

# Interfejs graficzny w Javie.

Robert A. Kłopotek  
r.klopotek@uksw.edu.pl

Wydział Matematyczno-Przyrodniczy. Szkoła Nauk Ścisłych, UKSW

20.04.2017

## Interfejs graficzny w Javie

---

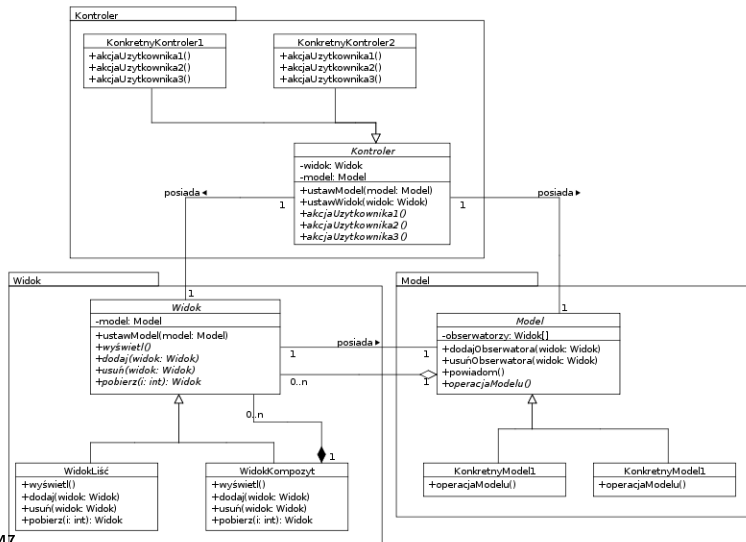
- Java dostarcza dwóch podstawowych środowisk graficznych do budowania aplikacji okienkowych: AWT i Swing
- AWT (Abstract Windowing Toolkit) - komponenty są "ciężkie" zależne od platformy i używają zasobów wyświetlanych przez system operacyjny. Komponenty znajdują się w pakiecie `java.awt`
- Swing - komponenty są "lekkie", niezależne od platformy. Komponenty znajdują się w pakiecie `javax.swing` i wspierają MVC

# Wzorzec MVC

---

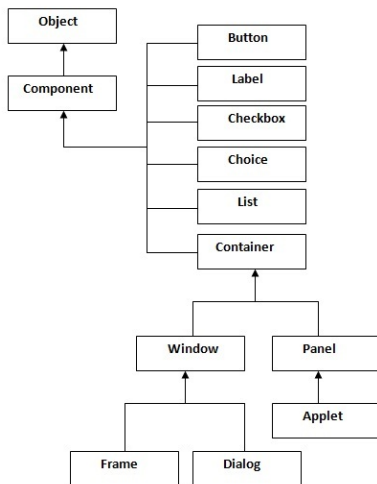
- Głównie służy do organizowania struktury aplikacji posiadających graficzne interfejsy użytkownika
- MVC (Model-View-Controller) - wzorzec projektowy posiadający 3 składowe:
  - Model – jest pewną reprezentacją problemu bądź logiki aplikacji.
  - Widok – opisuje, jak wyświetlić pewną część modelu w ramach interfejsu użytkownika. Może składać się z podwidoków odpowiedzialnych za mniejsze części interfejsu.
  - Kontroler – przyjmuje dane wejściowe od użytkownika i reaguje na jego poczynania, zarządzając aktualizacje modelu oraz odświeżenie widoków.

# MVC - schemat



# Hierarchia AWT

---



## Główne kontenery

---

- klasa `Container` - jest to klasa kontenerowa służąca do przetrzymywania innych komponentów takich jak przyciski, pola tekstowe, etykiety itp.
- klasa `Window` - kontener, który nie ma ramki i pasków menu. Aby utworzyć okno, trzeba użyć ramki (`Frame`), okna dialogowego (`Dialog`) lub innego okna.
- klasa `Panel` - kontener, który nie zawiera paska tytułu i pasków menu. Może zawierać inne komponenty, takie jak przycisk, pole tekstowe itd.
- Klasa `Frame` - kontener zawierający pasek tytułu i może mieć paski menu. Może zawierać inne składniki, takie jak przycisk, pole tekstowe itd.

