

PROGRAMOWANIE W JĘZYKU JAVA

Prowadzący: dr hab. inż. **Jan Prokop**, prof. PRz, e-mail: jprokop@prz.edu.pl,
Politechnika Rzeszowska, Wydział Elektrotechniki i Informatyki

LABORATORIUM

ĆWICZENIE nr 2

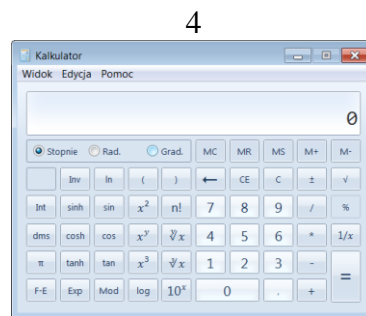
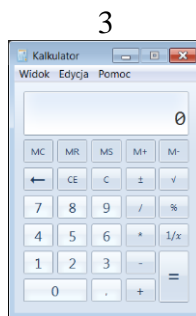
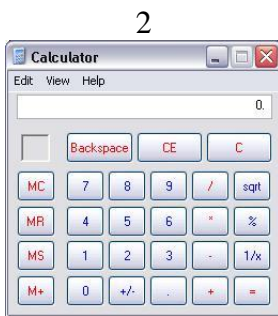
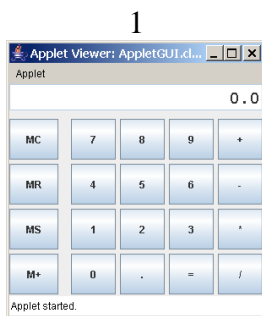
Temat: Java budowa GUI - Swing

1. Budowa GUI - kod przykładowej aplikacji biblioteki Swing

```
import java.awt.*;
import javax.swing.*;
public class AppGUI extends JFrame {
    public AppGUI() {
        setTitle("AppGUI");
        JPanel p1 = new JPanel();
        p1.setBackground(Color.RED);
        JTextField display = new JTextField("0.0");
        JButton b1 = new JButton("Button 1");
        JButton b2 = new JButton("Button 2");
        JButton b3 = new JButton("Button 3");
        JButton b4 = new JButton("Button 4");
        p1.add(display);
        p1.add(b1);
        p1.add(b2);
        p1.add(b3); p1.add(b4);
        add(p1);
    }
    public static void main(String[] args) {
        AppGUI frame = new AppGUI();
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(600, 500);
        frame.setVisible(true);
    }
}
```

1.1. Budowa GUI z zastosowaniem narzędzi podstawowych

Zrealizować GUI kalkulatora o wyglądzie jak na rysunku (GUI wybiera prowadzący dla każdej grupy laboratoryjnej) z zastosowaniem klas kontenerów, rozkładów, komponentów, podanych w punktach 2 - 4.



Uwaga ! W tym zadaniu wolno korzystać tylko z podstawowego edytora lub nakładki na pakiet JDK i dokumentacji Java SE 8 API: <http://docs.oracle.com/javase/8/docs/api/>

1.2. Budowa GUI w środowiskach programów NetBeans, Eclipse i innych

Zrealizować GUI kalkulatora o wyglądzie podanym przez prowadzącego w wybranym środowisku IDE.

2. Podstawowe rozkłady komponentów z biblioteki AWT

2.1. Rozkład ciągły - FlowLayout

```
FlowLayout layout = new FlowLayout();
layout.setHgap(10);
layout.setVgap(5);
FlowLayout layout = new FlowLayout(FlowLayout.RIGHT);
FlowLayout layout = new FlowLayout(FlowLayout.RIGHT, 5, 10);
setLayout(layout);
```

- **Przykład**

```
import java.awt.*;
import javax.swing.*;
public class FlowLayoutDemo extends JFrame {
    public FlowLayoutDemo() {
        JButton b1 = new JButton("1");
        JButton b2 = new JButton("2");
        JButton b3 = new JButton("3");
        JButton b4 = new JButton("4");
        JButton b5 = new JButton("5");
        JButton b6 = new JButton("6");
        add(b1);
        add(b2);
        add(b3);
        add(b4);
        add(b5);
        add(b6);
        setTitle("FlowLayoutDemo");
        setLayout(new FlowLayout(FlowLayout.RIGHT));
        setSize(300, 300);
        setVisible(true);
    }
    public static void main(String[] args) {
        FlowLayoutDemo flowLayoutDemo = new FlowLayoutDemo();
    }
}
```

2.2. Rozkład siatkowy - GridLayout

```
setLayout(new GridLayout(3,2));
add(new Button("Button 1"));
add(new Button("Button 2"));
setLayout(new GridLayout(3,2,5,5));
```

