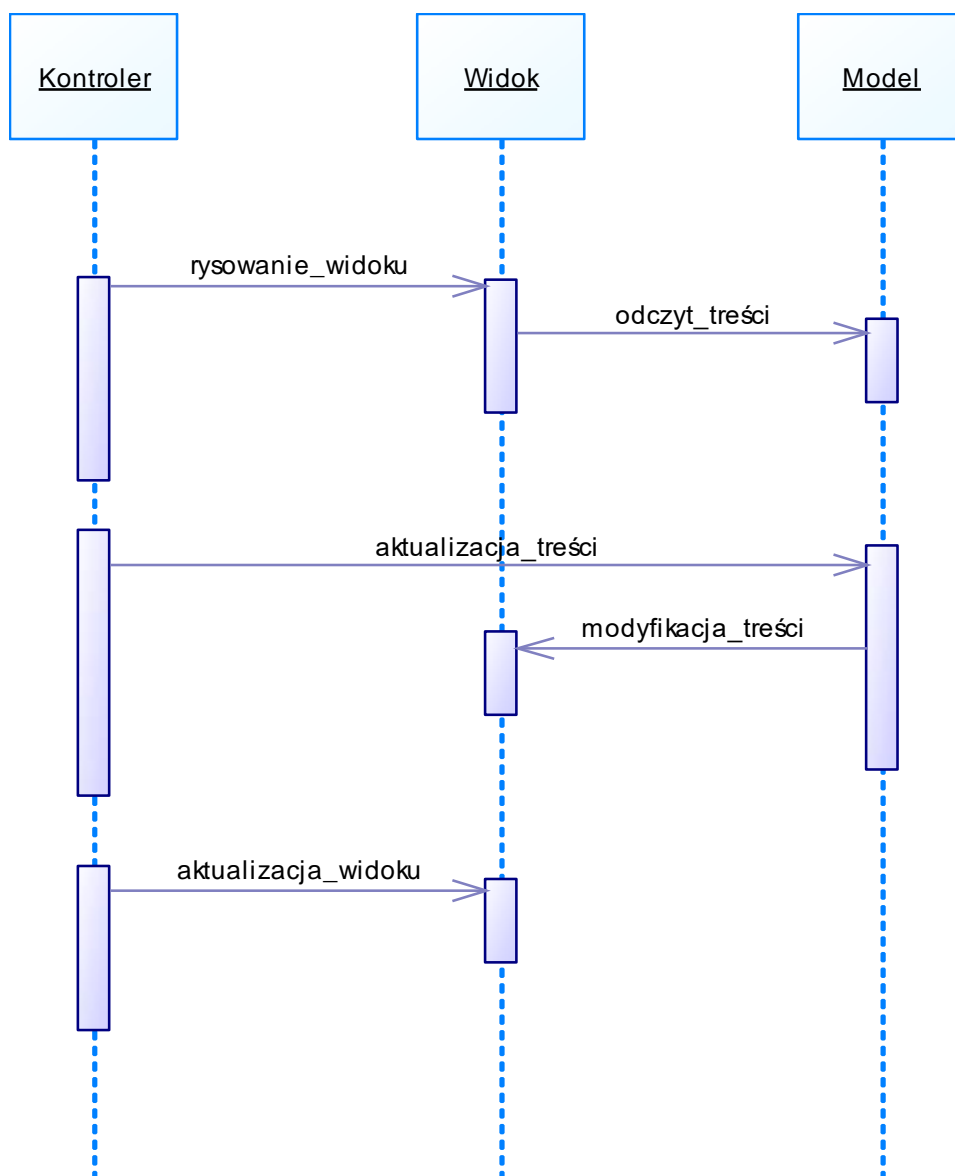


Swing a wzorzec MVC

MVC – Model-View-Controller

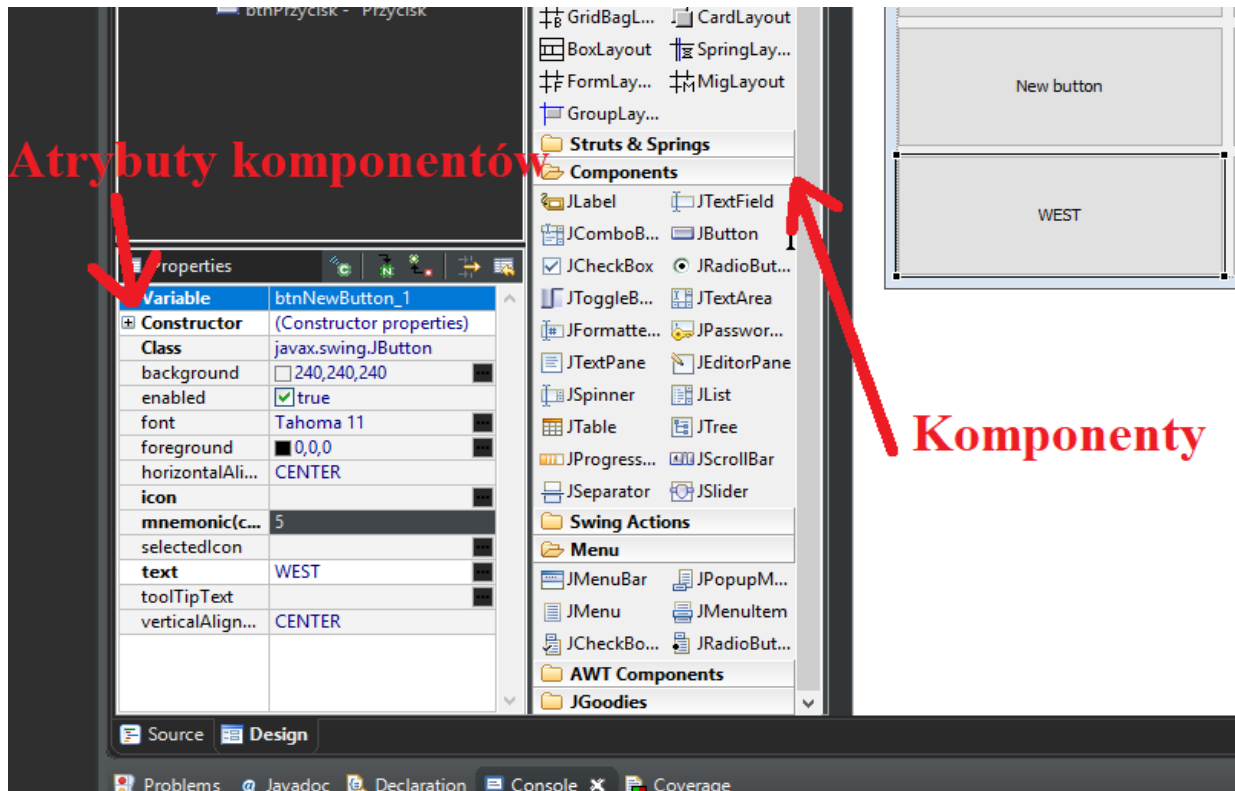
Wzorzec ten zakłada podział klasy na trzy składniki:

- Model – Treść – informacje zawarte przez obiekt
- Widok – Wygląd – co jest pokazane
- Kontroler – Zachowanie – reakcje na zdarzenia



Podstawowe klasy biblioteki SWING

- Przyciski - `JButton` `buton = new JButton();`
- Pole tekstowe – `TextField` `textField = new JTextField();`
- Etykieta - `JLabel` `label = new JLabel();` (tekst i ikona)
- Pole hasła – `JPasswordField` `password = new JPasswordField();`
- Obszar tekstowy – `TextArea` `textArea = new JTextArea();`
- Panel przewijania - `JScrollPane` `scroll = new JScrollPane(textArea);`
- CheckBox - `JCheckBox` `check = new JCheckBox();`
- Listy rozwijane - `JComboBox` `combo = new JComboBox();`
- Suwak – `JSlider` `slider = new JSlider(min, max, initialValue);`
- Menu `JMenuBar` `menu = new JMenuBar();`
 - `frame.setJMenuBar(menu);`
 - `JMenu` `editMenu = new JMenu();`
 - `Menu.add(editMenu);`
- Menu podręczne `JPopupMenu` `popup = new JPopupMenu();`
- Pasek narzędzi – `JToolBar` `toolBar = new JToolBar();`