## Główne metody klasy Component

---

- `public void add(Component c)`
- `public void setSize(int width,int height)`
- `public void setLayout(LayoutManager m)`
- `public void setVisible(boolean status)`
- `public void setBounds(int x, int y, int width, int height)`

# Główne komponenty AWT

---

- Button
- Label
- TextField, TextArea
- Checkbox, CheckboxGroup
- Choice
- List(int rows, boolean multipleMode)
- Canvas, Panel
- MenuItem, Menu, MenuBar [frame.setMenuBar(mb)], PopupMenu
- Scrollbar
- Dialog
- Toolkit



# Komponenty AWT

---



## Frame i Button - przykład

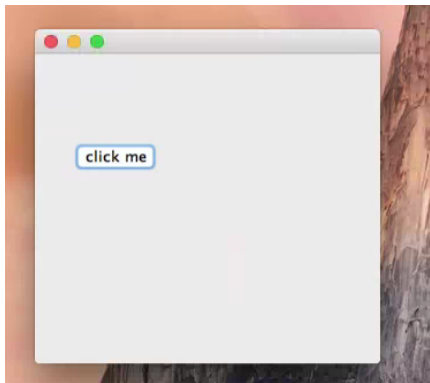
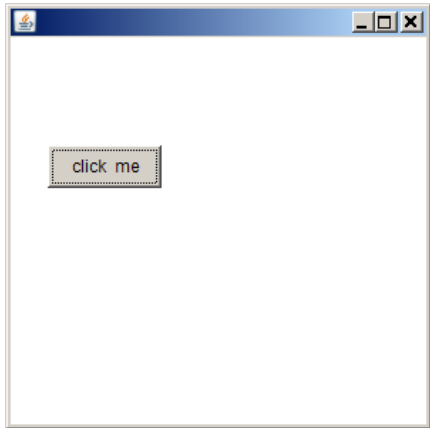
---

```
import java.awt.*;
class First extends Frame{
    First(){
        Button b=new Button("click me");
        b.setBounds(30,100,80,30);
        add(b);
        setSize(300,300);
        setLayout(null);
        setVisible(true);
    }
    public static void main(
        String args[]){
        First f=new First();
    }
}
```

```
import java.awt.*;
class First2{
    First2(){
        Frame f=new Frame();
        Button b=new Button("click me");
        b.setBounds(30,50,80,30);
        f.add(b);
        f.setSize(300,300);
        f.setLayout(null);
        f.setVisible(true);
    }
    public static void main(
        String args[]){
        First2 f=new First2();
    }
}
```

## Frame i Button - Windows vs Mac

---



# Obsługa zdarzeń

---

- Zdarzenie to zmiana stanu obiektu
- Zdarzenia to np. naciśnięcie przycisku, kliknięcie przycisku, przeciąganie myszą itp.
- Pakiet `java.awt.event` dostarcza wielu klas zdarzeń (event classes) i wielu klas nasłuchiaczy zdarzeń (Listener interfaces)
- Aby obsłużyć zdarzenia należy zarejestrować komponent do nasłuchiacza zdarzeń

## Klasy zdarzeń i interfejsy nasłuchiwczy

---

Event Classes	Listener Interfaces
ActionEvent	ActionListener
MouseEvent	MouseListener/MouseMotionListener
MouseWheelEvent	MouseWheelListener
KeyEvent	KeyListener
ItemEvent	ItemListener
TextEvent	TextListener
AdjustmentEvent	AdjustmentListener
WindowEvent	WindowListener
ComponentEvent	ComponentListener
ContainerEvent	ContainerListener
FocusEvent	FocusListener

## ActionListener - przykład

---

```
import java.awt.*;
import java.awt.event.*;
class AEvent extends Frame implements ActionListener{
    TextField tf;
    AEvent(){
        tf=new TextField();
        tf.setBounds(60,50,170,20);
        Button b=new Button("click me");
        b.setBounds(100,120,80,30);
        b.addActionListener(this);
        add(b);add(tf);
        setSize(300,300);
        setLayout(null);
        setVisible(true);
    }
    public void actionPerformed(ActionEvent e){
        tf.setText("Welcome");
    }
    public static void main(String args[]){
        new AEvent();
    }
}
14 / 47
```

