



JAVA

Kontenery i komponenty graficzne

Opracował: Andrzej Nowak

Bibliografia: **JAVA Szkoła programowania**, D. Trajkowska
Ćwiczenia praktyczne JAVA. Wydanie III, M. Lis



Kontenery

Aplikacja okienkowa składa się z głównego elementu zwanego oknem, wewnątrz którego znajdują się pozostałe elementy aplikacji. Okno stanowi kontener na inne elementy.

Za tworzenie okna odpowiada klasa **JFrame** należąca do pakietu **javax.swing**.

Przykład aplikacji wyświetlającej okno

/zadanie1.txt/

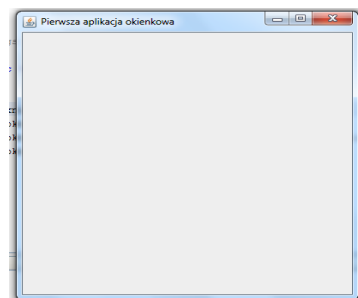
```
import javax.swing.*;

public class AplikacjaOkienkowa {
    public static void main (String args[]){
        JFrame okno = new JFrame("Pierwsza aplikacja okienkowa");
        okno.setSize(400,400);
        okno.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        okno.setVisible(true);
    }
}
```

Metoda `setSize()` umożliwia ustawienie szerokości okna.

Metoda `setDefaultCloseOperation()` umożliwia zamknięcie okna ikoną krzyżyka w pasku tytułowym po przekazaniu do metody wartości stałej `EXIT_ON_CLOSE`.

Na koniec ustawiamy widoczność okna funkcją `setVisible()`. Pobiera ona parametr typu **boolean**. Przekazanie wartości **true** powoduje wyświetlenie okna.



Zamykanie okna

Okno utworzone za pomocą `JFrame` możemy zamknąć za pomocą metody `setDefaultCloseOperation()`. Wystarczy przekazać do niej parametr stałej statycznej klasy `JFrame` lub jego odpowiednik w postaci liczbowej.

Np.:

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

jest równoważne :

```
okno.setDefaultCloseOperation(3);
```

Stała statyczna	Liczba całkowita	efekt
<code>JFrame.DO_NOTHING_ON_CLOSE</code>	0	Żadna operacja nie zostanie wykonana
<code>JFrame.HIDE_ON_CLOSE</code>	1	Okno zostanie ukryte
<code>JFrame.DISPOSE_ON_CLOSE</code>	2	Okno zostanie ukryte i zamknięte
<code>JFrame.EXIT_ON_CLOSE</code>	3	Okno oraz cała aplikacja zostaną zamknięte



Komponenty

Komponent to obiekt graficzny widoczny na ekranie pozwalający na interakcję z użytkownikiem.

Szczególnym przypadkiem komponentu jest kontener, który może zawierać inne komponenty.

W kontenerze takim jak okno możemy umieszczać inne elementy graficzne zwane komponentami, które wypełniają obszar okna i najczęściej same nie zawierają innych komponentów.

Podstawowe komponenty:

Panel - obiekt klasy JPanel – komponent, który spełnia również funkcję kontenera.

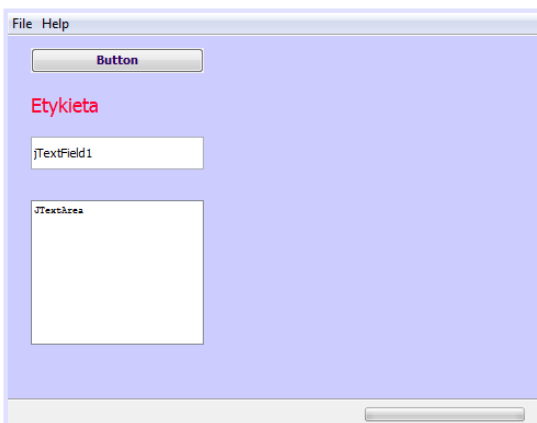
Przycisk - obiekt klasy JButton – przycisk wywołujący wykonanie jakiejś funkcji

Etykieta – obiekt klasy JLabel – napis

Pole tekstowe – obiekt klasy JTextField – umożliwia wpisywanie i wypisywanie ciągów znakowych (tekstów) oraz pobieranie ich jako ciągi znakowe.

