заданьице 1:

TestFrame.java

```java
package pz_23;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

new *
public class TestFrame extends JFrame {
    // serialVersionUID для контроля версий сериализации какой-то
    no usages
    private static final long serialVersionUID = 1L;

    // текстовое поле для ввода
    4 usages
    private JTextField textField;
    // кнопки
    4 usages
    private JButton button1;
    4 usages
    private JButton button2;
    4 usages
    private JButton button3;

    // конструктор класса
    1 usage  new *
    public TestFrame() {
        super( title: "Test frame"); // устанавливаем заголовок окна
        createGUI(); // вызываем метод создания пользовательского интерфейса
    }
```

TestFrame.java

```java
    // метод создания графического интерфейса
    1 usage  new *
    public void createGUI() {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); // закрытие приложения при закрытии окна
        JPanel panel = new JPanel(); // создаем панель для компоновки элементов
        panel.setLayout(new FlowLayout()); // устанавливаем менеджер компоновки

        // создаем кнопку 1
        button1 = new JButton( text: "Button 1");
        button1.setActionCommand("Button 1 was pressed"); // устанавливаем команду действия
        panel.add(button1); // добавляем кнопку на панель

        // создаем кнопку 2
        button2 = new JButton( text: "Button 2");
        button2.setActionCommand("Button 2 was pressed");
        panel.add(button2);

        // создаем кнопку 3
        button3 = new JButton( text: "Button 3");
        button3.setActionCommand("Button 3 was pressed");
        panel.add(button3);

        // создаем текстовое поле
        textField = new JTextField();
        textField.setColumns(23); // устанавливаем количество колонок (я всё подобное использовала в калькуляторе 😇)
        panel.add(textField); // добавляем текстовое поле на панель

        // создаем слушатель действий и добавляем его ко всем кнопкам
        ActionListener actionListener = new TestActionListener();
        // добавляет слушателя действий к компоненту, который реагирует на событие
```

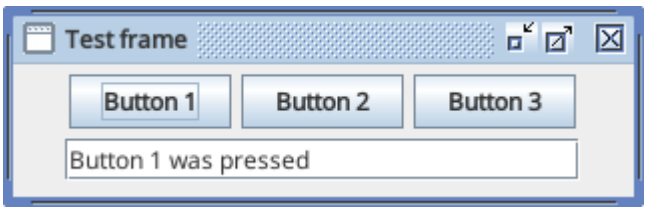
TestFrame.java

```
53          // добавляет слушателя действий к компоненту, который реагирует на событие
54          button1.addActionListener(actionListener);
55          button2.addActionListener(actionListener);
56          button3.addActionListener(actionListener);
57
58          // добавляем панель на контентное окно
59          getContentPane().add(panel);
60          setPreferredSize(new Dimension( width: 320,  height: 100)); // устанавливаем предпочтительные размеры окна
61      }
62
63      // внутренний класс для обработки действий кнопок
        1 usage   new *
64      public class TestActionListener implements ActionListener {
65          // устанавливаем текст в текстовом поле при действии кнопки
            no usages   new *
66          public void actionPerformed(ActionEvent e) {
67              textField.setText(e.getActionCommand());
68          }
69      }
70
        new *
71      public static void main(String[] args) {
72          // запускаем графический интерфейс в отдельном потоке
            new *
73          javax.swing.SwingUtilities.invokeLater(new Runnable() {
                new *
74              @Override
75              public void run() {
76                  JFrame.setDefaultLookAndFeelDecorated(true);
77                  TestFrame testFrame = new TestFrame();
78                  // устанавливает размеры окна в соответствии с заданными значениями
```

TestFrame.java

```
            no usages   new *
66          public void actionPerformed(ActionEvent e) {
67              textField.setText(e.getActionCommand());
68          }
69      }
70
        new *
71      public static void main(String[] args) {
72          // запускаем графический интерфейс в отдельном потоке
            new *
73          javax.swing.SwingUtilities.invokeLater(new Runnable() {
                new *
74              @Override
75              public void run() {
76                  JFrame.setDefaultLookAndFeelDecorated(true);
77                  TestFrame testFrame = new TestFrame();
78                  // устанавливает размеры окна в соответствии с заданными значениями
79                  testFrame.pack();
80                  // устанавливает положение окна относительно компонента-владельца (null - по центру экрана)
81                  testFrame.setLocationRelativeTo(null);
82                  testFrame.setVisible(true); // устанавливаем видимость окна
83              }
84          });
85      }
86  }
87
```

результатик:



заданьице 2:

TestFrame.java

```
62
63      // внутренний класс для обработки действий кнопок
        1usage  new*
64      public class TestActionListener implements ActionListener {
65          // устанавливаем текст в текстовом поле при действии кнопки
            1usage  new*
66 @        public void actionPerformed(ActionEvent e) {
67              textField.setText(e.getActionCommand()); // устанавливаем текст в текстовом поле
68
69              JButton button = (JButton) e.getSource(); // получаем источник события как кнопку
70
71              // выводим информацию о нажатой кнопке в консоль
72              System.out.println(button.getText() + ", " + e.getActionCommand());
73
74              // проверяем, что источник события не равен кнопке button3
75              if (e.getSource() != button3) {
76                  textField.setText(e.getActionCommand()); // устанавливаем текст в текстовом поле
77              } else {
78                  // создаем событие программно для кнопки button2
79                  ActionEvent e1 = new ActionEvent(button2, Event.MOUSE_DOWN,  command: "Круассан 2 was pressed programmatically!!!😊");
80
81                  // получаем массив слушателей для кнопки button2
82                  ActionListener[] listeners;
83                  listeners = button2.getActionListeners();
84
85                  // вызываем метод actionPerformed у первого слушателя кнопки button2
86                  listeners[0].actionPerformed(e1);
87              }
88          }
89      }
90
```

результатик поразил морзянкой:

```
/usr/lib/jvm/java-11/bin/java -javaagent:/home/divan/Down
Button 3, Button 3 was pressed
Button 2, Круассан 2 was pressed programmatically!!!😉
Button 1, Button 1 was pressed
Button 1, Button 1 was pressed
Button 2, Button 2 was pressed
Button 1, Button 1 was pressed
Button 2, Button 2 was pressed
Button 1, Button 1 was pressed
Button 2, Button 2 was pressed
Button 1, Button 1 was pressed
Button 1, Button 1 was pressed
Button 1, Button 1 was pressed
Button 1, Button 1 was pressed
Button 1, Button 1 was pressed
Button 1, Button 1 was pressed
Button 1, Button 1 was pressed
Button 2, Button 2 was pressed

Process finished with exit code 0
```