

# Programowanie obiektowe

## Wykład 12: GUI. Aplety.

5/19/2013

S.Deniziak:Programowanie obiektowe

1

## Pakiety graficzne w Javie

- AWT (Abstract Window Toolkit)
  - Java 1.1
- Swing
  - Java 2.0
- Java 2D
- Java 3D
- Java Media Framework
- Java Advance Imaging

5/19/2013

S.Deniziak:Programowanie obiektowe

2

## Struktura aplikacji graficznej

```
import javax.swing.*;
```

Importowanie pakietów:

```
javax.swing  
javax.swing.event  
java.awt  
java.awt.event
```

5/19/2013

S.Deniziak:Programowanie obiektowe

3

## Struktura aplikacji graficznej, cd.

```
import javax.swing.*;  
public class HelloWorldSwing {  
    public static void main(String[] args) {  
        JFrame.setDefaultLookAndFeelDecorated(true);  
        JFrame frame = new JFrame("HelloWorldSwing");  
    }  
}
```

Utworzenie głównego kontenera:  
JFrame, JDialog, JApplet

5/19/2013

S.Deniziak:Programowanie obiektowe

4

## Struktura aplikacji graficznej, cd.

```
import javax.swing.*;
public class HelloWorldSwing {
    public static void main(String[] args) {
        JFrame.setDefaultLookAndFeelDecorated(true);
        JFrame frame = new JFrame("HelloWorldSwing");
        final JLabel label = new JLabel("Hello World");
        frame.getContentPane().add(label);

    }
}
```

Utworzenie i rozmieszczenie komponentów:  
JLabel, JButton, JTextField, JMenu, ...

5/19/2013

S.Deniziak:Programowanie obiektowe

5

## Struktura aplikacji graficznej, cd.

```
import javax.swing.*;
public class HelloWorldSwing {
    public static void main(String[] args) {
        JFrame.setDefaultLookAndFeelDecorated(true);
        JFrame frame = new JFrame("HelloWorldSwing");
        final JLabel label = new JLabel("Hello World");
        frame.getContentPane().add(label);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    }
}
```

Dodanie obsługi zdarzeń:

```
frame.addWindowListener(new WindowAdapter() {
    public void windowClosing(WindowEvent e) {
        System.exit(0);
    }
});
```

5/19/2013

S.Deniziak:Programowanie obiektowe

6

## Struktura aplikacji graficznej, cd.

```
import javax.swing.*;
public class HelloWorldSwing {
    public static void main(String[] args) {
        JFrame.setDefaultLookAndFeelDecorated(true);
        JFrame frame = new JFrame("HelloWorldSwing");
        final JLabel label = new JLabel("Hello World");
        frame.getContentPane().add(label);
        frame setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.pack();
        frame.setVisible(true);
    }
}
```

Wyświetlenie okna:



5/19/2013

S.Deniziak:Programowanie obiektowe

7

## Rozmieszczanie komponentów w kontenerach

- Kontenery:
  - główne: **JFrame, JDialog, JApplet**
  - pośrednie: **JPanel,**
- Rozmieszczanie komponentów:
  - menadżer rozmieszczenia (**LayoutManager**):

[BorderLayout](#)    [BoxLayout](#)  
[CardLayout](#)    [FlowLayout](#)  
[GridBagLayout](#)    [GridLayout](#)  
[SpringLayout](#)

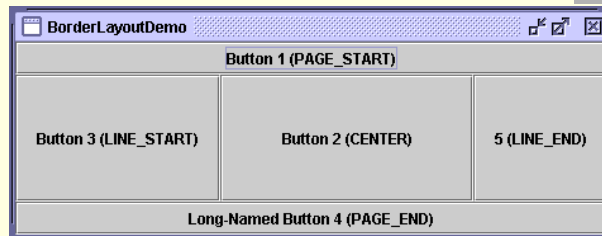
```
np.contentPane.setLayout(new FlowLayout());
```

