

# [GUI] Ćwiczenia VIII:

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

public class SwingApp {
    public static void main(String[] args) {
        new SwingApp();
    }
    public SwingApp() {
        // poprawny sposób uruchomienia Swing'a
        SwingUtilities.invokeLater(this::createGUI);           // () -> createGUI()
    }
    protected void createGUI() {
        // utworzenie okna typu JFrame
        JFrame jf = new JFrame();

        // określenie tytułu okna typu JFrame
        jf.setTitle("Simple Swing App");

        // obsługa zamknięcia okna typu JFrame
        jf.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        // okienka dialogowe do wprowadzenia danych
        String x = JOptionPane.showInputDialog("Podaj położenie x okna");
        String y = JOptionPane.showInputDialog("Podaj położenie y okna");

        // określenie położenia okna typu JFrame
        jf.setLocation(Integer.parseInt(x), Integer.parseInt(y));

        // dodatkowy panel do grupowania komponentów
        JPanel jp = new JPanel();

        // ustawianie zarządcy rozkładu (layout) dla panelu
        jp.setLayout(new GridLayout(2, 3, 2, 2));

        // dodawanie komponentów do panelu
        for (int i = 0; i < 6; i++)
            jp.add(new JButton(""+i));

        // ustawianie zarządcy rozkładu (layout) dla okna typu JFrame
        jf.setLayout(new FlowLayout(FlowLayout.CENTER));

        // dodawanie komponentów do okna typu JFrame
        jf.add(jp);
        jf.add(new JButton("7"));

        // nie działa
        jf.setBackground(Color.RED);
    }
}
```

```

        // działa
        //jf.getContentPane().setBackground(Color.RED);
        // ustawianie możliwości zmiany rozmiarów okna typu JFrame
        jf.setResizable(true);

        // upakowanie okna tyou JFrame
        jf.pack();

        // wyświetlenie okna typu JFrame
        jf.setVisible(true);
    }
}

```

```

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

import javax.swing.*;

public class SwingApp2 implements ActionListener {

    public static void main(String[] args) {
        new SwingApp2();
    }

    public SwingApp2() {
        // poprawny sposób uruchamiania Swing'a
        SwingUtilities.invokeLater(this::createGUI);           // () -> createGUI()
    }

    protected void createGUI() {
        // utworzenie okna typu JFrame
        JFrame jf = new JFrame();

        // określenie tytułu okna typu JFrame
        jf.setTitle("Simple Swing Event Handler");

        // obsługa zamknięcia okna typu JFrame
        jf.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        // okno wyśrodkowane na ekranie
        jf.setLocationRelativeTo(null);

        // komponent z obsługą zdarzeń (za pomocą obiektu klasy słuchacza utworzonej g
        dzie indziej)
        JButton jb1 = new JButton("1");
        jb1.addActionListener(new InnerHandler());

        // dodatkowy panel typu JPanel do grupowania komponentów
    }
}

```

```
JPanel jp = new JPanel();

// ustawianie zarządcy rozkładu dla panelu typu JPanel
jp.setLayout(new GridLayout(2, 3, 2, 2));

// dodawanie komponentów (z obsługą zdarzeń za pomocą klas anonimowych/wyrażeń
ia lambda) do panelu
for (int i = 0; i < 5; i++) {
    JButton jb = new JButton("" + (i+2));
    jb.addActionListener(
        (e) -> {
            JButton b = (JButton)(e.getSource());
            System.out.println(b.getText());
        }
    );
}

// klasa anonimowa
new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        JButton jb = (JButton)(e.getSource());
        System.out.println(jb.getText());
    }
};

jp.add(jb);
}

// komponent z obsługą zdarzeń za pomocą samej klasy SwingApp2 będącej k
lasą słuchacza
JButton jb7 = new JButton("7");
jb7.addActionListener(this);
jp.add(jb7);

// ustawianie zarządcy rozkładu dla okna typu JFrame
jf.setLayout(new FlowLayout(FlowLayout.CENTER));

// dodawanie komponentów do okna typu JFrame
jf.add(jb1);
jf.add(jp);

// ustawianie możliwości zmiany rozmiarów okna
jf.setResizable(true);

// upakowanie okna typu JFrame
jf.pack();

// wyświetlenie okna typu JFrame
jf.setVisible(true);
}

private class InnerHandler implements ActionListener {
    @Override
```

```
        public void actionPerformed(ActionEvent e) {  
            JButton jb = (JButton)(e.getSource());  
            System.out.println(jb.getText());  
        }  
    }  
  
    @Override  
    public void actionPerformed(ActionEvent e) {  
        JButton jb = (JButton)(e.getSource());  
        System.out.println(jb.getText());  
    }  
}
```