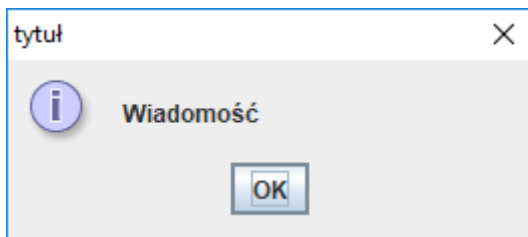


Okna dialogowe

Obiekt pane jest z klasy JPanel

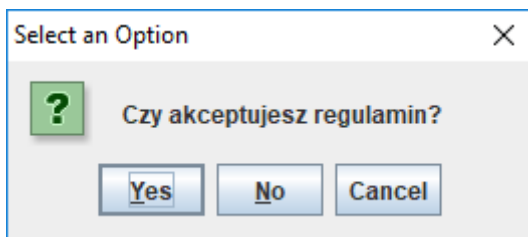
```
JOptionPane.showMessageDialog(pane, "Wiadomość",  
"tytuł", 1);
```

Ostatni parametr określa typ komunikatu

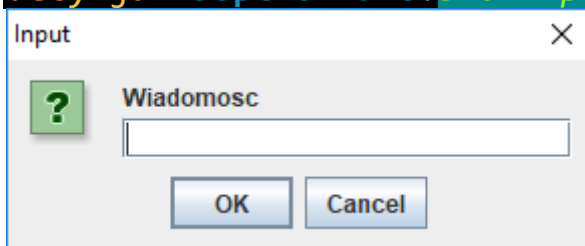


```
int decyzja;  
decyzja = JOptionPane.showConfirmDialog(pane, "Czy  
akceptujesz regulamin?");
```

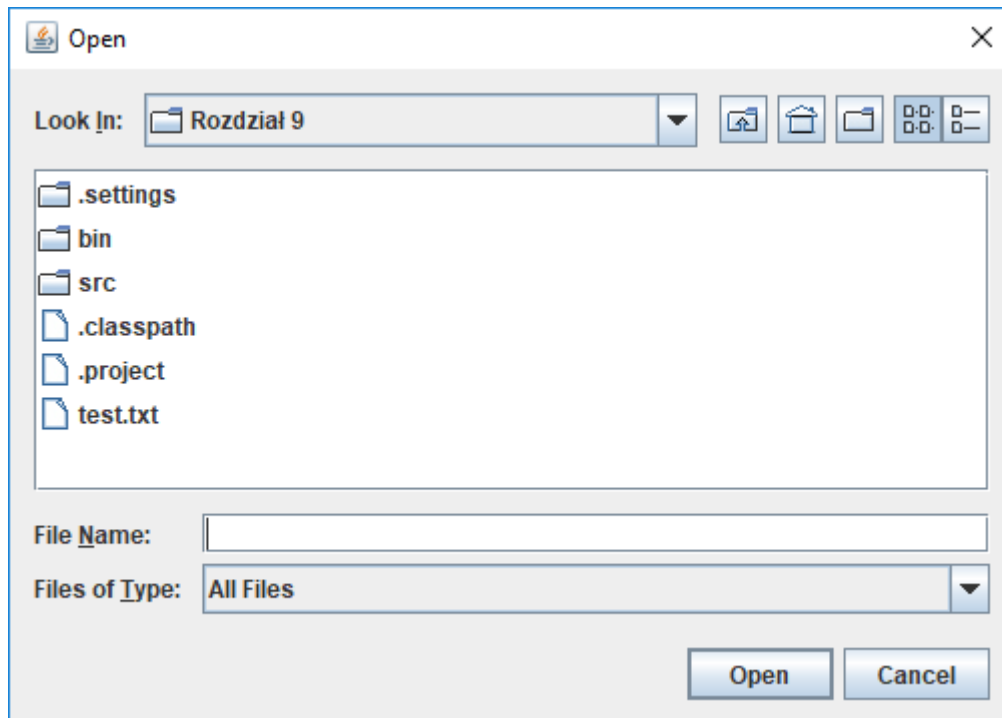
W zależności od wciśniętego przycisku zmienna decyzja przyjmie różną wartość (0,1,2)



```
String decyzja;  
decyzja = JOptionPane.showInputDialog("Wiadomosc");
```

