



Java SWING wprowadzenie



AWT

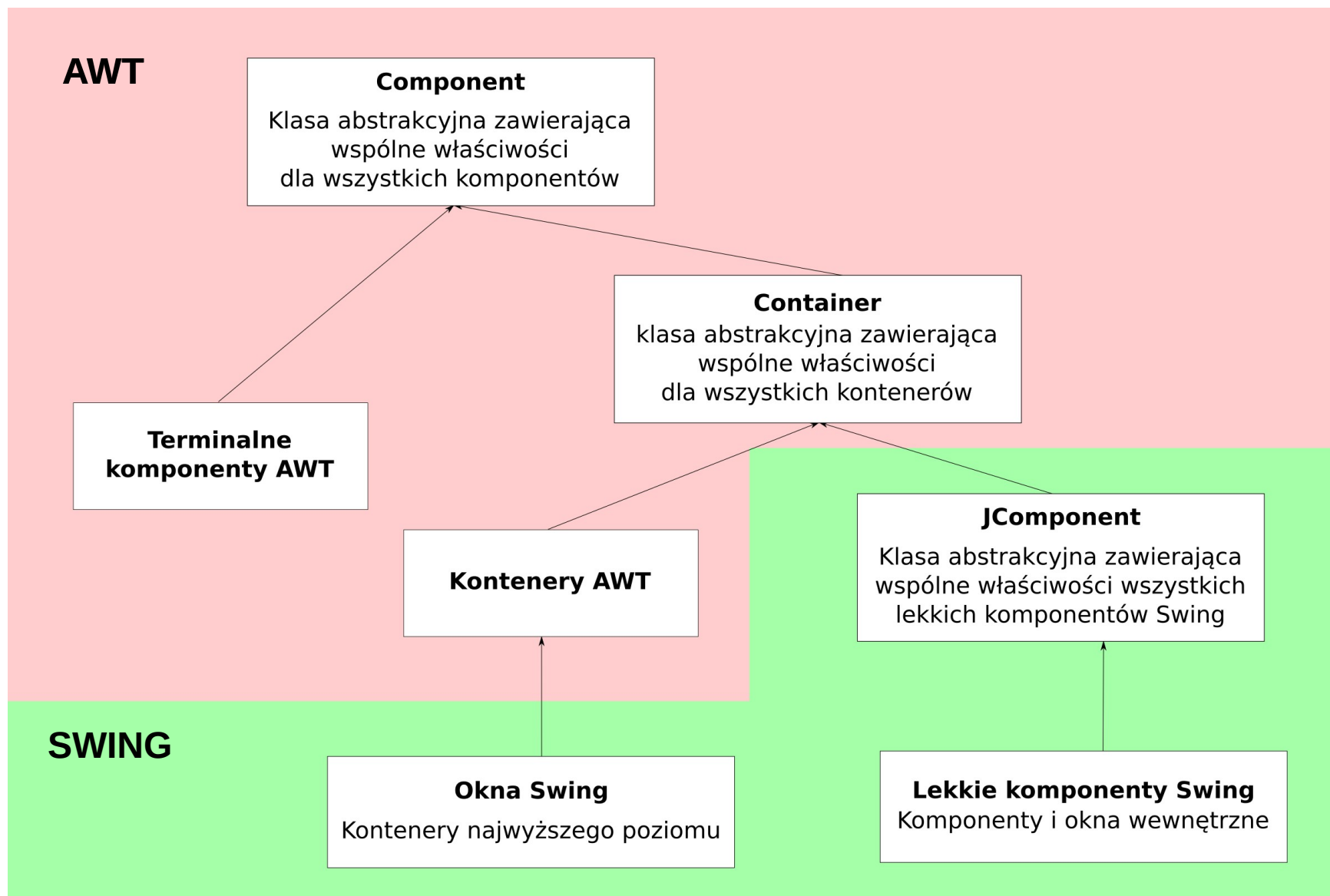
(abstract window toolkit)