- **Przykład**

```
import java.awt.*;
import javax.swing.*;
public class GridLayoutDemo extends JFrame {
    public GridLayoutDemo() {
        JButton b1 = new JButton("1");
        JButton b2 = new JButton("2");
        JButton b3 = new JButton("3");
        JButton b4 = new JButton("4");
        JButton b5 = new JButton("5");
        JButton b6 = new JButton("6");
        add(b1);
        add(b2);
        add(b3);
        add(b4);
        add(b5);
        add(b6);
    }
}
```

```
        setTitle("GridLayoutDemo");
        setLayout(new GridLayout(2, 3));
        setSize(600, 600);
        setVisible(true);
    }
    public static void main(String[] args) {
        GridLayoutDemo gridLayoutDemo = new GridLayoutDemo();
    }
}
```

2.3. Rozkład brzegowy – BorderLayout

```
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);
```

- **Przykład**

```
import java.awt.*;
import javax.swing.*;
public class BorderLayoutDemo extends JFrame {
    public BorderLayoutDemo() {
        JButton b1 = new JButton("NORTH");
        JButton b2 = new JButton("SOUTH");
        JButton b3 = new JButton("EAST");
        JButton b4 = new JButton("WEST");
        JButton b5 = new JButton("CENTER");
        add(b1, BorderLayout.NORTH);
        add(b2, BorderLayout.SOUTH);
        add(b3, BorderLayout.EAST);
        add(b4, BorderLayout.WEST);
        add(b5, BorderLayout.CENTER);
        setTitle("BorderLayoutDemo");
        setSize(600, 600);
        setVisible(true);
    }
    public static void main(String[] args) {
        BorderLayoutDemo borderLayoutDemo = new BorderLayoutDemo();
    }
}
```

2.4. Rozkład pudełkowy – BoxLayout

```
import javax.swing.*;
import java.awt.*;
public class BoxLayoutTest extends JFrame {
    public BoxLayoutTest() {
        JButton button1 = new JButton("1");
        JButton button2 = new JButton("Long-Named Button 2");
        button2.setAlignmentX(Component.CENTER_ALIGNMENT);
        JButton button3 = new JButton("Button 3");
        button3.setFont(new Font("Arial", Font.PLAIN, 20));
        JPanel p = new JPanel();
        p.setLayout(new BoxLayout(p, BoxLayout.Y_AXIS));
        // X_AXIS, Y_AXIS, LINE_AXIS, PAGE_AXIS
        p.add(button1);
        p.add(button2);
```

```
p.add(button3);
getContentPane().add(p, BorderLayout.CENTER);
}
public static void main(String[] args) {
    BoxLayoutTest frame = new BoxLayoutTest();
    frame.setBounds(10, 10, 300, 200);
    frame.setTitle("BoxLayoutTest");
    frame.setVisible(true);
}
}
```

2.5. Rozkład GridBagLayout

```
import java.awt.*;
import javax.swing.*;
public class GridBagLayoutDemo extends JFrame {
    public 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);
    }
}
```

```
}
}
```

2.6. Rozkład kart – CardLayout

```
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);
```

2.7. Wyłączenie domyślnego rozkładu - brak rozkładu - null Layout

```
setLayout(null);
setLocation(10,160);
setSize(50,50);
setBounds(180,160,50,50);
setPreferredSize(new Dimension(100,80));
setHorizontalAlignment(SwingConstants.RIGHT);
```

2.8. Własny rozkład komponentów

Zadanie - Zaprojektować własny układ rozkładu z komponentami na okręgu (jak na rysunku), zastosować interfejs `LayoutManager`

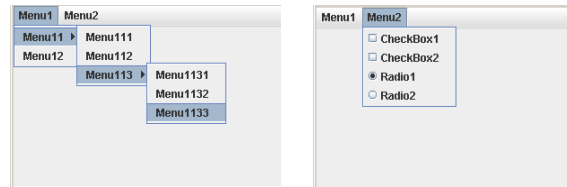


3. Budowa menu, belki narzędziowej i paska statusu

3.1. Menu

```
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);
```

- **Zadanie** - zbudować aplikację posiadającą menu o strukturze jak na rysunku



3.2. Belka narzędziowa

```
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);
```

• Przykład

```
import java.awt.BorderLayout;
import javax.swing.JButton;
import javax.swing.JComboBox;
import javax.swing.JFrame;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;
import javax.swing.JToolBar;

public class JToolBarDemo extends JFrame {
    public JToolBarDemo() {
        JToolBar toolbar = new JToolBar();
        toolbar.setRollover(true);
        JButton button = new JButton("File");
        toolbar.add(button);
        toolbar.addSeparator();
        toolbar.add(new JButton("Edit"));
        toolbar.addSeparator();
        toolbar.add(new JComboBox(new String[]{"Item 1", "Item 2", "Item 3"}));
        toolbar.addSeparator();
        add(toolbar, BorderLayout.NORTH);
        JTextArea textArea = new JTextArea("Text Area");
        JScrollPane mypane = new JScrollPane(textArea);
        add(mypane, BorderLayout.CENTER);
        setSize(450, 250);
        setVisible(true);
    }
    public static void main(String[] args) {
        JToolBarDemo jToolBarDemo = new JToolBarDemo();
    }
}
```

