

PROGRAMOWANIE W JĘZYKU JAVA

Prowadzący: dr inż. Jan Prokop

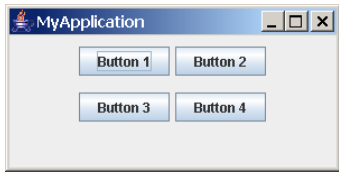
e-mail: jprokop@prz.edu.pl, Politechnika Rzeszowska, Wydział Elektrotechniki i Informatyki

LABORATORIUM 3

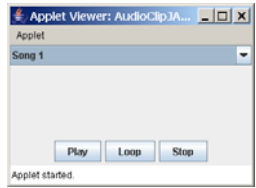
Temat: Budowa GUI - Kontenery, komponenty, układy rozkładów, menu

1. Dodawanie komponentów do kontenerów, właściwości komponentów

1.a. Aplikacja Swing



Kod przykładowej aplikacji	Zadania
<pre>import java.awt.*; import javax.swing.*; public class AddComponents extends JFrame { AddComponents() { setTitle("AddComponents"); JPanel p = new JPanel(); p.setBackground(Color.RED); JButton b1 = new JButton("Button 1"); JButton b2 = new JButton("Button 2"); JButton b3 = new JButton("Button 3"); JButton b4 = new JButton("Button 4"); p.add(b1); p.add(b2); p.add(b3); p.add(b4); getContentPane().add(p); } public static void main(String[] args){ AddComponents frame = new AddComponents(); frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); frame.setSize(300, 150); frame.setVisible(true); } }</pre>	<p>1. Dodać przyciski bezpośrednio do okna aplikacji (bez pośrednictwa panelu)</p> <p>2. Dodać przyciski tak aby otrzymać rozkład jak na rysunku (bez menedżera rozkładu)</p>  <p>3. Zbadać właściwości przycisku</p> <pre>b.setForeground(); b.setBackground(); Dimension dim = new Dimension(100,80); b.setPreferredSize(dim); b.setHorizontalAlignment(); b.setVerticalAlignment(); b.setEnabled();</pre> <p>4. Przebudować strukturę aplikacji tak aby kod związany z GUI był w oddzielnej klasie</p>

1.b. Aplet AWT


Kod przykładowego apletu	Zadania
<pre>import java.awt.*; import java.applet.*; public class ComponentsApplet extends Applet { public void init() { setBackground(Color.yellow); add(new Label("User:")); add(new TextField("Jan Prokop",25)); add(new Label("Password:")); TextField tf = new TextField(25); tf.setEchoChar('*'); add(tf); TextArea ta = new TextArea("Text", 5, 20); add(ta); add(new Button("Button 1")); add(new Button("Button 2")); add(new Button("Button 3")); add(new Checkbox("Option 1")); add(new Checkbox("Option 2", true)); add(new Checkbox("Option 3")); CheckboxGroup cbg = new CheckboxGroup(); add(new Checkbox("Opcja 1", false, cbg)); add(new Checkbox("Opcja 2", true, cbg)); add(new Checkbox("Opcja 3", false, cbg)); } }</pre>	<p>1. Dodać do apletu komponenty klas List, Choice i inne.</p> <p>2. Dodać do apletu suwaki</p> <pre>Scrollbar sb = new Scrollbar(Scrollbar.HORIZONTAL, 1, 10, 1, 100); add(sb); Scrollbar sb = new Scrollbar(Scrollbar.VERTICAL, 1, 10, 1, 200); add(sb);</pre> <p>3. Przepisać aplet z komponentami biblioteki Swing, zastosować metodę getContentPane().</p> <p>4. Zaprojektować aplet o wyglądzie</p> 

