МИНОБРНАУКИ РФ

ФГБОУ ВО «ИжГТУ имени М.Т. Калашникова»

Кафедра «Программное обеспечение»

Отчет

по лабораторной работе №3 по дисциплине

"Архитектура электронных вычислительных машин"

Вариант 5

Выполнил студент группы Б18-191-2: Р. А. Гумметов

Принял: К. С. Чернышев

Ижевск 2020

1. Постановка задачи

1) Выписать все классы и методы, использованные в примере, и их описание из справки, и еще несколько полезных методов для каждого класса.

2) Создать интерфейс по изображению (использовать разные виды компоновок, если одинаковых объектов больше 3, то использовать массив объектов).

2. Классы и методы

Класс JFrame

В библиотеке Swing - окно с рамкой. Является тяжеловесным контейнером для размещения в нем легковесных компонентов, которые операционная система не видит. Окно **JFrame** наследует свойства класса JWindow и представляет собой наиболее часто используемое в приложениях окно «общего назначения».

Метода класса JFrame

|  |  |
| --- | --- |
| **Методы** | **Описание** |
| setLocation() | Эта группа методов позволяет задать позицию и размеры окна на экране. setLocation() задает позицию окна, setSize() позволяет указать его размеры, а с помощью setBounds() можно сразу задать прямоугольник, который займет ваше окно на экране |
| setSize() |
| setBounds() |
| раск() | Позволяет «упаковать» имеющиеся в окне компоненты, так чтобы они занимали столько места, сколько им необходимо. Компоненты при вызове этого метода переходят в «видимое» состояние, хотя и не появляются на экране до вызова одного из следующих методов |
| show() | Отображение окна на экране. После вызова этих методов компоненты переходят в «видимое» состояние и начинают обслуживаться очередью событий. Метод show() к тому же проводит валидацию содержащихся в окне компонентов |
| setVisible() |
| dispose() | Убирает окно с экрана (если оно в момент вызова метода видимо) и освобождает все принадлежащие ему ресурсы. |

Класс JPanel

**JPanel** — это элемент управления, представляющий собой прямоугольное пространство, на котором можно размещать другие элементы. Элементы добавляются и удаляются методами, унаследованными от класса Container.

Методы класса JPanel

setLayout(LayoutManager mgr) - Определяет менеджера размещения

add(Component comp) - Добавить компонент к панели.remove(Component comp или ind index) - удалить компонент.

removeAll() -Удалить все.

Класс JButton

Класс JButton используется для создания элемента управления «кнопка» , который может генерировать ActionEvent при нажатии.

Методы класса JButton

get/setRolloverIcon(Icon icon) - позволяет задать значок, который будет появляться при наведении на кнопкумыши.

get/setPressedIcon(Icon icon) — значок для кнопки в нажатом состоянии.

get/setDisableIcon(Icon icon) — значок для неактивной кнопки.

get/setMargin(Insets margin) - позволяет задать величину отступов от текста надписи на кнопке до ее полей.

setBorderPainted(boolean borderPainted) - Включить/Выключить прорисовку рамки.

setFocusPainted(boolean focusPainted) - Включить/Выключить прорисовку фокуса (кнопка, на которой находится фокус, выделяется пунктирным прямоугольником).

### Класс JTextField

Текстовое поле JTextField является самым простым компонентом и наиболее часто встречающимся в пользовательских интерфейсах. Как правило, поле является однострочным и служит для ввода текста.

Методы класса JTextField

JTextField(int columns) — создает пустое текстовое поле, ширина которого достаточна для размещения columns символов. При этом пользователь может вводить в текстовое поле строку какой угодно длины: она просто будет прокручиваться.

JTextField(String text) — создает текстовое поле с начальным текстом text.

JTextField(String text, int columns) — устанавливает и ширину и начальный текст.

get/setText(String text) -Занести текст в поле и считать.

getSelectedText() - Позволяет получить выделенную часть текста.

replaceSelection(String content) - Заменить выделенный текст другим.

get/setCaretPosition() - Возвращает позицию курсора (каретки) в текстовом поле и позволяет задать ее программно.

setCaretColor(Color color) - Изменить цвет курсора.

setHorizontalAlignment(int align) -Выравнивание текстового поля в качестве параметра передается одна из констант выравнивания, определенных в этом же классе JTextField: LEFT, CENTER, RIGHT.