5/19/2013

S.Deniziak:Programowanie obiektowe

8

## BorderLayout (domyślny)



```
JButton button = new JButton("Button 1 (PAGE_START)");
pane.add(button, BorderLayout.PAGE_START);
button = new JButton("Button 2 (CENTER)");
button.setPreferredSize(new Dimension(200, 100));
pane.add(button, BorderLayout.CENTER);
button = new JButton("Button 3 (LINE_START)");
pane.add(button, BorderLayout.LINE_START);
button = new JButton("Long-Named Button 4 (PAGE_END)");
pane.add(button, BorderLayout.PAGE_END);
button = new JButton("5 (LINE_END)");
pane.add(button, BorderLayout.LINE_END);
```

5/19/2013

S.Deniziak:Programowanie obiektowe

9

## Obsługa zdarzeń

### ■ Interfejs **ActionListener**

```
public class Beeper ... implements ActionListener {
    ... //inicjalizacja
    button.addActionListener(this);
    ... public void actionPerformed(ActionEvent e) {
        ...// Akcja
    }
}
```

### ■ klasa **ActionEvent**

```
String getActionCommand()
int getModifiers()
Object getSource()
```

5/19/2013

S.Deniziak:Programowanie obiektowe

10

# Obsługa klawiatury

Interfejs	Klasa implementująca	Metody
<b>KeyListener</b>	<b>KeyAdapter</b>	<b>keyPressed(KeyEvent)</b> <b>keyReleased(KeyEvent)</b> <b>keyTyped(KeyEvent)</b>

np. `element.addKeyListener(new KeyAdapter());`  
`element.removeKeyListener(new KeyAdapter());`

5/19/2013

S.Denziak:Programowanie obiektowe

11

# Obsługa myszki

Interfejs	Klasa implementująca	Metody
<b>MouseListener</b>	<b>MouseAdapter</b>	<b>mouseClicked(MouseEvent)</b> <b>mouseEntered(MouseEvent)</b> <b>mouseExited(MouseEvent)</b> <b>mousePressed(MouseEvent)</b> <b>mouseReleased(MouseEvent)</b>
<b>MouseMotionListener</b>	<b>MouseMotionAdapter</b>	<b>mouseDragged(MouseEvent)</b> <b>mouseMoved(MouseEvent)</b>
<b>MouseListener</b>	<b>MouseListener</b>	<b>-  -</b>

np. `element.addMouseListener(new MouseAdapter());`  
`element.removeMouseListener(new MouseAdapter());`

5/19/2013

S.Denziak:Programowanie obiektowe

12

# Grafika użytkownika

Klasa **JComponent**



**Graphics2D**

```
protected void paintComponent(Graphics g) {  
    ...  
    Insets insets = getInsets();  
    int currentWidth = getWidth() - insets.left - insets.right;  
    int currentHeight = getHeight() - insets.top - insets.bottom;  
    ... ..  
    Rysowanie: g.drawLine(),g.drawRect(), g.drawArc(),  
              g.drawString(), g.fillRect(), g.setColor(), itp..  
}
```

5/19/2013

S.Deniziak:Programowanie obiektowe

13

# Struktura apletu

**MojApplet.class:**

```
public class MojApplet extends JApplet {  
    public void init() {}  
    public void start() {}  
    public void stop() {}  
    public void destroy() {}  
}
```

**MojApplet.html:**

```
<APPLET CODE=MojApplet.class WIDTH=200 HEIGHT=100>  
</APPLET>
```

**appletviewer MojApplet.html**

5/19/2013

S.Deniziak:Programowanie obiektowe

14

## Przykład

```
// <applet code=SineWave width=700 height=400></applet>
import javax.swing.*;
import javax.swing.event.*;
import java.awt.*;

class SineDraw extends JPanel {
    private static final int SCALEFACTOR = 200;
    private int cycles;
    private int points;
    private double[] sines;
    private int[] pts;
    public SineDraw() { setCycles(5); }
    public void setCycles(int newCycles) {
        cycles = newCycles;
        points = SCALEFACTOR * cycles * 2;
        sines = new double[points];
        for(int i = 0; i < points; i++) {
            double radians = (Math.PI/SCALEFACTOR) * i;
            sines[i] = Math.sin(radians);
        }
        repaint();
    }
}
5/19/2013
```