- Zestaw klas definiujące proste komponenty interakcji wizualnej
- Standardowa biblioteka Javy
- Należy do pakietu `java.awt`
- Wykorzystuje komponenty ciężkie - realizowane poprzez użycie graficznych bibliotek GUI systemu operacyjnego
- Wygląd zależny od systemu operacyjnego
- Ubogie możliwości graficzne i interaktywne (brak np. tabel)

SWING

- Rozwinięcie AWT dające większy zbiór zaawansowanych komponentów
- Początkowo dodatek do JDK 1.1.8, a później włączony w skład Java 2 Platform
- Wykorzystuje komponenty lekkie
 - Mają wygląd niezależny od platformy
 - Rysowane z pomocą Javy (nie wykorzystują systemowych bibliotek GUI)
 - Mogą być przezroczyste - przybierają wizualnie dowolne kształty

Zależności



Komponenty i kontenery

Komponenty	Kontenery
<ul style="list-style-type: none">• komponenty tekstowe (TextField ,TextArea, JTextField, JTextArea)• przyciski (Button, JButton)• etykiety (Label, JLabel)• pola wyboru (Checkbox, JCheckbox)• listy (List, JList)• paski przewijania (ScrollBar, JScrollBar)• menu (MenuItem, JMenuItem)	<ul style="list-style-type: none">• ramki (Frame, JFrame)• okna (Window, JWindow)• kontenery (Container)• panele (Panel, JPanel)• okna dialogowe (Dialog, JDialog)• okna przewijalne (ScrollPane, JScrollPane)



Hello swing

```
package helloswing;

import javax.swing.JFrame;

public class HelloSwing {

    public static void main(String[] args) {

        JFrame frame = new JFrame();
        frame.setTitle("Tytuł");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        //frame.pack();
        frame.setSize(400, 300);
        frame.setVisible(true);

        Container c = frame.getContentPane();
        c.add(new JLabel("Witaj"));
    }
}
```

Takie tam...

```
//panele
JPanel p = new JPanel();

//etykieta
JLabel l = new JLabel(<tresc>);

//przyciski
JButton b = new JButton(<labelka>);
b.addActionListener(this); //wykorzystuje interfejs ActionListener

@Override
public void actionPerformed(ActionEvent e) {
    Object target = e.getSource();

    if (target == b) {

    }
}
```

Takie tam...

```
//pole tekstowe
TextField t = new TextField(<ilość kolumn>);
int k = Integer.parseInt(t.getText());

t.addKeyListener(new KeyListener() {
    @Override
    public void keyTyped(KeyEvent ke) {

    }

    @Override
    public void keyPressed(KeyEvent ke) {
        char c = ke.getKeyChar();
        System.out.println(c);
    }

    @Override
    public void keyReleased(KeyEvent ke) {

    }

});
```




Takie tam...

```
//lista
```

```
DefaultListModel model = new DefaultListModel();
```

```
model.addElement(<tresc wiersza>);
```

```
model.addElement(<tresc wiersza>);
```

```
JList l = new JList(model);
```

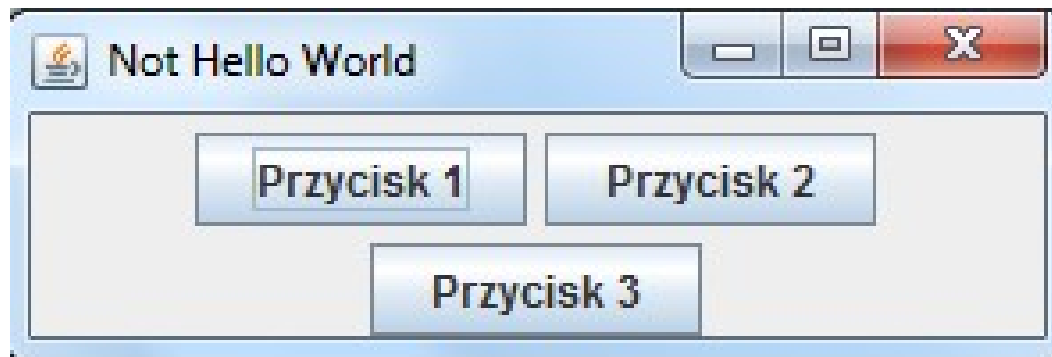
```
//okno dialogowe
```

```
JOptionPane.showMessageDialog(this,<tytul>,<tresc>,  
    JOptionPane.WARNING_MESSAGE);
```