Rozszerzone pole tekstowe – obiekt klasy JTextArea – umożliwia wprowadzanie przez użytkownika dłuższego tekstu

Menu – obiekt klasy JMenuBar – umożliwia tworzenie rozwijanego menu aplikacji

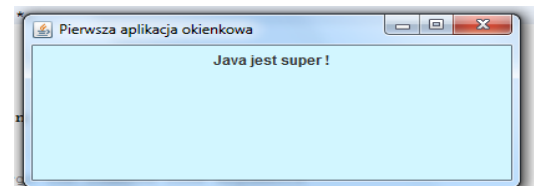


Przykład użycia komponentów –Panel i Etykieta :

/zadanie2.txt/

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

```
public class AplikacjaOkienkowa {
    public static void main (String args[]){
        JFrame okno = new JFrame("Pierwsza aplikacja okienkowa");
        okno.setSize(400,400);
        JPanel panel = new JPanel();
        panel.setBackground(new Color(210,246,255));
        JLabel etykieta = new JLabel("Java jest super !");
        panel.add(etykieta);
        okno.getContentPane().add(panel);
        okno.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        okno.setLocationRelativeTo(null);
        okno.setVisible(true);
    }
}
```



Przykład użycia komponentu – Przycisk:

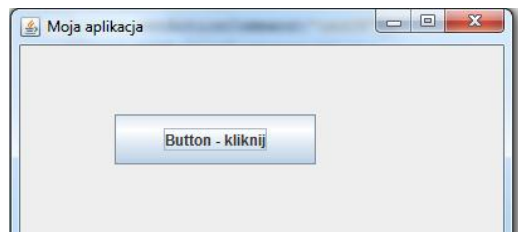
/zadanie3.txt/

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

public class Aplikacja extends JFrame implements ActionListener{
    JButton button; //deklaracja zmiennej button
    public Aplikacja() {
        super();
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(400,400);
        setTitle("Moja aplikacja");
        setLayout(null);

        button = new JButton("Button - kliknij");
        button.setActionCommand("cmdOK");
        button.addActionListener(this);
        button.setBounds(75, 55, 160, 40);
        this.add(button);

        setVisible(true);
    }
    public void actionPerformed(ActionEvent e) {
        if (e.getActionCommand().equals("cmdOK")){
            dispose();
        }
    }
    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {
            public void run(){
                new Aplikacja();
            }
        });
    }
}
```



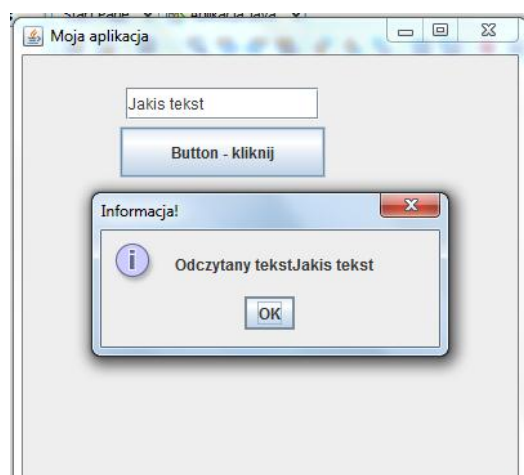
Przykład użycia komponentów

– Przycisk i Pole tekstowe:

/zadanie4.txt/

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

public class Aplikacja extends JFrame implements ActionListener{
    JButton button; //deklaracja zmiennej button
    JTextField textField; //deklaracja zmiennej textField
    public Aplikacja() {
        super();
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(400,400);
```



```
setTitle("Moja aplikacja");
setLayout(null);

button = new JButton("Button - kliknij");
button.setActionCommand("cmdOK");
button.addActionListener(this);
button.setBounds(75, 55, 160, 40);
this.add(button);

textField = new JTextField();
textField.setBounds(80, 25, 150, 25);
this.add(textField);

setVisible(true);
}
public void actionPerformed(ActionEvent e) {
    if (e.getActionCommand().equals("cmdOK")){
        String tekst = textField.getText();
        JOptionPane.showMessageDialog(this, "Odczytany tekst" + tekst, "Informacja!",
        JOptionPane.INFORMATION_MESSAGE);
    }
}
public static void main(String[] args) {
    SwingUtilities.invokeLater(new Runnable() {
        public void run(){
            new Aplikacja();
        }
    });
}
```