## WindowListener - przykład

---

```
import java.awt.*;
import java.awt.event.*;
public class First3{
    First3(){
        Frame f=new Frame();
        Button b=new Button("click me");
        b.setBounds(30,50,80,30);
        f.add(b);
        f.setSize(300,300);
        f.setLayout(null);
        f.setVisible(true);
        f.addWindowListener(new WindowAdapter(){
            public void windowClosing(WindowEvent we){
                //System.exit(0);
                dispose();
            }
        });
    }
    public static void main(String args[]){
        First3 f=new First3();
    }
}15 / 47
```

## Nasłuchiwanie w AWT (1/2)

---

- ActionListener - jest powiadamiany, gdy naciśniemy guzik lub opcje menu
  - `public abstract void actionPerformed(ActionEvent e);`
- MouseListener - jest powiadamiany, gdy wykonamy akcję myszką
  - `public abstract void mouseClicked(MouseEvent e);`
  - `public abstract void mouseEntered(MouseEvent e);`
  - `public abstract void mouseExited(MouseEvent e);`
  - `public abstract void mousePressed(MouseEvent e);`
  - `public abstract void mouseReleased(MouseEvent e);`
- MouseMotionListener - jest powiadamiany, gdy ruszamy lub przeciągamy coś myszką
  - `public abstract void mouseDragged(MouseEvent e);`
  - `public abstract void mouseMoved(MouseEvent e);`
- ItemListener - jest powiadamiany, gdy klikniemy w checkbox
  - `public abstract void itemStateChanged(ItemEvent e);`



## Nasłuchiwanie w AWT (2/2)

---

- **KeyListener** - jest powiadamiany, gdy zmienimy stan klawiszy
  - `public abstract void keyPressed(KeyEvent e);`
  - `public abstract void keyReleased(KeyEvent e);`
  - `public abstract void keyTyped(KeyEvent e);`
- **WindowListener** - jest powiadamiany, gdy zmienimy stan okna
  - `public abstract void windowActivated(WindowEvent e);`
  - `public abstract void windowClosed(WindowEvent e);`
  - `public abstract void windowClosing(WindowEvent e);`
  - `public abstract void windowDeactivated(WindowEvent e);`
  - `public abstract void windowDeiconified(WindowEvent e);`
  - `public abstract void windowIconified(WindowEvent e);`
  - `public abstract void windowOpened(WindowEvent e);`

## WindowListener i Toolkit - przykład

---

```
import java.awt.*;
import java.awt.event.*;
public class ToolkitExample {
    public static void main(String[] args) {
        Frame f=new Frame("ToolkitExample");
        Button b=new Button("beep");
        b.setBounds(50,100,60,30);
        f.add(b);
        f.setSize(300,300);
        f.setLayout(null);
        f.setVisible(true);
        b.addActionListener(new ActionListener(){
            public void actionPerformed(ActionEvent e){
                Toolkit.getDefaultToolkit().beep();
            }
        });
        Toolkit t = Toolkit.getDefaultToolkit();
        System.out.println("Screen resolution = "
            + t.getScreenResolution());
        Dimension d = t.getScreenSize();
        System.out.println("Screen width = " + d.width);
        System.out.println("Screen height = " + d.height);
    }
}
```

## Klasy Adapterów AWT

---

Adapter class	Listener interface
WindowAdapter	WindowListener
KeyAdapter	KeyListener
MouseAdapter	MouseListener
MouseMotionAdapter	MouseMotionListener
FocusAdapter	FocusListener
ComponentAdapter	ComponentListener
ContainerAdapter	ContainerListener
HierarchyBoundsAdapter	HierarchyBoundsListener

