

Практическая работа №12. Создание программ с графическим интерфейсом пользователя на языке Джава.

Компоновка объектов с помощью Layout менеджеров

Цель: цель данной практической работы - научиться создавать графический интерфейс пользователя, освоить на практике работу с различными объектами для создания GUI, менеджерами размещения компонентов.

Теоретические сведения

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

Компонент TextField

Компонент 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.

Пример создания объекта класса JTextField:

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

В параметре конструктора задано число 10, это количество символов, которые могут быть видны в текстовом поле. Текст введенный в поле JTextField может быть возвращен с помощью метода `getText()`. Также в поле можно записать новое значение с помощью метода `setText(String s)`.

Как и у других компонентов, мы можем изменять цвет и шрифт текста в текстовом поле.

Листинг 11.1 – Пример пользовательского класса окна, наследуемая от JFrame

```
class LabExample extends JFrame{
    JTextField jta = new JTextField(10);
    //можно задать свойства шрифта
    Font      fnt      =      new      Font("Times      new
roman",Font.BOLD,20); LabExample() {
    // вызываем конструктор родителя JFrame
    super("Example");
```

```

//устанавливаем менеджер компоновки FlowLayout
setLayout(new FlowLayout());
//устанавливаем размеры окна
setSize(250,100);
//добавляем текстовое поле к окну
add(jta);
//задаем цвет фона
jta.setForeground(Color.PINK);
//задаем цвет шрифта
jta.setFont(fnt);
setVisible(true);
}
public static void main(String[]args){
//вызываем конструктор класса LabExample
new LabExample();
}
}

```

На рис.11.1 представлен пример работы программы на листинге 11.



Рисунок 11.1 Пример работы программы на листинге 11.1

Важное замечание.

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

Выполните следующий пример и наблюдайте за результатом, когда число вводится в неправильном формате:

Менеджеры компоновки: BorderLayout

Разделяет компонент на пять областей (WEST, EAST, NORTH, SOUTH and Center). Другие компоненты могут быть добавлены в любой из этих компонентов пятерками.

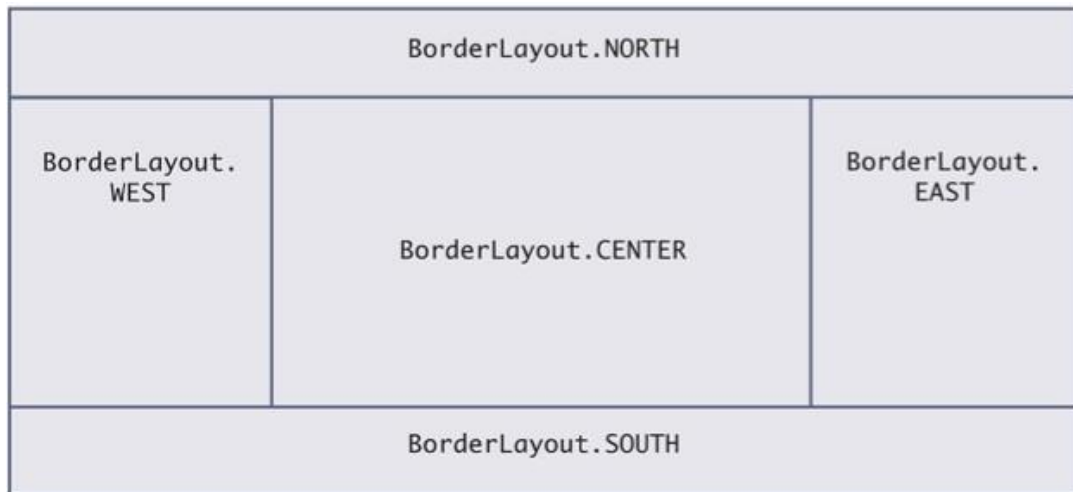


Рисунок 11.2 – Схема менеджера BorderLayout

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

GridLayout менеджер

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

1	2	3	4
5	6	7	8
9	1	1	1
	0	1	2

если компоненту GridLayout задать 3 строки и 4 столбца, то компоненты будут принимать форму таблицы, показанной выше, и будут всегда добавляться в порядке их появления.

Следующий пример иллюстрирует смесь компоновки различных компонентов

Листинг 11.4 – Пример программы

```
import javax.swing.*.*;
import java.awt.*.*;
```

```

import java.awt.event.*;
class BorderExample extends JFrame{
    JPanel[] pnl = new JPanel[12];
    public BorderExample(){
        setLayout(new GridLayout(3,4));
        for(int i = 0 ; i < pnl.length ; i++){
            int r = (int) (Math.random() *255);
            int b = (int) (Math.random() * 255);
            int g =(int) (Math.random() * 255);
            pnl[i] = new JPanel();
            pnl[i].setBackground(new Color(r,g,b));
            add(pnl[i]);
        }
        pnl[4].setLayout(new BorderLayout());
        pnl[4].add(new JButton("one"),BorderLayout.WEST);
        pnl[4].add(new JButton("two"),BorderLayout.EAST);
        pnl[4].add(new
JButton("three"),BorderLayout.SOUTH);
        pnl[4].add(new
JButton("four"),BorderLayout.NORTH);
        pnl[4].add(new
JButton("five"),BorderLayout.CENTER);
        pnl[10].setLayout(new FlowLayout());
        pnl[10].add(new JButton("one"));
        pnl[10].add(new JButton("two"));
        pnl[10].add(new JButton("three"));
        pnl[10].add(new JButton("four"));
        pnl[10].add(new JButton("five"));
        setSize(800,500);
    }
    public static void main(String[] args){
        new BorderExample().setVisible(true);
    }
}