S.Deniziak:Programowanie obiektowe

15

## Przykład, cd.

```
public void paintComponent(Graphics g) {
    super.paintComponent(g);
    int maxWidth = getWidth();
    double hstep = (double)maxWidth/(double)points;
    int maxHeight = getHeight();
    pts = new int[points];
    for(int i = 0; i < points; i++)
        pts[i] =
            (int)(sines[i] * maxHeight/2 * .95 + maxHeight/2);
    g.setColor(Color.RED);
    for(int i = 1; i < points; i++) {
        int x1 = (int)((i - 1) * hstep);
        int x2 = (int)(i * hstep);
        int y1 = pts[i-1];
        int y2 = pts[i];
        g.drawLine(x1, y1, x2, y2);
    }
}
}
```

5/19/2013

S.Deniziak:Programowanie obiektowe

16



## Przykład, cd.

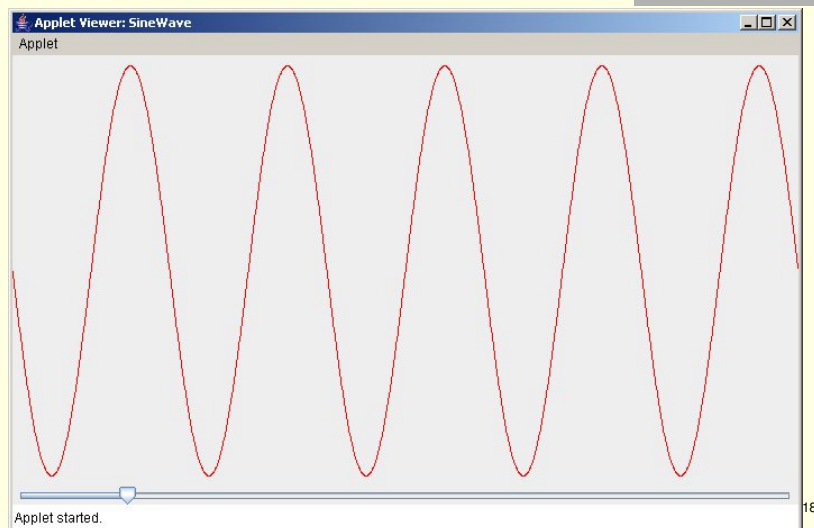
```
public class SineWave extends JApplet {
    private SineDraw sines = new SineDraw();
    private JSlider adjustCycles = new JSlider(1, 30, 5);
    public void init() {
        Container cp = getContentPane();
        cp.add(sines);
        adjustCycles.addChangeListener(new ChangeListener() {
            public void stateChanged(ChangeEvent e) {
                sines.setCycles(
                    ((JSlider)e.getSource()).getValue());
            }
        });
        cp.add(BorderLayout.SOUTH, adjustCycles);
    }
}
```

5/19/2013

S.Deniziak:Programowanie obiektowe

17

## Przykład, cd.



# Aplety w XHTML

```
<?xml version="1.0" encoding="ISO-8859-2"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="pl">
<head>
<meta http-equiv="Content-type" content="text/html; charset=iso-8859-2" />
<title>tytuł strony </title>
</head>
<body>
<p> <table cellpadding="10">
<tr> <td style="background-color:black;">
<applet code="SineWave.class" width="600" height="400"> <param name="pola"
value="5" /> Uwaga! Twoja przeglądarka nie widzi appletu!
</applet>
</td></tr>
</table> </p>
</body>
</html>
```

# Aplety złożone z kilku klas

- jar cf MojAplet.jar \*.class
- <head><title>Mój Aplet </title></head>  
<body> <applet code=MojAplet.class archive=MojAplet.jar  
width=200 height=100>  
</applet> </body>

## Ograniczenia apletów

- Łączność tylko z serwerem HTTP
- Brak dostępu do lokalnego systemu plików
- Brak możliwości odczytu własności systemowych
- Nie można ładować bibliotek ani uruchamiać metod typu „native”