2. Podstawowe rozkłady komponentów z biblioteki AWT

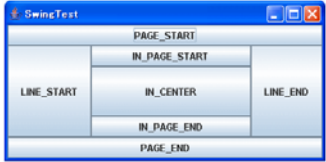

2.a. Rozkład ciągly - FlowLayout i wyłączenie domyślnego rozkładu

Rozkład ciągly - FlowLayout	Brak rozkładu - Null Layout
<p>1. Napisać aplikację dodając sześć przycisków i zbadać działanie menedżera układu FlowLayout</p> <pre>FlowLayout layout = new FlowLayout(FlowLayout.RIGHT, 5, 10); setLayout(layout); FlowLayout layout = new FlowLayout(); layout.setHgap(10); layout.setVgap(5);</pre> <p>2. Zrealizować rozkład</p> 	<p>1. Wyłączyć domyślnego menedżera układu i dodać przyciski zmieniając ich rozmiar i położenie</p> <pre>setLayout(null); setLocation(10,160); setSize(50,50); setBounds(180,160,50,50); setPreferredSize(new Dimension(100,80)); setHorizontalAlignment(Swing Constants.RIGHT);</pre> <p>2. Zrealizować rozkład</p> 


2.b. Rozkład siatkowy - GridLayout

Kod przykładowego apletu	Zadania
<pre>import java.awt.*; public class Grid extends java.applet.Applet { public void init() { setLayout(new GridLayout(3,2,10,5)); add(new Button("Button 1"));add(new Button("Button 2")); add(new Button("Button 3"));add(new Button("Button 4")); add(new Button("Button 5"));add(new Button("Button 6")); } }</pre>	<p>1. Sprawdzić podziały GridLayout(0,3) GridLayout(3,0)</p> <p>2. Zrealizować układ</p> 

2.c. Rozkład brzegowy - BorderLayout

Kod przykładowego apletu	Zadania
<pre>import java.awt.*; public class Border extends java.applet.Applet { public void init() { Button b1 = new Button("Wschód"); Button b2 = new Button("Zachód"); Button b3 = new Button("Północ"); Button b4 = new Button("Południe"); Button b5 = new Button("Środek"); setLayout(new BorderLayout(10,10)); add(b3, BorderLayout.NORTH); add(b4, BorderLayout.SOUTH); add(b1, BorderLayout.EAST); add(b2, BorderLayout.WEST); add(b5, BorderLayout.CENTER); } }</pre>	<p>1. Zrealizować aplikację o wyglądzie</p>  <p>2. Zrealizować GUI kalkulatora</p> 

2.d. Inne zadania

Zadanie 1 - Napisać aplikację (aplet) z rozkładem kart - CardLayout	Zadanie 2 - Zaprojektować własny układ rozkładu z komponentami na okręgu (jak na rysunku), zastosować interfejs LayoutManager
<pre>Panel cards = new Panel(); CardLayout cl = new CardLayout(); cards.setLayout(cl); cards.add(label1, "Pierwsza"); cards.add(label2, "Druga"); cl.show(cards, "Pierwsza"); cl.show(cards, "Druga"); cl.first(cards); cl.next(cards); cl.previous(cards);</pre>	