## MouseMotionAdapter - przykład

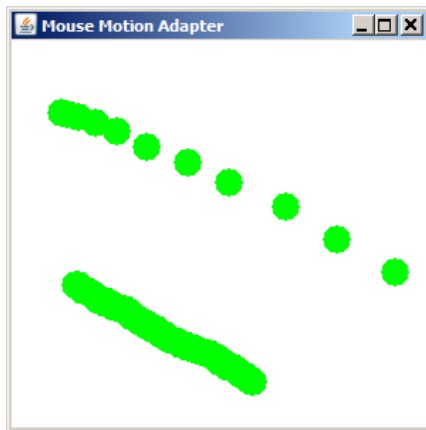
---

```
import java.awt.*;
import java.awt.event.*;
public class MouseMotionAdapterExample
        extends MouseMotionAdapter{
    Frame f;
    MouseMotionAdapterExample(){
        f=new Frame("Mouse_Motion_Adapter");
        f.addMouseMotionListener(this);

        f.setSize(300,300);
        f.setLayout(null);
        f.setVisible(true);
    }
    public void mouseDragged(MouseEvent e) {
        Graphics g=f.getGraphics();
        g.setColor(Color.GREEN);
        g.fillOval(e.getX(),e.getY(),20,20);
    }
    public static void main(String[] args) {
        new MouseMotionAdapterExample();
    }
}
20 / 47
```

# MouseMotionAdapter Windows

---



## Wyświetlanie grafik - klasa Graphics

---

- `public abstract void drawString(String str, int x, int y)`
- `public void drawRect(int x, int y, int width, int height)`
- `public abstract void fillRect(int x, int y, int width, int height)`
- `public abstract void drawOval(int x, int y, int width, int height)`
- `public abstract void fillOval(int x, int y, int width, int height)`
- `public abstract void drawLine(int x1, int y1, int x2, int y2)`
- `public abstract boolean drawImage(Image img, int x, int y, ImageObserver observer)`
- `public abstract void drawArc(int x, int y, int width, int height, int startAngle, int arcAngle)`
- `public abstract void fillArc(int x, int y, int width, int height, int startAngle, int arcAngle)`
- `public abstract void setColor(Color c)`
- `public abstract void setFont(Font font)`

## Wyświetlanie obrazków - drawImage() przykład

---

```
import java.awt.*;

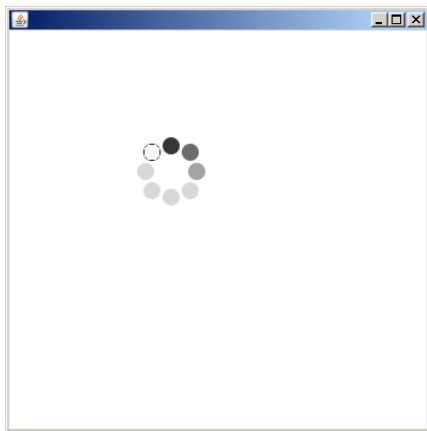
public class MyCanvas extends Canvas{

    public void paint(Graphics g) {

        Toolkit t=Toolkit.getDefaultToolkit();
        Image i=t.getImage("loading.gif");
        g.drawImage(i, 120,100,this);
    }
    public static void main(String[] args) {
        MyCanvas m=new MyCanvas();
        Frame f=new Frame();
        f.add(m);
        f.setSize(400,400);
        f.setVisible(true);
    }
}
```