- **Zadanie** - zbudować aplikację posiadającą belkę narzędziową z lewej strony jak na rysunku



3.3. Pasek statusu

```
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. Wybrane kontenery i komponenty

```
JFrame.setDefaultLookAndFeelDecorated(true);
JFrame frame = new JFrame("JFrame");

JPanel p = new JPanel();
p.setBorder(new EmptyBorder(new Insets(20, 20, 50, 50)));
p.setBackground(Color.yellow);
p.add(Box.createRigidArea(new Dimension(10,0)));
p.add(new JSeparator(SwingConstants.VERTICAL));

JButton b = new JButton("Button");
b.setForeground();
b.setBackground();

Dimension dim = new Dimension(100,80);
b.setPreferredSize(dim);
b.setHorizontalAlignment();
b.setVerticalAlignment();
b.setEnabled();

JLabel label = new JLabel("Text");
ImageIcon icon = createImageIcon("icon.gif");
JLabel label = new JLabel(icon);
JLabel label = new JLabel("Text", icon, JLabel.CENTER);
label.setVerticalTextPosition(JLabel.BOTTOM);
label.setHorizontalTextPosition(JLabel.CENTER);

JTextField tf = new JTextField(100);
tf.setEchoChar('*');
add(tf);
JPasswordField password = new JPasswordField(6);

JTextArea ta = new JTextArea(20, 200);
JScrollPane sp = new JScrollPane(ta);
ta.setEditable(false);

JCheckBox cb = new JCheckBox("Green");
String colors[] = {"Red", "Blue", "Green", "Yellow"};
JComboBox cb = new JComboBox(colors);
JList jl = new JList(colors);
cb.setSelected(true);

JRadioButton rb1 = new JRadioButton("A");
JRadioButton rb2 = new JRadioButton("B");
rb1.setSelected(true);
ButtonGroup bg = new ButtonGroup();
bg.add(rb1);
bg.add(rb2);
```

```
Scrollbar sb = new Scrollbar(Scrollbar.HORIZONTAL, 1, 10, 1, 100);  
Scrollbar sb = new Scrollbar(Scrollbar.VERTICAL, 1, 10, 1, 200);
```

- **Przykład, komponent JComboBox**

```
import javax.swing.*;  
public class ComboBoxExample extends JFrame {  
  
    public ComboBoxExample() {  
  
        String item[]={"Item 1","Item 2","Item 3","Item 4","Item 5"};  
        JComboBox cb=new JComboBox(item);  
        cb.setBounds(50, 50,90,20);  
        add(cb);  
        setLayout(null);  
        setSize(400,500);  
        setVisible(true);  
    }  
    public static void main(String[] args) {  
        new ComboBoxExample();  
    }  
}
```

5. Zrealizować interfejs użytkownika w środowisku IDE

NetBeans IDE - <http://www.netbeans.org>

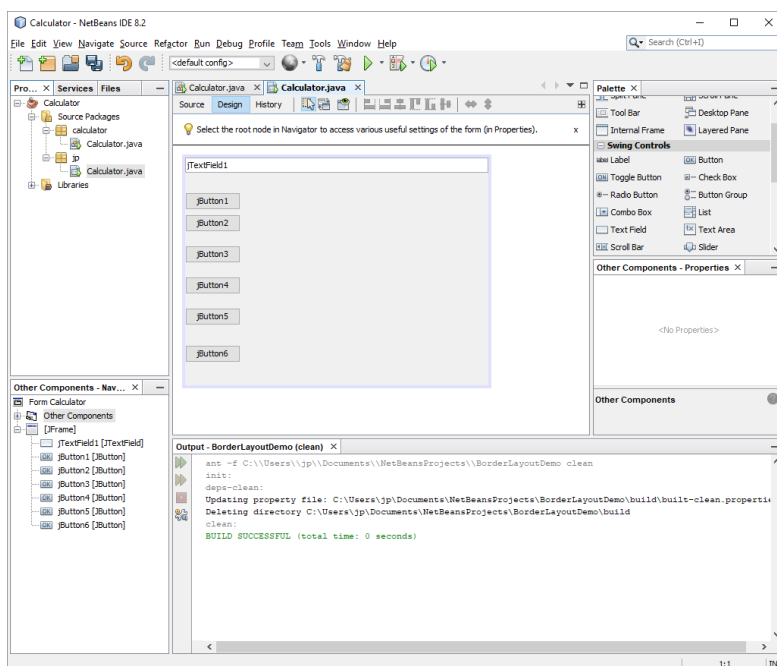
Eclipse - <http://www.eclipse.org/downloads/>

IntelliJ IDEA - <http://www.jetbrains.com/idea/>

JDeveloper - <http://www.oracle.com/technology/products/jdev>

Android Studio - <http://developer.android.com/sdk/index.html>

NetBeans



6. Zadania

Podaje prowadzący w każdej grupie laboratoryjnej