3. Текст программы

import java.util.\*;

import java.util.Scanner;

import javax.swing.\*;

import java.awt.\*;

public class TestTask

{

public static void main(String[] args)

{

Scanner in = new Scanner(System.in);

JFrame frame = new JFrame("FrameDemo");

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setSize(300,530);

JPanel myPanel1=new JPanel();//создаем панель

JTextArea area1 = new JTextArea("TextArea", 9, 25);

area1.setFont(new Font("Dialog", Font.PLAIN, 14));

area1.setTabSize(10);

myPanel1.add(new JScrollPane(area1));

JPanel myPanel2=new JPanel();

myPanel2.setLayout(new FlowLayout());

myPanel2.add(new JButton("Кнопка 1"));

myPanel2.add(new JButton("Меню"));

myPanel2.add(new JButton("Кнопка 2"));

GridLayout layout = new GridLayout(4, 0, 0, 0);

JPanel myPanel3=new JPanel();

myPanel3.setPreferredSize(new Dimension(300,270));

myPanel3.setLayout(layout);

JButton[] buttons = new JButton[12];

buttons[0] = new JButton("1");

buttons[1] = new JButton("2 ABC");

buttons[2] = new JButton("3 DEF");

buttons[3] = new JButton("4 GHI");

buttons[4] = new JButton("5 JKL");

buttons[5] = new JButton("6 MNO");

buttons[6] = new JButton("7 PQRS");

buttons[7] = new JButton("8 TUV");

buttons[8] = new JButton("9 WXYZ");

buttons[9] = new JButton("\*");

buttons[10] = new JButton("0");

buttons[11] = new JButton("#");

for (int i = 0; i < 12; i++)

{

myPanel3.add(buttons[i]);

buttons[i].setFont(new Font("Button", Font.PLAIN, 16));

}

frame.add(myPanel1,BorderLayout.NORTH);

frame.add(myPanel2,BorderLayout.CENTER);

frame.add(myPanel3,BorderLayout.SOUTH);

frame.setVisible(true);//делаем его видимым

}

}

4. Результаты работы программы

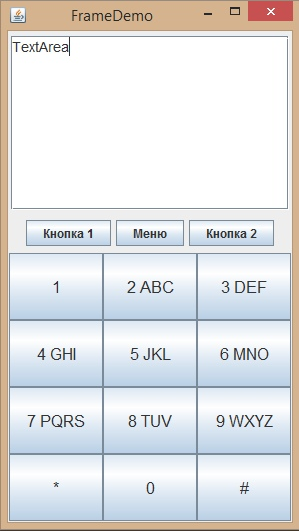


Рис. 1 - Интерфейс по изображению

5. Постановка задачи

1) Найти описание следующих классов-слушателей и выписать основное: ChangeListener, MouseWheelListener.

2) К созданному интерфейсу добавить функциональность – как минимум 3 различных слушателя (например, при нажатии кнопки выводить текст в текстовое поле, при наведении курсора мыши выводить соответствующее сообщение, при изменении ползунка типа JSlider выводить его значение, при нажатии клавиши Enter после набора текста перемещать его куда-нибудь и т.д.).

6. Классы

Класс ChangeListener

Слушатель ChangeListener реагирует на изменение состояния объекта. Каждый элемент управления по своему определяет понятие «изменение состояния». Таким образом, хотя событие это достаточно общее, необходимо уточнять его специфику для каждого конкретного компонента. В интерфейсе определен всего один метод:

public void stateChanged (ChangeEvent event).

Класс MouseWheelListener

Слушатель MouseWheelListener оповещается при вращении колесика мыши в тот момент, когда данный компонент находится в фокусе. Этот интерфейс содержит всего один метод:

public void mouseWheelMoved (MouseWheelEvent event).

7. Текст программы

import org.omg.CosNaming.IstringHelper;

import java.util.\*;

import java.awt.BorderLayout;

import java.awt.event.ActionEvent; //класс для обработки событий

import java.awt.event.ActionListener; //класс- интерфейс слушателя

import java.util.Scanner;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

class ActionListenerClear implements ActionListener {

public JTextArea area;

ActionListenerClear(JTextArea area) {

this.area = area;

}

public void actionPerformed(ActionEvent e) {

//Код, который нужно выполнить при нажатии

area.setText(null);

area.setCaretPosition(0);

}

}

class ActionListenerRegister implements ActionListener {

public JTextArea area;

ActionListenerRegister(JTextArea area) {

this.area = area;

}

public void actionPerformed(ActionEvent e) {

String text = area.getText().toUpperCase();

area.setText(null);

area.setText(text);

}

}

class MyMouseAdapter extends MouseAdapter{

JButton button3;

MyMouseAdapter(JButton button) {

this.button3 = button;

}

public void mouseEntered(MouseEvent e) {

button3.setBackground(Color.lightGray);

button3.setForeground(Color.GREEN);

}

public void mouseExited(MouseEvent e) {

button3.setBackground(null);

button3.setForeground(Color.BLACK);

}

}

class MyKeyListener extends KeyAdapter {

public JTextArea area;

MyKeyListener(JTextArea area) {

this.area = area;

}

public void keyPressed(KeyEvent e) {

if (e.isAltDown()) {

area.setForeground(Color.RED);

}

else if (e.isControlDown()) {

area.setForeground(Color.BLUE);

}

else if (e.isShiftDown()) {

area.setForeground(null);