## Podpisywanie apletów (1)

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

public class FileAccessApplet extends JApplet {
    private JTextField
        filename = new JTextField(),
        dir = new JTextField();
    private JButton
        open = new JButton("Open"),
        save = new JButton("Save");
    private JEditorPane ep = new JEditorPane();
    private JScrollPane jsp = new JScrollPane();
    private File file;
    public void init() {
        JPanel p = new JPanel();
        open.addActionListener(new OpenL());
        p.add(open);
        save.addActionListener(new SaveL());
        p.add(save);
        Container cp = getContentPane();
        jsp.getViewport().add(ep);
        cp.add(jsp, BorderLayout.CENTER);
        cp.add(p, BorderLayout.SOUTH);
    }
}
```

```
dir.setEditable(false);
save.setEnabled(false);
ep.setContentType("text/html");
filename.setEditable(false);
p = new JPanel();
p.setLayout(new GridLayout(2, 1));
p.add(filename);
p.add(dir);
cp.add(p, BorderLayout.NORTH);
}
```

## Podpisywanie apletów (2)

```
class OpenL implements ActionListener {
    public void actionPerformed(ActionEvent e) {
        JFileChooser c = new JFileChooser();
        c.setFileFilter(new TextFileFilter());
        // Demonstrate "Open" dialog:
        int rVal = c.showOpenDialog(FileAccessApplet.this);
        if(rVal == JFileChooser.APPROVE_OPTION) {
            file = c.getSelectedFile();
            filename.setText(file.getName());
            dir.setText(c.getCurrentDirectory().toString());
            try {
                System.out.println("Url is " + file.toURL());
                ep.setPage(file.toURL());
                // ep.repaint();
            } catch (IOException ioe) {
                throw new RuntimeException(ioe);
            }
        }
        if(rVal == JFileChooser.CANCEL_OPTION) {
            filename.setText("You pressed cancel");
            dir.setText("");
        } else {
            save.setEnabled(true);
        }
    }
}
```

```
class SaveL implements ActionListener {
    public void actionPerformed(ActionEvent e) {
        JFileChooser c = new JFileChooser(file);
        c.setSelectedFile(file);
        // Demonstrate "Save" dialog:
        int rVal =
        c.showSaveDialog(FileAccessApplet.this);
        if(rVal == JFileChooser.APPROVE_OPTION) {
            filename.setText(c.getSelectedFile().getName());
            dir.setText(c.getCurrentDirectory().toString());
            try {
                FileWriter fw = new FileWriter(file);
                ep.write(fw);
            } catch (IOException ioe) {
                throw new RuntimeException(ioe);
            }
        }
        if(rVal == JFileChooser.CANCEL_OPTION) {
            filename.setText("You pressed cancel");
            dir.setText("");
        }
    }
}
```

rogramowanie obiektowe

23

## Podpisywanie apletów (3)

```
public class TextFileFilter extends
    javax.swing.filechooser.FileFilter {
    public boolean accept(File f) {
        return f.getName().endsWith(".txt")
            || f.isDirectory();
    }
    public String getDescription() {
        return "Text Files (*.txt)";
    }
}
}
```

```
jar cf fileaccess.jar *.class
```

5/19/2013

S.Deniziak:Programowanie obiektowe

24

## Podpisywanie apletów

1. Utworzenie pliku JAR:

```
jar cf aplet.jar *.class
```

2. Utworzenie klucza:

```
keytool -genkey -alias <keyname> -keystore <url>
```

```
keytool -list -keystore <url>
```

3. Podpisanie pliku JAR:

```
jarsigner -keystore <url> <jarfile> <keyname>
```

## Pytania

1. Które metody są wystarczające w aplecie?
2. Na czym polega programowanie zdarzeniowe?
3. Co realizuje menadżer rozmieszczenia?
4. Zastosowanie dziedziczenia w tworzeniu GUI.
5. Zastosowanie polimorfizmu w tworzeniu GUI.

Koniec