

Практическая работа №5 Создание программ с графическим интерфейсом пользователя на языке Джава

Цель: цель данной практической работы – научиться разрабатывать программы на языке Джава с использованием графического интерфейса пользователя.

Теоретические сведения

Swing в Джава — это набор инструментов графического интерфейса пользователя (GUI), который включает компоненты GUI. Swing предоставляет богатый набор виджетов и пакетов для создания сложных компонентов графического интерфейса пользователя для приложений Java. Swing является частью Java Foundation Classes (JFC), который представляет собой API для программирования Java GUI, который предоставляет GUI. Библиотека Java Swing построена на основе Java Abstract Widget Toolkit (AWT), более старого, зависящего от платформы инструментария графического интерфейса пользователя. Вы можете использовать простые программные компоненты Java с графическим интерфейсом пользователя, такие как кнопка, текстовое поле и т. д. Из библиотеки, и вам не нужно создавать компоненты с нуля. Схема иерархии классов графической библиотеки Swing представлена на рис. 5.1.

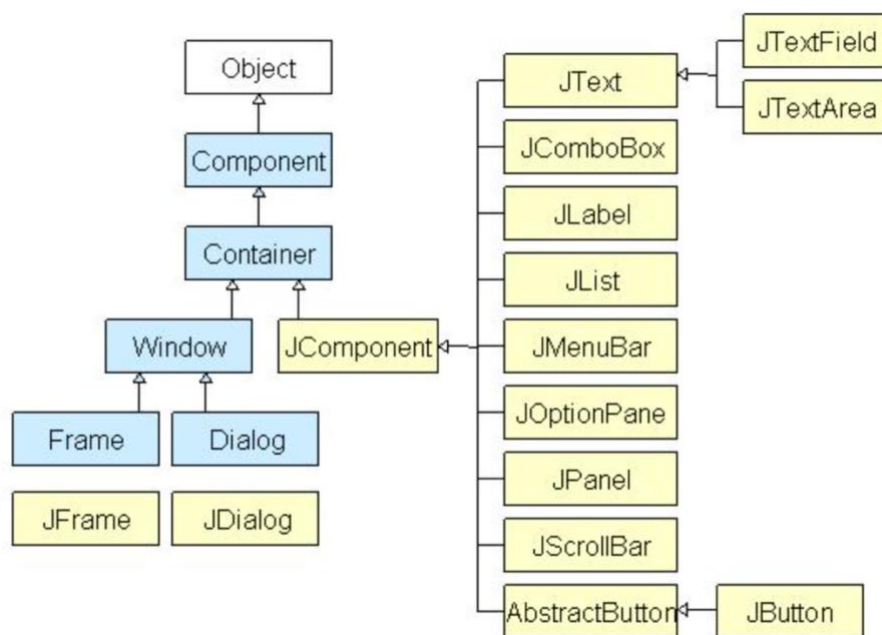


Рисунок 5.1. Схема иерархии классов Swing

Все компоненты в Java Swing — это JComponent, который можно добавлять в классы-контейнеры.

Классы-контейнеры

Классы-контейнеры — это классы, на которых могут быть другие компоненты. Итак, для создания графического интерфейса Java Swing нам понадобится хотя бы один объект-контейнер. Существует три типа контейнеров Java Swing:

- **JPanel (Панель):** это чистый контейнер, а не окно. Единственная цель Panel - организовать компоненты в окне.
- **JFrame (Фрейм):** это полностью функционирующее окно со своим заголовком и значками.
- **JDialog (Диалог):** это можно представить как всплывающее окно, которое выскакивает, когда необходимо отобразить сообщение. Это не полностью функционирующее окно, как Frame.

Рассмотрим последовательно все шаги по созданию графического интерфейса на Джава с помощью Swings

Листинг 5.1 – Пример 1 программы с GUI

```
import javax.swing.*;
class FirstGui{
public static void main(String args[]){
//создаем фрейм окна с помощью конструктора
//Конструктор берет параметр - название окна - это строка
JFrame frame = new JFrame("My First GUI");
// устанавливаем реакцию окна на закрытие по умолчанию
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
//задаем свойства окна - его размеры в пикселях
frame.setSize(300,300);
//создаем кнопку с помощью конструктора класса JButton
//конструктор берет параметр строку - название на кнопке
JButton button = new JButton("Press");
//добавляем кнопку ук окну
frame.getContentPane().add(button);
//делаем окно видимым
```

```
frame.setVisible(true);  
    }  
}
```