}

}

}

public class TestTask

{

public static void main(String[] args)

{

Scanner in = new Scanner(System.in);

JFrame frame = new JFrame("FrameDemo");

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setSize(300,530);//задаем размер окна

JPanel myPanel1=new JPanel();//создаем панель

JTextArea area1 = new JTextArea("TextArea\n", 9, 26);

area1.setCaretPosition(9);

area1.setFont(new Font("Dialog", Font.PLAIN, 14));

area1.setTabSize(10);

myPanel1.add(new JScrollPane(area1));

JPanel myPanel2=new JPanel();

myPanel2.setLayout(new FlowLayout());

JButton button1 = new JButton("Регистр");

JButton button2 = new JButton("Меню");

JButton button3 = new JButton("Очистить");

ActionListener actionListener1 = new ActionListenerClear(area1);

button3.addActionListener(actionListener1);

button3.setFocusable(false);

ActionListener actionListener2 = new ActionListenerRegister(area1);

button1.addActionListener(actionListener2);

button1.setFocusable(false);

button2.setFocusable(false);

KeyListener actionListener3 = new MyKeyListener(area1);

area1.addKeyListener(actionListener3);

button1.addMouseListener(new MyMouseAdapter(button1));

button2.addMouseListener(new MyMouseAdapter(button2));

button3.addMouseListener(new MyMouseAdapter(button3));

myPanel2.add(button1);

myPanel2.add(button2);

myPanel2.add(button3);

GridLayout layout = new GridLayout(4, 0, 0, 0);

JPanel myPanel3=new JPanel();

myPanel3.setPreferredSize(new Dimension(300,270));

myPanel3.setLayout(layout);

JButton[] buttons = new JButton[12];

buttons[0] = new JButton("1");

buttons[1] = new JButton("2 ABC");

buttons[2] = new JButton("3 DEF");

buttons[3] = new JButton("4 GHI");

buttons[4] = new JButton("5 JKL");

buttons[5] = new JButton("6 MNO");

buttons[6] = new JButton("7 PQRS");

buttons[7] = new JButton("8 TUV");

buttons[8] = new JButton("9 WXYZ");

buttons[9] = new JButton("\*");

buttons[10] = new JButton("0");

buttons[11] = new JButton("#");

for (int i = 0; i < 12; i++)

{

myPanel3.add(buttons[i]);

buttons[i].setFont(new Font("Button", Font.PLAIN, 16));

buttons[i].addMouseListener(new MyMouseAdapter(buttons[i]));

buttons[i].setFocusable(false);

}

frame.add(myPanel1,BorderLayout.NORTH);

frame.add(myPanel2,BorderLayout.CENTER);

frame.add(myPanel3,BorderLayout.SOUTH);

frame.setVisible(true);

frame.pack();

frame.setMinimumSize(frame.getSize());

}

}

8. Результаты работы программы

На рис. 2 - 4 представлены результаты программы:

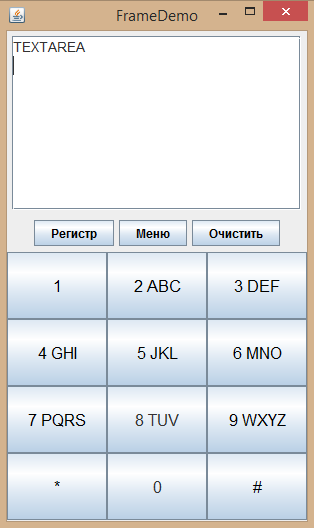


Рис. 2 - Преобразование текста в верхний регистр

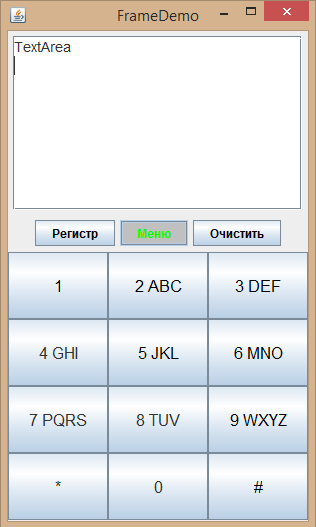


Рис. 3 - Подсвечивание кнопки при наведении курсором

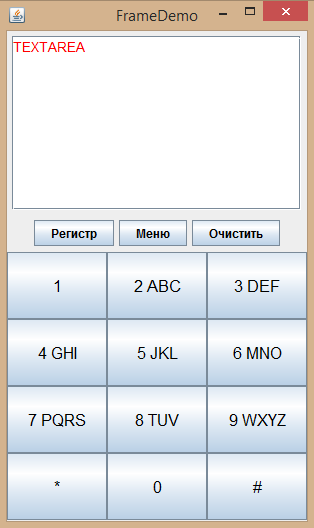


Рис. 4 - Изменение цвета текста при нажатии клавиши "Alt"