```

Вот такой будет иметь вид, представленный выше код

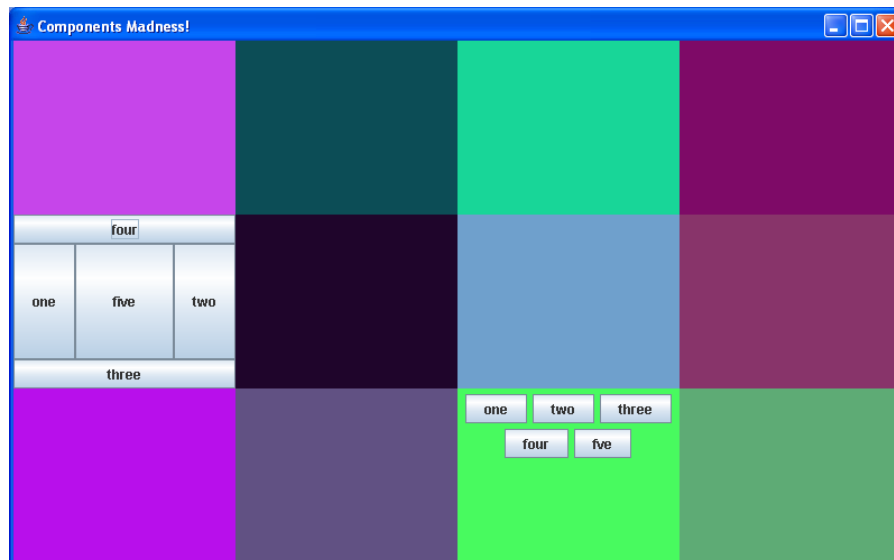


Рисунок 11.3 – Пример работы программы на листинге 11.4

Заметьте, что компонент `JFrame` использует менеджер компоновки `GridLayout` в виде таблицы размера 3 на 4, в то время как компонент `JPanel` размером 2 на 1 использует менеджер компоновки `BorderLayout`. А компонент `JPanel` размер 3 на 3 использует менеджер компоновки `FlowLayout`.

Менеджер Layout по умолчанию

Иногда бывает нужно изменить размер и расположение компонента в контейнере. Таким образом, мы должны указать программе не использовать никакой менеджер компоновки, то есть использовать `setLayout (null)`. Так что мы получим что-то вроде этого:

Листинг 11.5 – Пример программы

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
class NullLayout extends JFrame{
    JButton but1 = new JButton("One");
    JButton but2 =new JButton("two");
    JButton but3 = new JButton("three");
    public NullLayout(){
        setLayout(null);
        but1.setBounds(150,300,100,20);
        // добавляем 150,300 ширина = 100, высота =20
        but2.setSize(80,400);
```

```

// добавляем к 0,0 ширина = 80, высота =400
but3.setLocation(300,1 00);
but3.setSize(200,75);
// those two steps can be combined in oneseBounds
method call
add(but1);
add(but2);
add(but3);
setSize(500,500);
}
public static void main(String[]args){
new NullLayout().setVisible(true);
}
}

```

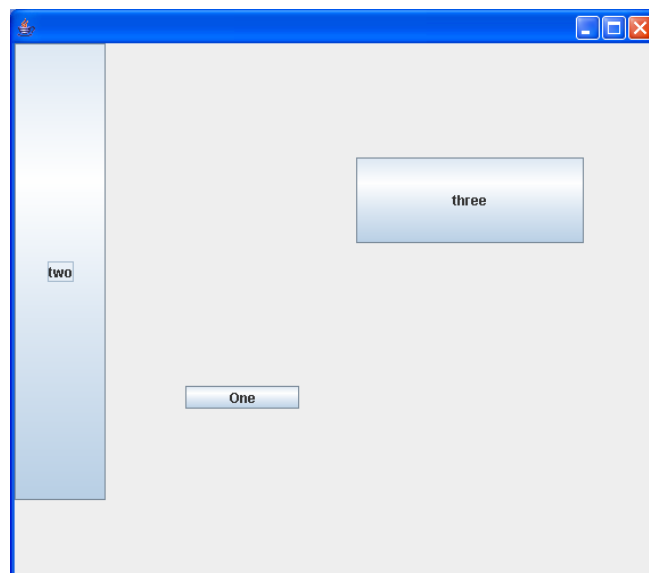


Рисунок 11.4 – Пример работы программы на листинге 11.5

Организация меню

Добавление меню в программе Джава проста. Джава определяет три компонента для обработки этих

JMenuBar: который представляет собой компонент, который содержит меню.

JMenu: который представляет меню элементов для выбора.



Рисунок 11.5 – Диаграмма класса MovableRectangle.

JMenuItem: представляет собой элемент, который можно кликнуть из меню.

Подобно компоненту Button (на самом деле MenuItems являются подклассами класса AbstractButton). Мы можем добавить ActionListener к ним так же, как мы делали с кнопками

Задания на практическую работу №11

1. Создать окно, нарисовать в нем 20 случайных фигур, случайного цвета. Классы фигур должны наследоваться от абстрактного класса Shape, в котором описаны свойства фигуры: цвет, позиция.
2. Создать окно, отобразить в нем картинку, путь к которой указан в аргументах командной строки.
3. Создать окно, реализовать анимацию, с помощью картинки, состоящей из нескольких кадров.

