

Kurs programowania

Wykład 4

Wojciech Macyna

23 marca 2016

Graficzny interfejs użytkownika - GUI

W Javie możemy skorzystać z dwóch bibliotek do tworzenia graficznych interfejsów:

AWT (Abstract Windowing Toolkit) – podstawowa biblioteka będąca interfejsem do graficznych funkcji systemu. Stąd wygląd komponentów zależy od systemu.

Biblioteka: `java.awt.*`

SWING – biblioteka w Javie niezależna od systemu (ale wykorzystująca niektóre elementy AWT). Stąd identyczny wygląd GUI na różnych platformach. Komponenty mają podobne nazwy jak w AWT ale poprzedzone literą J (np. Frame w AWT i JFrame w SWING).

Biblioteka: `javax.swing.*`

Na wykładzie zostanie omówiona głównie biblioteka AWT.

Główne konstruktory

- `public Frame()`
- `public Frame(String title)`
- `public Frame(String title, GraphicsConfiguration gc)`

Główne metody

- `public void setSize(int width, int height)` [domyślny rozmiar 20x20]
- `public void setBounds(int x, int y, int width, int height)`
- `public void pack()` [dopasowuje rozmiar okna do zawartości]
- `public void setVisible(boolean show)`

Kontenery i komponenty

W GUI rozróżniamy dwa podstawowe rodzaje elementów: komponenty i zawierające (wyświetlające) je kontenery.

Podstawowe typy komponentów AWT

Button, CheckBox, Choice, Label, List, TextField, TextArea, Menu, ...

Podstawowe typy kontenerów AWT

Frame, Panel, ScrollPane

Uwaga: Panel jest także komponentem.

Dodawanie komponentu do konteneru

Metoda `public Component add(Component c)`

Przykład

przyklad1.java

```
1 import java.awt.*;
2
3 public class przyklad1 {
4     public static void main(String[] args) {
5         Frame okno = new Frame("Przyklad_1");
6         Button przycisk = new Button("Przycisk");
7
8         okno.setBounds(100,100,640,480);
9         okno.add(przycisk);
10        okno.setVisible(true);
11    }
12 }
```

Kontener powinien użyć zarządcy wyglądu ustawianego metodą:
`public void setLayout(LayoutManager m)`

Podstawowe wyglądy

- `java.awt.FlowLayout`
- `java.awt.BorderLayout`
- `java.awt.GridLayout`
- `java.awt.GridBagLayout`
- `java.awt.CardLayout`

Przykład użycia

flowDemo.java

```
1 import java.awt.*;
2 public class flowDemo {
3     public static void main(String[] args) {
4         Frame okno = new Frame("flowDemo");
5         Button l1 = new Button("Napis1");
6         Button l2 = new Button("Napis2");
7         Button l3 = new Button("Napis3");
8         Button l4 = new Button("Napis4");
9
10        okno.setBounds(100,100,200,200);
11        okno.setLayout(new FlowLayout(FlowLayout.CENTER));
12        okno.add(l1);
13        okno.add(l2);
14        okno.add(l3);
15        okno.add(l4);
16        okno.setVisible(true);
17    }
18 }
```

Przykład użycia

gridDemo.java

```
1  import java.awt.*;
2  public class gridDemo {
3      public static void main(String[] args) {
4          Frame okno = new Frame("gridDemo");
5          Button l1 = new Button("Napis1");
6          Button l2 = new Button("Napis2");
7          Button l3 = new Button("Napis3");
8          Button l4 = new Button("Napis4");
9
10         okno.setBounds(100,100,200,200);
11         okno.setLayout(new GridLayout(2,2));
12         okno.add(l1);
13         okno.add(l2);
14         okno.add(l3);
15         okno.add(l4);
16         okno.setVisible(true);
17     }
18 }
```


Przykład użycia

borderDemo.java

```
1 import java.awt.*;
2 public class borderDemo {
3     public static void main(String[] args) {
4         Frame okno = new Frame("borderDemo");
5         Button l1 = new Button("Napis1");
6         Button l2 = new Button("Napis2");
7         Button l3 = new Button("Napis3");
8         Button l4 = new Button("Napis4");
9         Button l5 = new Button("Napis5");
10
11         okno.setBounds(100,100,200,200);
12         okno.setLayout(new BorderLayout());
13         okno.add(l1,BorderLayout.NORTH);
14         okno.add(l2,BorderLayout.SOUTH);
15         okno.add(l3,BorderLayout.WEST);
16         okno.add(l4,BorderLayout.EAST);
17         okno.add(l5,BorderLayout.CENTER);
18         okno.setVisible(true);
19     }
20 }
```

Delegacyjny model zdarzeń

- Źródło zdarzenia – jaki komponent jest źródłem zdarzenia.
- Słuchacz zdarzenia – kto słucha i jaką metodę powinien wywołać.
- Interfejs – zbiór metod odpowiedzialnych za obsługę zdarzeń.