## Wyświetlanie obrazków - drawImage() Windows

---



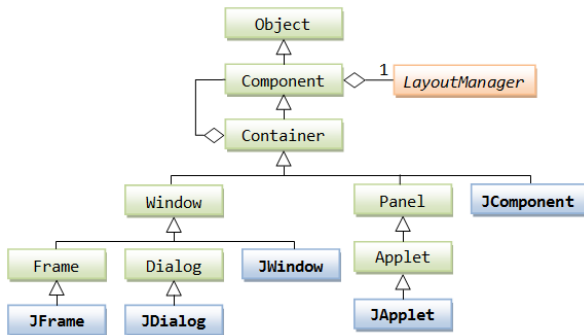


## Interfejs graficzny w Javie

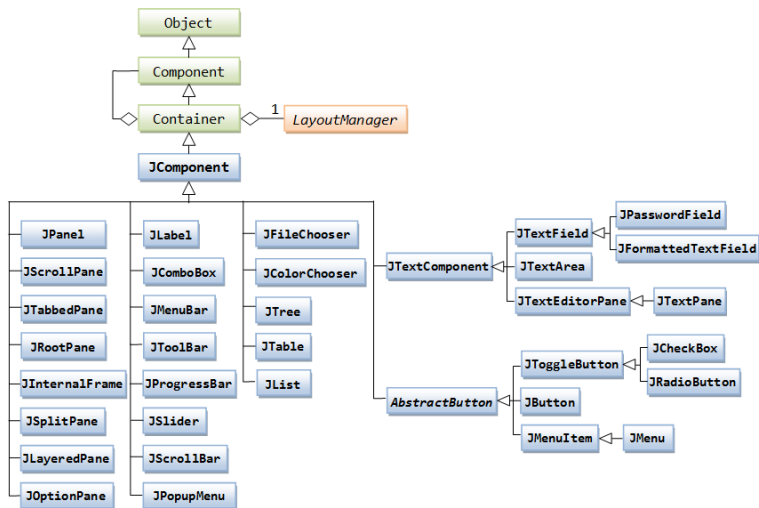
---

- Swing - komponenty są "lekkie", niezależne od platformy. Komponenty znajdują się w pakiecie `javax.swing` i wspierają MVC.
- Swing jest częścią Java Foundation Classes (JFC), którego zadaniem jest uproszczenie pisania interfejsu graficznego
- Swing w całości jest napisana w Javie, a rysowanie jej komponentów odbywa się piksel po pikselu
- Przykładowe klasy Swing: `JButton`, `JTextField`, `JTextArea`, `JRadioButton`, `JCheckBox`, `JMenu`, `JColorChooser`.

# Hierarchia kontenerów Swing



# Hierarchia komponentów Swing



# Główne komponenty Swing

---

- JButton, JLabel
- JTextField, JTextArea, JPasswordField
- JCheckBox, JRadioButton, JComboBox
- JTable, JList
- JOptionPane - okienka dialogowe
- Canvas, Panel
- JMenuItem, JMenu, JMenuBar, JPopupMenu
- JScrollbar, JProgressBar
- JTree, JTabbedPane, JSlider, JSpinner
- JColorChooser, JFileChooser

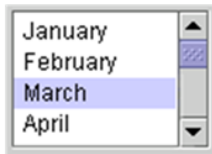
# Komponenty Swing



Buttons



Combo Box



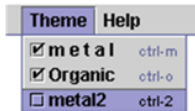
List



TextField



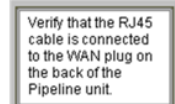
Slider



Menu



Label



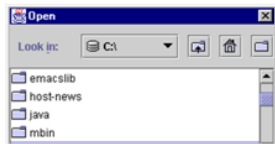
Text Area



Tool Tip



Progress Bar



File Chooser



Color Chooser

## JFrame i JButton - przykład

---

```
import javax.swing.*;
public class FirstSwingExample {
    public static void main(String[] args) {
        JFrame f=new JFrame();

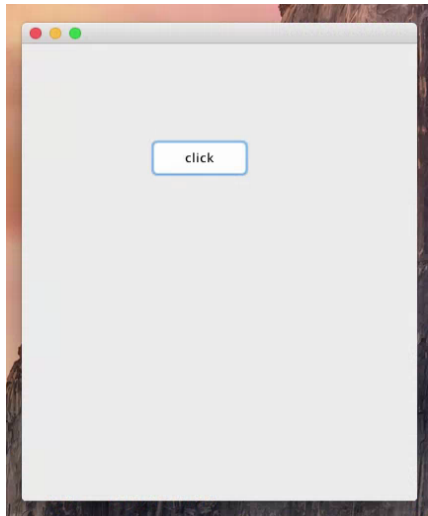
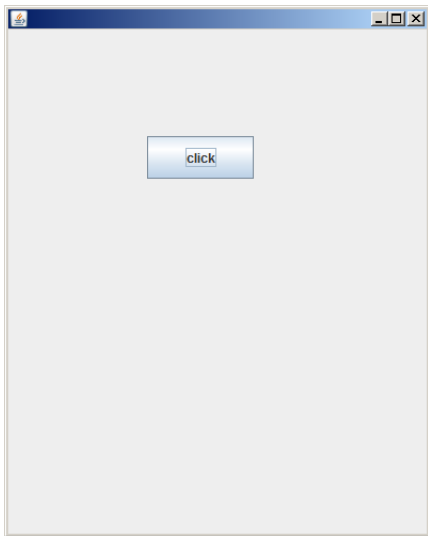
        JButton b=new JButton("click");
        b.setBounds(130,100,100, 40);

        f.add(b);

        f.setSize(400,500);
        f.setLayout(null);
        f.setVisible(true);
    }
}
```

