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

**Институт информационных технологий**

КАФЕДРА ИНСТРУМЕТНАЛЬНОГО И ПРИКЛАДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ (ИППО)

ПРАКТИЧЕСКАЯ РАБОТА № 4

по дисциплине «Объектно-ориентированное программирование»

«СОБЫТИЙНОЕ ПРОГРАММИРОВАНИЕ»

ВАРИАНТ 1

Выполнил студент группы ИНБО-01-18 *Трофимов С. М.*

Принял Батанов А.О.

Практическая работа выполнена «\_\_\_»\_\_\_\_\_\_\_2019г.

«Зачтено» «\_\_\_»\_\_\_\_\_\_\_2019г.

Москва 2019

**Цель работы**

Введение в событийное программирование.

**Теоретическое введение**

Text Fields - 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.

Ответственность за выполнение проверки на наличие ошибок в коде лежит полностью на программисте, например, чтобы проверить произойдет ли ошибка, когда в в качестве входных данных в JTextField ожидается ввод числа. Компилятор не будет ловить такого рода ошибку, поэтому ее необходимо обрабатывать пользовательским кодом.

Компонент TextAreas похож на TextFields, но в него можно вводить более одной строки. В качестве примера TextArea можно рассмотреть текст, который мы набираем в теле сообщения электронной почты

**Выполнение практической работы**

*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.

*Решение:*

Файл GUI.java

**package** com.gradess.company;  
  
**import** javax.swing.\*;  
**import** java.awt.\*;  
**import** java.awt.event.ActionEvent;  
**import** java.awt.event.ActionListener;  
  
**public class** GUI **extends** JFrame **implements** ActionListener {  
  
 **private** JButton **milanButton**;  
 **private** JButton **madridButton**;  
  
 **private static final** String ***MILAN\_NAME*** = **"AC Milan"**;  
 **private static final** String ***REAL\_MADRID\_NAME*** = **"ReaL Madrid"**;  
 **private final static int *MAX\_GOALS*** = 5;  
  
 **private** JLabel **result**;  
 **private** JLabel **lastScorer**;  
 **private** JLabel **winner**;  
  
 **private int milanPoints**;  
 **private int madridPoints**;  
  
 **public** GUI() {  
 **super**(**"Teams"**);  
  
 initWindow();  
  
 initButtons();  
 addButtonToPanel();  
  
 initLabels();  
 addLabelsToLayout();  
 }  
  
 **private void** initWindow() {  
 setSize(400, 200);  
 setLayout(**new** BorderLayout());  
 setVisible(**true**);  
 setResizable(**false**);  
 setLocationRelativeTo(**null**);  
 setDefaultCloseOperation(WindowConstants.***EXIT\_ON\_CLOSE***);  
 }  
  
 **private void** initButtons() {  
 **milanButton** = **new** JButton(***MILAN\_NAME***);  
 **madridButton** = **new** JButton(***REAL\_MADRID\_NAME***);  
  
 **milanButton**.addActionListener(**this**);  
 **madridButton**.addActionListener(**this**);  
 }  
  
 **private void** addButtonToPanel() {  
 JPanel panelButtons = **new** JPanel(**new** GridLayout());  
  
 panelButtons.add(**milanButton**);  
 panelButtons.add(**madridButton**);  
  
 add(panelButtons, BorderLayout.***SOUTH***);  
 }  
  
 **private void** initLabels() {  
 **result** = **new** JLabel();  
 **result**.setHorizontalAlignment(SwingConstants.***CENTER***);  
 updateResult();  
  
 **lastScorer** = **new** JLabel(**"Last Scorer: N/A"**);  
 **lastScorer**.setHorizontalAlignment(SwingConstants.***LEFT***);  
  
 **winner** = **new** JLabel(**"Winner: DRAW"**);  
 }  
  
 **private void** updateResult() {  
 **result**.setText(**"Result: "** + **milanPoints** + **" x "** + **madridPoints**);  
 }  
  
 **private void** addLabelsToLayout() {  
 add(**lastScorer**, BorderLayout.***EAST***);  
 add(**winner**, BorderLayout.***WEST***);  
 add(**result**, BorderLayout.***NORTH***);  
 }  
  
 @Override  
 **public void** actionPerformed(ActionEvent actionEvent) {  
 **if** (actionEvent.getSource() == **milanButton**) {  
 **milanPoints**++;  
 setData(**milanPoints**, ***MILAN\_NAME***);  
 } **else if** (actionEvent.getSource() == **madridButton**) {  
 **madridPoints**++;  
 setData(**madridPoints**, ***REAL\_MADRID\_NAME***);  
 }  
 }  
  
 **private void** setData(**final int** CURRENT\_POINTS, **final** String TEAM\_NAME) {  
 **lastScorer**.setText(**"Last Scorer: "** + TEAM\_NAME);  
 updateResult();  
 **if** (CURRENT\_POINTS >= ***MAX\_GOALS***) {  
 **winner**.setText(**"Winner: "** + TEAM\_NAME);  
 disableButtons();  
 }  
 }  
  
 **private void** disableButtons() {  
 **milanButton**.setEnabled(**false**);  
 **madridButton**.setEnabled(**false**);  
 }  
}

Файл Main.java

**package** com.gradess.company;  
  
**import** javax.swing.\*;  
  
**public class** Main {  
  
 **public static void** main(String[] args) {  
 SwingUtilities.*invokeLater*(GUI::**new**);  
 }  
}

**Результат выполнения программы**

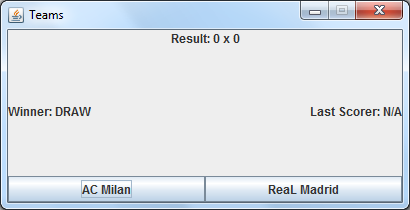


Рис. 1. Окно программы

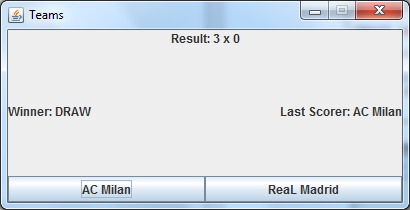


Рис. 2. AC Milan забил 3 гола

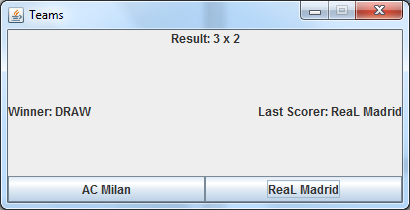


Рис. 3. Real Madrid забил 2 гола

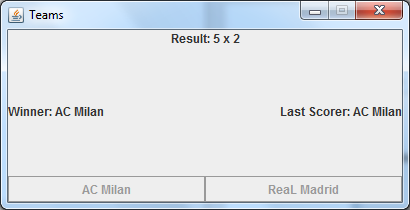


Рис. 4. Команда AC Milan одержала победу

**Выводы по работе**

Я освоил на практике событийное программирование.

**Используемая литература:**

1. К. Сьерра, Б. Бэйтс. Изучаем Java, 2-е издание.
2. С. Макконнелл. Совершенный код, 3-е издание.