Przykładowe interfejsy obsługi zdarzeń

- `java.awt.event.ActionListener` z metodą
`public void actionPerformed(ActionEvent e)`
- `java.awt.event.WindowListener` z metodami:
`public void windowOpened(WindowEvent e)`
`public void windowClosing(WindowEvent e)`
`public void windowClosed(WindowEvent e)`
`public void windowIconified(WindowEvent e)`
`public void windowDeiconified(WindowEvent e)`
`public void windowActivated(WindowEvent e)`
`public void windowDeactivated(WindowEvent e)`

Klasy implementujące interfejsy obsługi zdarzeń z pustymi metodami.

Podstawowe klasy adapterów

- WindowAdapter
- ComponentAdapter
- ContainerAdapter
- FocusAdapter
- KeyAdapter
- MouseAdapter

Przykład użycia adaptera

przyklad2.java

```
1 import java.awt.*;
2 import java.awt.event.*;
3 class WindowAdapterDemo extends WindowAdapter {
4     Label p;
5     WindowAdapterDemo(Label p) { this.p = p; }
6     public void windowClosing(WindowEvent e) { System.exit(0); }
7     public void windowActivated(WindowEvent e) {
8         p.setText("Aktywne"); }
9     public void windowDeactivated(WindowEvent e) {
10        p.setText("Nie_aktywne"); }
11 }
12 public class przyklad2 {
13     public static void main(String[] args) {
14         Frame okno = new Frame("Przyklad_2");
15         Label napis = new Label("", Label.CENTER);
16         napis.setBackground(Color.RED);
17         napis.setForeground(Color.BLUE);
18         napis.setFont(new Font(Font.SERIF, Font.BOLD, 100));
19         okno.addWindowListener(new WindowAdapterDemo(napis));
20         okno.setBounds(100, 100, 640, 240);
21         okno.add(napis);
22         okno.setVisible(true);
23     }
24 }
```

Klasa `java.awt.Button`

Główne konstruktory

- `public Button()`
- `public Button(String label)`

Główne metody

- `public void addActionListener(ActionListener a)`
- `public void setActionCommand(String c)`

Główne konstruktory

- `public Label()`
- `public Label(String text)`
- `public Label(String text, int alignment)` [wyrównanie za pomocą stałych `Label.LEFT`, `Label.RIGHT`, `Label.CENTER`]

Główne metody

- `public void setText(String text)`
- `public String getText()`

Klasa `java.awt.TextField`

Główne konstruktory

- `public TextField()`
- `public TextField(String text)`
- `public TextField(int columns)`
- `public TextField(String text, int columns)`

Główne metody

- `public String getText()`
- `public void setText(String text)`
- `public void setEditable(boolean b)`

Przykład

Program.java (1/3)

```
1 import java.awt.*;
2 import java.awt.event.*;
3 class MyWindowAdapter extends WindowAdapter {
4     public void windowClosing(WindowEvent e) { System.exit(0); }
5 }
6 class EndButtonAdapter implements ActionListener {
7     public void actionPerformed(ActionEvent e) { System.exit(0); }
8 }
9 class EndButton extends Button {
10     EndButton() {
11         super("Koniec"); addActionListener(new EndButtonAdapter()); }
12 }
13 class MyButtonAdapter implements ActionListener {
14     Program p;
15     MyButtonAdapter(Program p) { this.p = p; }
16     public void actionPerformed(ActionEvent e) { p.action(); }
17 }
18 class MyButton extends Button {
19     MyButton(Program p) {
20         super("Przepisz"); addActionListener(new MyButtonAdapter(p)); }
21 }
```


Przykład cd.

Program.java (2/3)

```
22 class MyFrame extends Frame {
23     MyFrame(Program p) {
24         super("Program");
25         setBounds(100,100,640,480);
26         addWindowListener(new MyWindowAdapter());
27         setFont(new Font(Font.SANS_SERIF,Font.PLAIN,40));
28         setLayout(new GridLayout(4,1));
29
30         EndButton koniec = new EndButton();
31         MyButton akcja = new MyButton(p);
32         p.wynik = new Label();
33         p.dane = new TextField(40);
34         add(p.dane);
35         add(akcja);
36         add(p.wynik);
37         add(koniec);
38
39         pack();
40         setResizable(false);
41     }
42 }
```

Program.java (3/3)

```
43 public class Program {  
44     MyFrame frame;  
45     Label wynik;  
46     TextField dane;  
47  
48     void action() {  
49         wynik.setText(dane.getText());  
50         dane.setText("");  
51     }  
52     public static void main(String[] args) {  
53         Program p = new Program();  
54         p.frame = new MyFrame(p);  
55         p.frame.setVisible(true);  
56     }  
57 }
```