

# SWING

Создание GUI приложений на Java

# JFC

- Java Foundation Classes
  - Swing
  - Pluggable look-and-feel
  - Accessibility API
  - Java 2D API
  - Internationalization

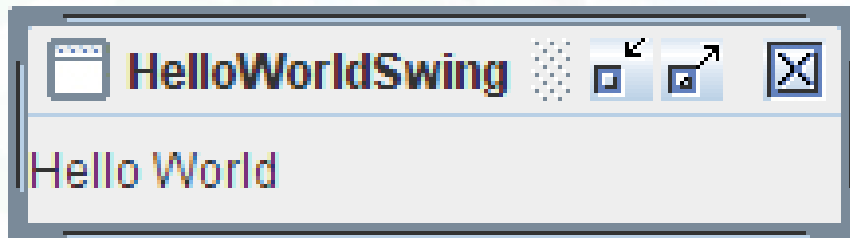
# Пакеты swing

- javax.accessibility
- javax.swing.plaf
- javax.swing.text
- javax.swing
- javax.swing.plaf.basic
- javax.swing.text.html
- javax.swing.border
- javax.swing.plaf.metal
- javax.swing.text.html.parser
- javax.swing.colorchooser
- javax.swing.plaf.multi
- javax.swing.text.rtf
- javax.swing.event
- javax.swing.plaf.synth
- javax.swing.tree
- javax.swing.filechooser
- javax.swing.table
- javax.swing.undo



# HelloWorld!

- import  
javax.swing.\*;
- JFrame - окно
- JLabel — текст  
(метка)



# HelloWorld!

```
import javax.swing.*;

// Это очень простое приложение, демонстрирующее
// базовые возможности Swing.

public class HelloWorldSwing {

    public static void main(String[] args) {

        // тело

    }

}
```

# HelloWorld!

```
// Создаем главное окно
```

```
JFrame frame = new JFrame("Привет, мир от Swing");
```

```
// Создаем метку в этом окне
```

```
final JLabel label = new JLabel("Привет, мир!");
```

```
frame.getContentPane().add(label);
```

```
// Эти операции исполняются практически в каждом Swing-приложении
```

```
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

```
frame.pack();
```

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



# HelloWorld!

```
public class HelloWorldSwing {  
    // Создаем главное окно и показываем  
    // Этот метод должен выполняться в потоке обработчика событий GUI  
    private static void createAndShowGUI() {  
    }  
    public static void main(String[] args) {  
        // Задание для запуска в потоке обработчика событий GUI  
        javax.swing.SwingUtilities.invokeLater(new Runnable() {  
            public void run() {  
                createAndShowGUI();  
            }  
        }); } } }
```

# GUI приложение

- Поток обработки событий (event-dispatching thread)

```
while (true) {  
    if (!eventQueue.empty()) {  
        processEvents();  
    }  
    Treads.sleep(1);  
}
```

- Событие



ButtonClick!

# ButtonClick!

```
import javax.swing.*;
import javax.swing.border.Border;

import java.awt.*;
import java.awt.event.*;

public class SwingApplication {
    private static final String labelPrefix = "Число нажатий на кнопку: ";
    private static int numClicks = 0;

    // Создаем внутреннее содержание окна
    public static Component createComponents() {
    }

    // В главной функции создается окно приложения.
    public static void main(String[] args) {
    }
}
```

# ButtonClick!

```
public static void main(String[] args) {  
  
    // Создаем окно приложения и помещаем в него компоненты.  
    JFrame frame = new JFrame("SwingApplication");  
    Component contents = createComponents();  
    frame.getContentPane().add(contents, BorderLayout.CENTER);  
  
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
    frame.pack();  
    frame.setVisible(true);  
}
```



# ButtonClick!

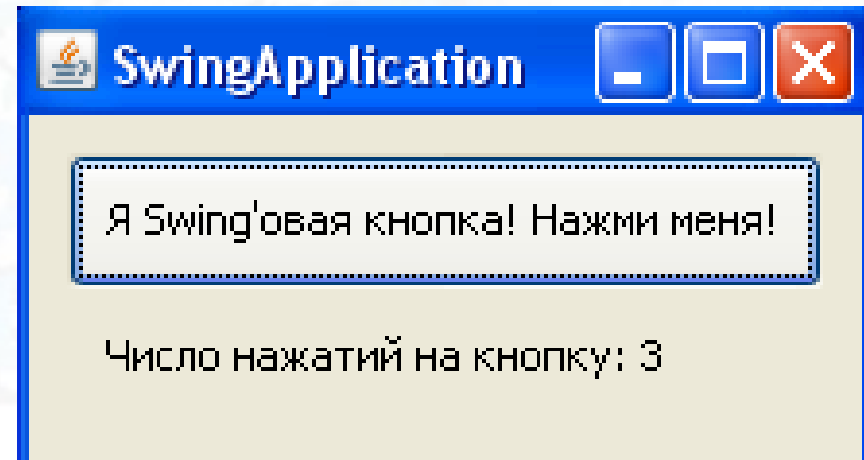
```
public static Component createComponents() {  
    // Кнопка, которую мы нажимаем  
    JButton button = new JButton("Я Swing'овая кнопка! Нажми меня!");  
    button.setMnemonic(KeyEvent.VK_S);  
    // верх, лево, низ, право.  
    Border border = BorderFactory.createEmptyBorder(10, 10, 10, 10);  
    button.setBorder(border);  
  
    // Метка, содержащая информацию о числе нажатий кнопки  
    final JLabel label = new JLabel(labelPrefix + numClicks + " ");  
    label.setLabelFor(button);  
    label.setBorder(border);  
  
    // ...  
  
    // Поместим кнопку и метку внутрь панели с внутренними границами  
    JPanel pane = new JPanel();  
    pane.setBorder(border);  
    pane.setLayout(new BoxLayout(pane, BoxLayout.PAGE_AXIS));  
    pane.add(button);  
    pane.add(label);  
  
    return pane;  
}
```

# ButtonClick!

```
public static Component createComponents()
{
    // Кнопка, которую мы нажимаем
    JButton button = new JButton("Я
    Swing'овая кнопка! Нажми меня!");
    // ...

    // Задаем действие, выполняющееся при
    // нажатии кнопки.
    button.addActionListener(new
    ActionListener() {
        public void actionPerformed(ActionEvent
        e) {
            numClicks++;
            label.setText(labelPrefix + numClicks);
        }
    });

    // ...
    pane.add(button);
    // ...
    return pane;
}
```





# Look-and-feel

```
public static void main(String[] args) {  
    try {  
        // Кросс-платформенный Look & Feel - Java-стандарт.  
        UIManager.setLookAndFeel(  
            UIManager.getSystemLookAndFeelClassName());  
    } catch (Exception e) {}  
  
    // ...  
}
```