3. Podstawowe rozkłady komponentów z biblioteki AWT - układ GridBagLayout

3.a. GridBagLayout - przykład aplikacji

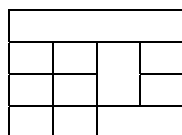
```
import java.awt.* ; import javax.swing.* ;
public class GridBagLayoutDemo extends JFrame {
    GridBagLayoutDemo(String title){
        setTitle(title);
        GridBagLayout gridbag = new GridBagLayout();
        GridBagConstraints constraints = new GridBagConstraints();
        JPanel panel = new JPanel();
        panel.setLayout(gridbag);
        constraints.ipadx = 10;
        constraints.ipady = 10;
        //constraints.insets = new Insets(15, 5, 15, 5);
        constraints.fill = GridBagConstraints.BOTH;
        constraints.weightx = 0.0;
        constraints.weighty = 0.0;
        constraints.gridx = 0;
        constraints.gridy = 0;
        constraints.gridwidth = 1;
        constraints.gridheight = GridBagConstraints.REMAINDER;
        JButton b1 = new JButton("Button1");
        gridbag.setConstraints(b1, constraints);
        panel.add(b1);
        constraints.gridx = GridBagConstraints.RELATIVE;
        constraints.gridheight = 1;
        JButton b2 = new JButton("Button2");
        gridbag.setConstraints(b2, constraints);
        panel.add(b2);
        JButton b3 = new JButton("Button3");
        gridbag.setConstraints(b3, constraints);
        panel.add(b3);
        JButton b4 = new JButton("Button4");
        gridbag.setConstraints(b4, constraints);
        panel.add(b4);
        constraints.gridheight = GridBagConstraints.REMAINDER;
        JButton b5 = new JButton("Button5");
        panel.add(b5, constraints);
        constraints.gridx = 1;
        constraints.gridy = 1;
        constraints.gridwidth = GridBagConstraints.RELATIVE ;
        constraints.gridheight = 1;
        JButton b6 = new JButton("Button6");
        panel.add(b6, constraints);
        constraints.gridy = GridBagConstraints.RELATIVE;
        JButton b7 = new JButton("Button7");
        panel.add(b7, constraints);
        getContentPane().add(panel, BorderLayout.CENTER);
    }
    public static void main(String[] args){
        GridBagLayoutDemo demo = new GridBagLayoutDemo("GridBagLayoutDemo");
        demo.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        demo.pack();
        demo.setVisible(true);
    }
}
```

3.b. **Zadania** - zaprojektować rozkłady elementów aplikacji i apletów według poniższych schematów.

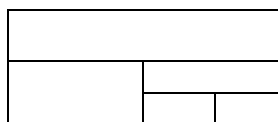
a)



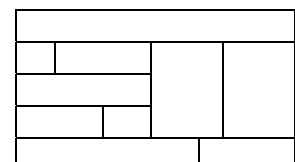
b)



c)



d)



4. Budowa menu, belki narzędziowej i paska statusu - pakiet Swing

4.a. Menu - przykład apletu

```
import java.awt.*;
import javax.swing.*;
public class JMenuAppletDemo extends JApplet {
    public void init() {
        JMenuBar menuBar = new JMenuBar();
        JMenu menu1 = new JMenu("Menu1");

        ...
        JMenuItem item12 = new JMenuItem("Menu12");
        JMenu item113 = new JMenu("Menu113");
        item113.add(new JMenuItem("Menu1131"));

        ...
        menu11.add(new JMenuItem("Menu111"));
        menu11.add(new JMenuItem("Menu112"));
        menu11.add(item113);
        menu1.add(menu11);

        ...
        menu2.add(new JCheckBoxMenuItem("CheckBox1"));

        ...
        JMenuItem item1 = new JRadioButtonMenuItem("Radio1" , true);
        JMenuItem item2 = new JRadioButtonMenuItem("Radio2");
        ButtonGroup group = new ButtonGroup();

        ...
        menuBar.add(menu1);

        ...
        setJMenuBar(menuBar);
    }
}
```

4.b. Belka narzędziowa - przykład apletu

```
import java.awt.*;
import javax.swing.*;
public class JToolBarAppletDemo extends JApplet {
    public void init() {
        JToolBar tool = new JToolBar();
        tool.add(new JButton(new ImageIcon("new.gif")));
        tool.addSeparator();
        tool.add(new JButton(new ImageIcon("open.gif")));
        tool.add(new JButton(new ImageIcon("save.gif")));
        getContentPane().add(tool, BorderLayout.NORTH);
    }
}
```

4.c. Pasek statusu - konstruktor aplikacji

```
public StatusBarApplicationDemo() {
    JLabel textStatus = new JLabel("Text in the Status Bar !!!");
    JPanel statusBar = new JPanel(new FlowLayout(FlowLayout.LEFT));
    statusBar.setBackground(Color.LIGHT_GRAY);
    statusBar.add(textStatus);
    getContentPane().setBackground(Color.WHITE);
    getContentPane().add(statusBar, BorderLayout.PAGE_END);
}
```

4.d. Zadania - zbudować aplikację (aplet) posiadającą menu o strukturze jak na rysunku, belkę narzędziową z lewej strony i pasek statusu z tekstem