На рис. 5.2 показан результат работы программы на листинге 5.1

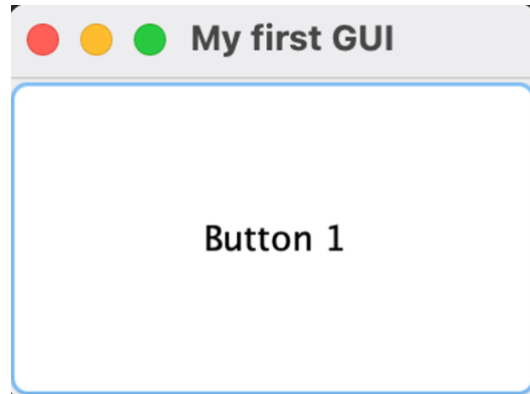


Рисунок 5.2. Окно графического интерфейса программы 5.2

Листинг 5.2 – Пример 2 программы с GUI

```
import javax.swing.*;  
class SecondGui{  
    public static void main(String args[]){  
        JFrame frame = new JFrame("My First GUI");  
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        frame.setSize(300,300);  
        JButton button1 = new JButton("Button 1");  
        //создали еще одну кнопку  
        JButton button2 = new JButton("Button 2");  
        frame.getContentPane().add(button1);  
        //добавили вторую кнопку  
        frame.getContentPane().add(button2);  
        frame.setVisible(true);  
    }  
}
```

На рис. 5.3 показан результат работы программы на листинге 5.2

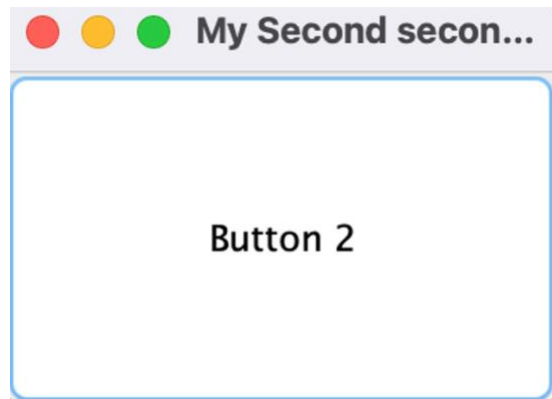


Рисунок 5.3. Окно графического интерфейса программы 5.3

Мы видим на рисунке только вторую кнопку. Дело в том, что приложение имеет многослойную структуру и следующий компонент, который мы добавили закрывает или наслаивается на предыдущий. Для того чтобы избежать этого эффекта нужно использовать контейнеры для организации компонентов и менеджеры компоновки. Добавим к нашему коду объект класса `JPanel`, он использует менеджер компоновки по умолчанию. Но вы можете изменить его с помощью метода `setLayout()`.

Листинг 5.3 – Пример 3 программы с GUI

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

public class ThirdGui {
    public static void main(String args[]) {
        JFrame frame = new JFrame("My Second second GUI");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(200, 150);
        //создали панель
        JPanel panel= new JPanel();
        //задали свойство панели – цвет фона
        panel.setBackground(Color.GRAY);
        // задали свойство панели размеры
        panel.setPreferredSize( new Dimension(200,300));
        JButton button1 = new JButton("Button 1");
        JButton button2 = new JButton("Button 2");
        panel.add(button1);
        panel.add(button2);
        //добавили панель к фрейму
        frame.getContentPane().add(panel);
    }
}
```

```
//упакуем во фрейм
frame.pack();
frame.setVisible(true);

}
}
```

На рис. 5.4 показан результат работы программы на листинге 5.3. Два объекта класса `JButton` будут добавляться последовательно один за другим, том порядке в котором они были добавлены с помощью метода `add` в программе, поскольку в данном случае используется менеджер компоновки по умолчанию. Если вы хотите изменить расположение объектов используйте менеджеры компоновки и их сочетания. Изменить можно с помощью метода `setLayout`.



Рисунок 5.4. Окно графического интерфейса программы 5.3

Класс JLabel

`JLabel` это область для отображения короткой строки, изображения или того и другого. Обычно объекты `JLabel` добавляются на панели. При добавлении следующего кода Java после создания `JFrame` создается надпись с текстом “I’m a JLabel”. Пример кода:

```
JLabel label = new JLabel("I'm a JLabel",
JLabel.CENTER);
frame.add(label);
```

Для позиции метки, он может быть указан `JLabel.LEFT`, `JLabel.CENTER`, `JLabel.RIGHT`, из которых позиция может быть установлена слева, по центру и справа correspondently. На рисунке ниже показано, что метка установлена в центре окна.

Мы получим результат как на рис. 5.5



Рисунок 5.5. Окно Программы с JLabel

Задания на практическую работу №5

1. Напишите интерактивную программу с использованием GUI имитирует таблицу результатов матчей между командами Милан и Мадрид. Порядок работы: 1) Создайте пользовательское JFrame приложение, у которого есть следующие компоненты GUI:

- одна кнопка JButton подписана “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. 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.

3. Создать окно, отобразить в нем картинку, путь к которой указан в аргументах командной строки.

4. Создать окно, реализовать анимацию, с помощью картинки,

состоящей из нескольких кадров.

