|  |  |
| --- | --- |
|  |  |
| 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)

ЛАБОРАТОРНАЯ РАБОТА № 6

по дисциплине «Объектно-ориентированное программирование»

«ОБРАБОТКА СОБЫТИЙ В JAVA ПРОГРАММАХ С ГРАФИЧЕСКИМ ИНТЕРФЕЙСОМ ПОЛЬЗОВАТЕЛЯ»

ВАРИАНТ 2

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

Принял Батанов А.О.

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

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

Москва 2019

**Цель работы**

Цель данной лабораторной работы - научиться обрабатывать различные события для разных компонентов(кнопок, меню и т. д.).

**Теоретическое введение**

Механизм обработки событий библиотеки Swing. В контексте графического интерфейса пользователя наблюдаемыми объектами являются элементы управления: кнопки, флажки, меню и т.д. Они могут сообщить своим наблюдателям об определенных событиях, как элементарных (наведение мышкой, нажатие клавиши на клавиатуре), так и о высокоуровневых (изменение текста в текстовом поле, выбор нового элемента в выпадающем списке и т.д.).

Наблюдателями должны являться объекты классов, поддерживающих специальные интерфейсы (в классе наблюдателя должны быть определенные методы, о которых «знает» наблюдаемый и вызывает их при наступлении события). Такие классы в терминологии Swing называются слушателями.

Интерфейс MouseListener и обработка событий от мыши. События от мыши — один из самых популярных типов событий. Практически любой элемент управления способен сообщить о том, что на него навели мышь, щелкнули по нему и т.д. Об этом будут оповещены все зарегистрированные слушатели событий от мыши.

Слушатель событий от мыши должен реализовать интерфейс MouseListener. В этом интерфейсе перечислены 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 43 объектом
* public void mouseReleased(MouseEvent event) — кнопка мыши отпущена в момент, когда курсор находится над наблюдаемым объектом

Чтобы обработать нажатие на кнопку, требуется описать класс, реализующий интерфейс MouseListener. Далее необходимо создать объект этого класса и зарегистрировать его как слушателя интересующей нас кнопки. Для регистрации слушателя используется метод addMouseListener().

Класс MouseAdapter реализует интерфейс MouseListener, определяя пустые реализации для каждого из его методов. Можно унаследовать своего слушателя от этого класса и переопределить те методы, которые нам нужны.

**Выполнение практической работы**

*Задание:*

Реализация приложения Java, который имеет макет границы и надписи для каждой области в макете. Теперь определим события мыши, чтобы описать действия:

1. Когда мышь входит CENTER программа показывает диалоговое окно (Добро пожаловать в )
2. Когда мышь входит WEST программа показывает диалоговое окно (Добро пожаловать в Джидда) 50
3. Когда мышь входит SOUTH программа показывает диалоговое окно (Добро пожаловать Абха)
4. Когда мышь входит в NORTH программа показывает диалоговое окно (Добро пожаловать в)
5. Когда мышь входит EAST программа показывает диалоговое окно (Добро пожаловать в Дахране)

*Решение:*

Файл LayoutPart.java

**package** com.gradess.company;  
  
**import** javax.swing.\*;  
**import** java.awt.\*;  
**import** java.awt.event.MouseAdapter;  
**import** java.awt.event.MouseEvent;  
  
**public class** LayoutPart **extends** JButton {  
  
 **private final** String **REGION\_NAME**;  
  
 **public** LayoutPart(**final** String REGION\_NAME, **final** String PART\_NAME) {  
 **this**.**REGION\_NAME** = REGION\_NAME;  
  
 JLabel text = **new** JLabel(PART\_NAME, SwingConstants.***CENTER***);  
 setLayout(**new** BorderLayout());  
 add(text);  
  
 addMouseListener(**new** PanelListener());  
 setSize(getSize().**width**, 100);  
 }  
  
 **class** PanelListener **extends** MouseAdapter {  
 @Override  
 **public void** mouseEntered(MouseEvent mouseEvent) {  
 **super**.mouseEntered(mouseEvent);  
  
 JOptionPane.*showMessageDialog*(**null**, **"Добро пожаловать в "** + (**REGION\_NAME** == **null** ? **""** : **REGION\_NAME**));  
 }  
 }  
}

Файл GUI.java

**package** com.gradess.company;  
  
**import** javax.swing.\*;  
**import** java.awt.\*;  
  
**public class** GUI **extends** JFrame {  
  
 **private static final int *SIZE\_X*** = 500;  
 **private static final int *SIZE\_Y*** = 500;  
  
 **public** GUI() **throws** HeadlessException {  
  
 setSize(***SIZE\_X***, ***SIZE\_Y***);  
 setLocationRelativeTo(**null**);  
 setDefaultCloseOperation(WindowConstants.***EXIT\_ON\_CLOSE***);  
  
 add(**new** LayoutPart(**null**, **"CENTER"**), BorderLayout.***CENTER***);  
 add(**new** LayoutPart(**"Джидда"**, **"WEST"**), BorderLayout.***WEST***);  
 add(**new** LayoutPart(**"Абха"**, **"SOUTH"**), BorderLayout.***SOUTH***);  
 add(**new** LayoutPart(**null**, **"NORTH"**), BorderLayout.***NORTH***);  
 add(**new** LayoutPart(**"Дахране"**, **"EAST"**), BorderLayout.***EAST***);  
  
 setVisible(**true**);  
  
 }  
}

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**

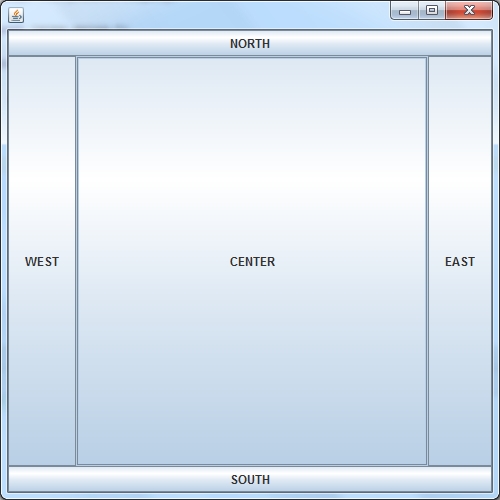


Рис. 1. Главное окно приложения

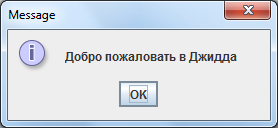


Рис. 2. Диалоговое окно WEST

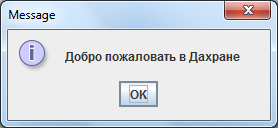


Рис. 3. Диалоговое окно EAST

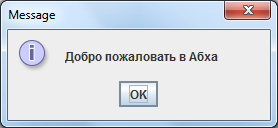


Рис. 4. Диалогое окно SOUTH

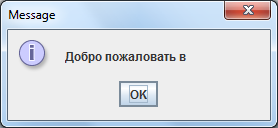


Рис. 5. Диалогое окно NORTH и CENTER

**Выводы по работе**

Я научился обрабатывать различные события для разных компонентов(кнопок, меню и т. д.).

**Используемая литература:**

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