## JFrame i JButton - Windows vs Mac

---



## Ikona w JButton - przykład

---

```
import javax.swing.*;
public class JButtonExample{
    JButtonExample(){
        JFrame f=new JFrame("JButton□Example");
        JButton b=new JButton(new ImageIcon("button_icon.png"));
        b.setBounds(50,50,286, 90);
        f.add(b);
        f.setSize(400,300);
        f.setLayout(null);
        f.setVisible(true);
        f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }

    public static void main(String[] args) {
        new JButtonExample();
    }
}
```



## Ikona w JButton - Windows

---



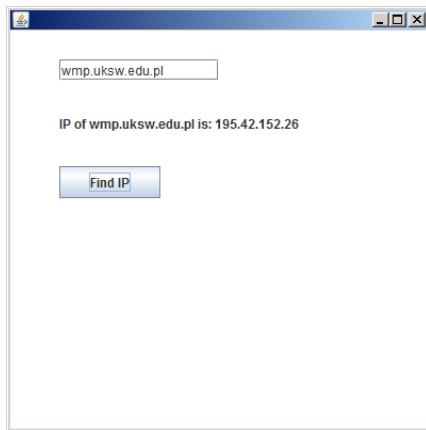
## JLabel, JTextField i wyświetlanie IP - przykład

---

```
import javax.swing.*;
import java.awt.event.*;
public class LabelExample extends JFrame implements ActionListener{
    JTextField tf; JLabel l; JButton b;
    LabelExample(){
        tf=new JTextField();
        tf.setBounds(50,50, 150,20);
        l=new JLabel(); l.setBounds(50,100, 250,20);
        b=new JButton("Find IP");
        b.setBounds(50,150,95,30); b.addActionListener(this);
        add(b);add(tf);add(l);
        setSize(400,400); setLayout(null); setVisible(true);
    }
    public void actionPerformed(ActionEvent e) {
        try{ String host=tf.getText();
            String ip=java.net.InetAddress.getByName(host).getHostAddress();
            l.setText("IP of "+host+" is: "+ip);
        }catch(Exception ex){System.out.println(ex);}
    }
    public static void main(String[] args) {
        new LabelExample();
    }
}
```

# Ikona w JButton - Windows

---



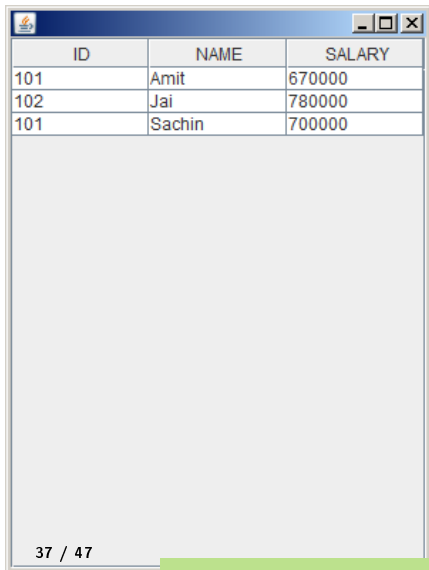
## JTable i JScrollPane - przykład

---

```
import javax.swing.*;
public class TableExample {
    JFrame f;
    TableExample(){
        f=new JFrame();
        String data[][]={ {"101","Amit","670000"},
                           {"102","Jai","780000"},
                           {"101","Sachin","700000"}};
        String column[]={ "ID","NAME","SALARY"};
        JTable jt=new JTable(data,column);
        jt.setBounds(30,40,200,300);
        JScrollPane sp=new JScrollPane(jt);
        f.add(sp);
        f.setSize(300,400);
        f.setVisible(true);
    }

    public static void main(String[] args) {
        new TableExample();
    }
}
```

