, мы должны указать программе не использовать никакой менеджер компоновки, то есть

```
setLayout (null)
```

5. Слушатель событий мыши **MouseListener**

Мы можем реализовывать слушателей мыши и также слушателей клавиатуры на компонентах GUI. Интерфейс `MouseListener` имеет следующие методы:

- `public void mouseClicked(MouseEvent event)` — выполнен щелчок мышкой на наблюдаемом объекте;
- `public void mouseEntered(MouseEvent event)` — курсор мыши вошел в область наблюдаемого объекта;
- `public void mouseExited(MouseEvent event)` — курсор мыши вышел из области наблюдаемого объекта;
- `public void mousePressed(MouseEvent event)` — кнопка мыши нажата в момент, когда курсор находится над наблюдаемым объектом;
- `public void mouseReleased(MouseEvent event)` — кнопка мыши отпущена в момент, когда курсор находится над наблюдаемым объектом.

Слушатель `MouseListener` можно добавить к компоненту следующим образом:

```
Component.addMouseListener(listener);
```

6. Меню

Добавление меню в программе Java проста. Java определяет три компонента:

- `JMenuBar`: который представляет собой компонент, который содержит меню;
- `JMenu`: который представляет меню элементов для выбора;
- `JMenuItem`: представляет собой элемент, который можно кликнуть из меню.

РЕШЕНИЕ ЗАДАЧИ

1. Постановка задачи

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.

2. Программный код

Файл App.java:

```
package ru.mirea;
```

```
import java.awt.*;
```

```
import java.awt.event.*;
```

```
import javax.swing.*;
```

```
public class App extends JFrame {
```

```
    /*===== КОНСТАНТЫ =====*/
```

```
    private static final int windowWidth = 350;           // ширина  
    окна
```

```
    private static final int windowHeight = 200;          // высота  
    окна
```

```
    private static final String windowTitle = "Real Madrid vs AC  
Milan";           // заголовок окна
```

```
    private static final Font windowFont = new Font("Ubuntu Mono",  
Font.BOLD, 25); // оформление шрифта
```

```
    /*===== ПЕРЕМЕННЫЕ =====*/
```

```
    private short acMilanScore = 0;
```

```
    private short realMadridScore = 0;
```

```

/**
 * Создаёт окно, имитирующее таблицу результатов матчей
 */
private App() {
    super(windowTitle);

    setSize(windowWidth, windowHeight);
    setLayout(new BorderLayout());

    JLabel score = new JLabel("Result: " + acMilanScore + " X " +
realMadridScore, SwingConstants.CENTER);
    JButton acMilan = new JButton("AC Milan");
    JButton realMadrid = new JButton("Real Madrid");
    JLabel lastScorer = new JLabel("Last Scorer: N/A",
SwingConstants.CENTER);
    JLabel winner = new JLabel("Winner: DRAW",
SwingConstants.CENTER);

    acMilan.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            score.setText("Result: " + (++acMilanScore) + " X " +
realMadridScore);
            lastScorer.setText("Last Scorer: AC Milan");
            winner.setText("Winner: " + ((acMilanScore >
realMadridScore)
                                ? "AC Milan"
                                : (acMilanScore == realMadridScore)
                                ? "DRAW"
                                : "Real Madrid"));
        }
    });

    realMadrid.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            score.setText("Result: " + acMilanScore + " X " + (+
+realMadridScore));
            lastScorer.setText("Last Scorer: Real Madrid");
            winner.setText("Winner: " + ((acMilanScore >
realMadridScore)
                                ? "AC Milan"
                                : (acMilanScore == realMadridScore)
                                ? "DRAW"
                                : "Real Madrid"));
        }
    });
}

```

```

score.setFont(windowFont);
acMilan.setFont(windowFont);
realMadrid.setFont(windowFont);
lastScorer.setFont(windowFont);
winner.setFont(windowFont);

    JPanel centerPanel = new JPanel();                // центральная
панель
    centerPanel.setLayout(new FlowLayout());

    centerPanel.add(acMilan);
    centerPanel.add(realMadrid);

    JPanel bottomPanel = new JPanel();                // нижняя
панель
    bottomPanel.setLayout(new GridLayout(2, 1));
    bottomPanel.add(lastScorer);
    bottomPanel.add(winner);

    add(score, BorderLayout.NORTH);
    add(centerPanel, BorderLayout.CENTER);
    add(bottomPanel, BorderLayout.SOUTH);
}

/**
 * Запускает приложение
 * @param args      аргументы командной строки
 */
public static void main(String[] args) {
    App window = new App();

    window.setVisible(true);
    window.setDefaultCloseOperation(EXIT_ON_CLOSE);
}
}

```

3. Вывод программы

Программа создаёт следующее окно (рис.1).

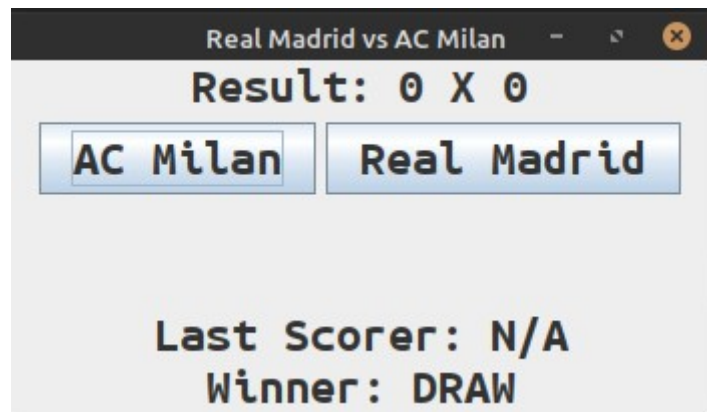


Рис.1 Окно приложения

Что получается, когда Милан ведёт в счёте (рис.2)?



Рис.2 Милан выигрывает

Теперь Реал Мадрид выигрывает (рис.3).

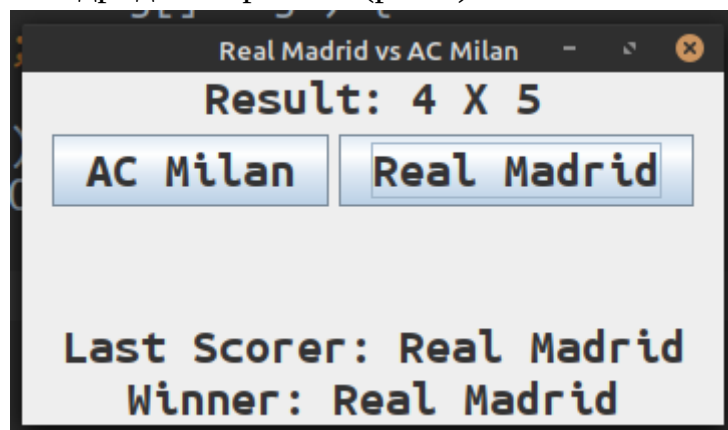


Рис.3 Реал Мадрид выигрывает

ВЫВОД

В ходе выполнения ознакомился с событийным программированием на языке Java.