FlowLayout

- Komponenty ułożone są w wierszu
- Zmiana rozmiaru kontenera nie zmienia rozmiarów komponentów - gdy szerokość kontenera jest za mała, pojawiają się dodatkowe wiersze

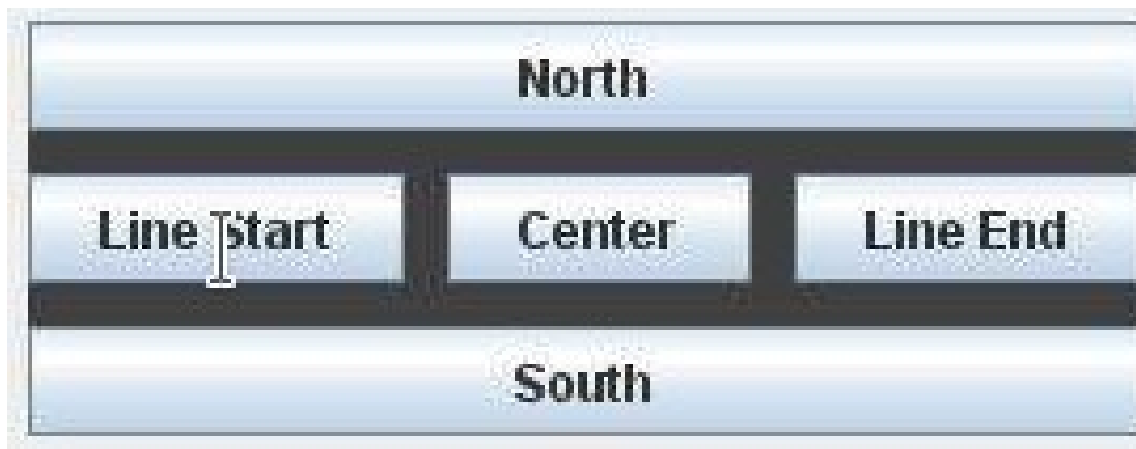
```
panel.setLayout(new FlowLayout());  
panel.add(new JButton("Przycisk 1"));  
panel.add(new JButton("Przycisk 2"));  
panel.add(new JButton("Przycisk 3"));
```



BorderLayout

- Komponenty ułożone są "geograficznie": "North", "East", "South", "West", "Center" (BorderLayout.NORTH, BorderLayout.CENTER itp)
- Komponent dodany w miejscu "Center" wypełnia całe pozostawiane przez inne komponenty miejsce

```
panel.setLayout(new BorderLayout());  
panel.add(new JButton("Center"), BorderLayout.CENTER);  
panel.add(new JButton("Line Start"), BorderLayout.WEST);  
panel.add(new JButton("Line End"), BorderLayout.EAST);  
panel.add(new JButton("North"), BorderLayout.NORTH);  
panel.add(new JButton("South"), BorderLayout.SOUTH);
```



GridLayout

- Siatka (tablica) komponentów
- Rozmiary wszystkich komponentów będą takie same i zmieniają się wraz ze zmianami rozmiaru kontenera

`new GridLayout(n, m)`

- Tworzy tablice $n \times m$ komponentów (n -wiersze, m -kolumny)
- Jeśli $n=0$ lub $m=0$, to dany wymiar tablicy zostanie ustalony dynamicznie na podstawie drugiego wymiaru i liczby komponentów w kontenerze

```
panel.setLayout(new GridLayout(0,3));  
panel.add(new JButton("Button 1"));  
panel.add(new JButton("Button 2"));  
panel.add(new JButton("Button 3"));  
panel.add(new JButton("Button 4"));  
panel.add(new JButton("Button 5"));
```