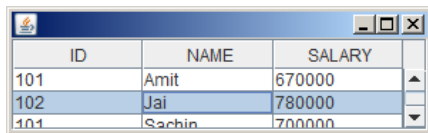
## JTable i JScrollPane - Windows



A screenshot of a Java Swing window titled "JTable i JScrollPane - Windows". The window contains a JTable with three columns: "ID", "NAME", and "SALARY". The table has three rows of data. The first row has ID 101, NAME Amit, and SALARY 670000. The second row has ID 102, NAME Jai, and SALARY 780000. The third row has ID 101, NAME Sachin, and SALARY 700000. The table is displayed in a standard Windows-style window with a title bar and control buttons.

ID	NAME	SALARY
101	Amit	670000
102	Jai	780000
101	Sachin	700000

37 / 47



A screenshot of a Java Swing window titled "JTable i JScrollPane - Windows". The window contains a JTable with three columns: "ID", "NAME", and "SALARY". The table has three rows of data. The first row has ID 101, NAME Amit, and SALARY 670000. The second row has ID 102, NAME Jai, and SALARY 780000. The third row has ID 101, NAME Sachin, and SALARY 700000. The second row is highlighted in blue. The table is displayed in a standard Windows-style window with a title bar and control buttons.

ID	NAME	SALARY
101	Amit	670000
102	Jai	780000
101	Sachin	700000

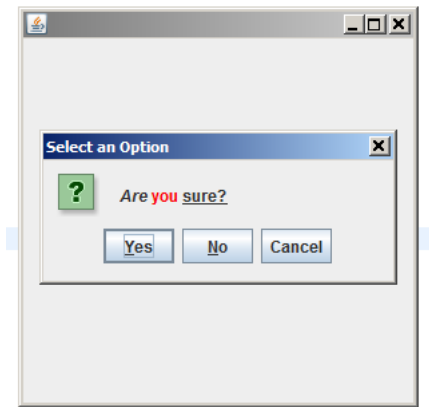
## JOptionPane - przykład

---

```
import javax.swing.*;
import java.awt.event.*;
public class OptionPaneExample extends WindowAdapter{
    JFrame f;
    OptionPaneExample(){
        f=new JFrame();
        f.addWindowListener(this);
        f.setSize(300, 300); f.setLayout(null); f.setVisible(true);
        f.setDefaultCloseOperation(JFrame.DO_NOTHING_ON_CLOSE);
    }
    public void windowClosing(WindowEvent e) {
        int a=JOptionPane.showConfirmDialog(f,"<html>␣<i>Are</i>␣" +
            "<font␣color=#ff0000>you</font>␣" +
            "<u>sure?</u>");
        if(a==JOptionPane.YES_OPTION){
            f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        }
    }
    public static void main(String[] args) {
        new OptionPaneExample();
    }
} 38 / 47
```

## Ikona w JButton - Windows

---



## JPasswordField i Tooltip - przykład

---

```
import javax.swing.*;
public class TooltipExample {
    public static void main(String[] args) {
        JFrame f=new JFrame("PasswordFieldExample");

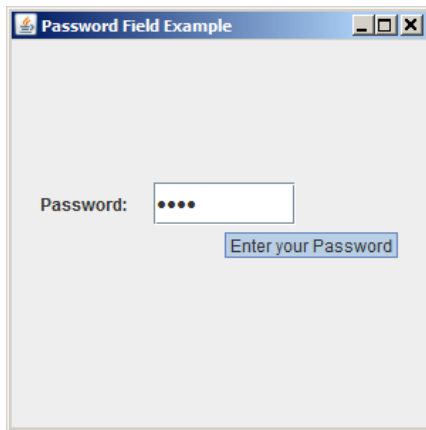
        JPasswordField value = new JPasswordField();
        value.setBounds(100,100,100,30);
        value.setToolTipText("Enter your Password");
        JLabel l1=new JLabel("Password:");
        l1.setBounds(20,100, 80,30);

        f.add(value);  f.add(l1);
        f.setSize(300,300);
        f.setLayout(null);
        f.setVisible(true);
    }
}
```



## JPasswordField i ToolTip - Windows

---

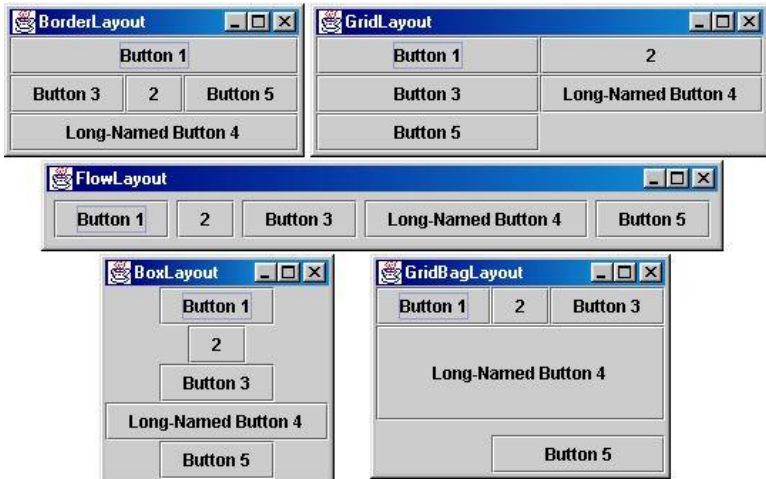


# Swing - LayoutManagers

---

- `LayoutManager` - jest używany do ułożenia komponentów w pewien sposób
- `LayoutManager` jest interfejsem implementowanym przez kilka klas:
  - `java.awt.BorderLayout`
  - `java.awt.FlowLayout`
  - `java.awt.GridLayout`
  - `java.awt.CardLayout`
  - `java.awt.GridBagLayout`
  - `javax.swing.BoxLayout`
  - `javax.swing.GroupLayout`
  - `javax.swing.ScrollPaneLayout`
  - `javax.swing.SpringLayout`

# LayoutManagers - Windows



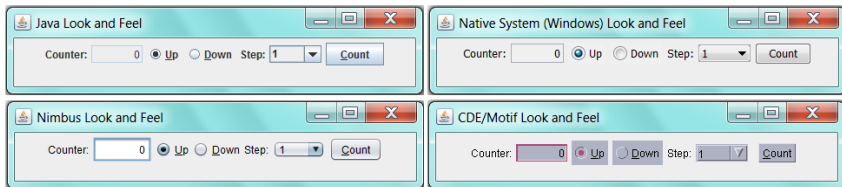
## Swing - LookAndFeel

---

- Swing obsługuje tak zwane "pluggable look and feel (plaf)" dla JComponents.
- "Look" odnosi się do wyglądu widżetów (JComponent);
- "feel" odnosi się do zachowania widżetów (na przykład zachowań kliknięć myszą na różne przyciski myszy).
- "Pluggable" odnosi się do możliwości zmiany wyglądu i uczucia w czasie wykonywania.
- Można wybrać domyślny "look and feel" Java lub natywne systemu (Windows, Linux, Mac) lub nowszy Nimbus "look and feel" na wielu platformach

# LookAndFeel - Windows

---



# LayoutManager i LookAndFeel - przykład

---

```
import java.awt.FlowLayout;
import java.awt.event.*;
import javax.swing.*;
public class LookAndFeelTest extends JFrame implements ActionListener{
    final UIManager.LookAndFeelInfo[] infos
        =UIManager.getInstalledLookAndFeels();

    LookAndFeelTest(){
        for(UIManager.LookAndFeelInfo info : infos){
            JButton b = new JButton(info.getName());
            b.setActionCommand(info.getClassName());
            b.addActionListener(this);
            add(b);
        }
        setTitle("Change Look and Feel");
        setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
        setLayout(new FlowLayout(FlowLayout.LEADING));
        setSize(300,300); setVisible(true);
    }
    public void actionPerformed(ActionEvent e) {
        JButton b = (JButton) e.getSource();
        try {
            UIManager.setLookAndFeel(b.getActionCommand());
            SwingUtilities.updateComponentTreeUI(this);
        } catch (Exception ex) {ex.printStackTrace();}
    }
    public static void main(String[] args) {
        new LookAndFeelTest();
    }
}46 / 47
```

Pytania?