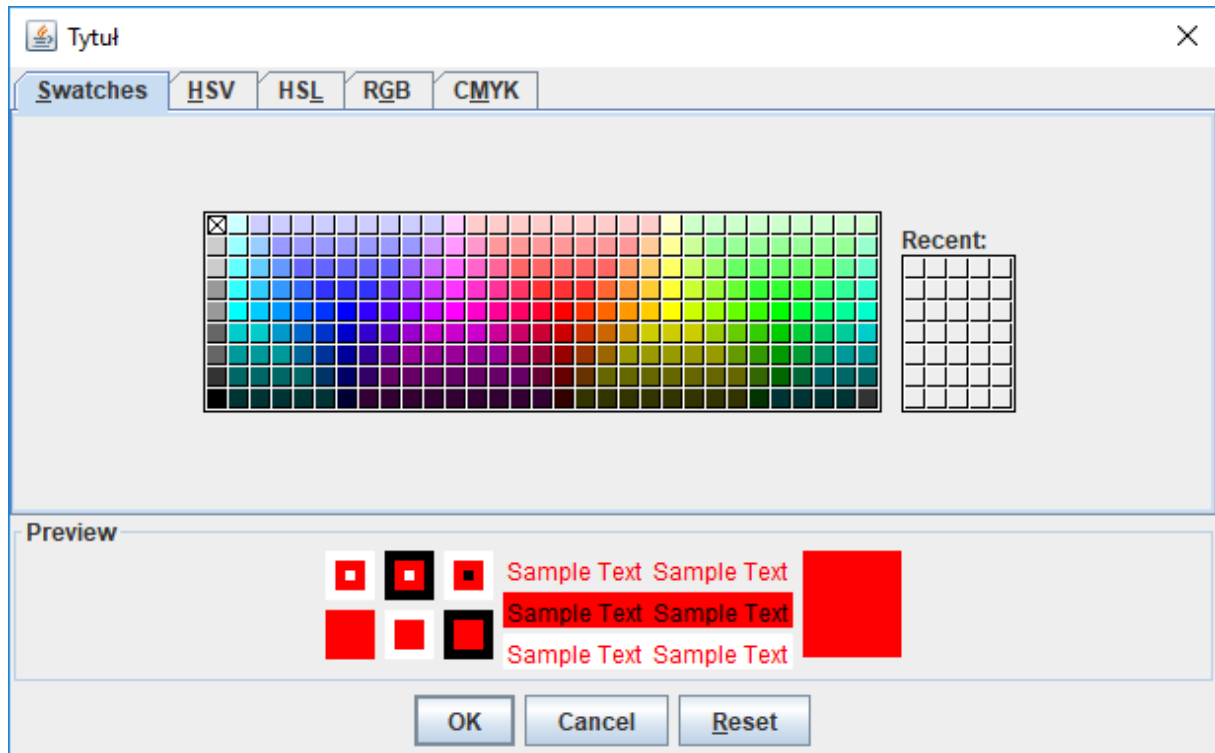


```
int result;  
JFileChooser chooser = new JFileChooser();  
chooser.setCurrentDirectory(new File("."));  
// ustawienie aktualnej ścieżki na katalog roboczy  
result = chooser.showOpenDialog(pane);
```



```
Color kolor = JColorChooser.showDialog(pane, "Tytuł",  
Color.RED);
```

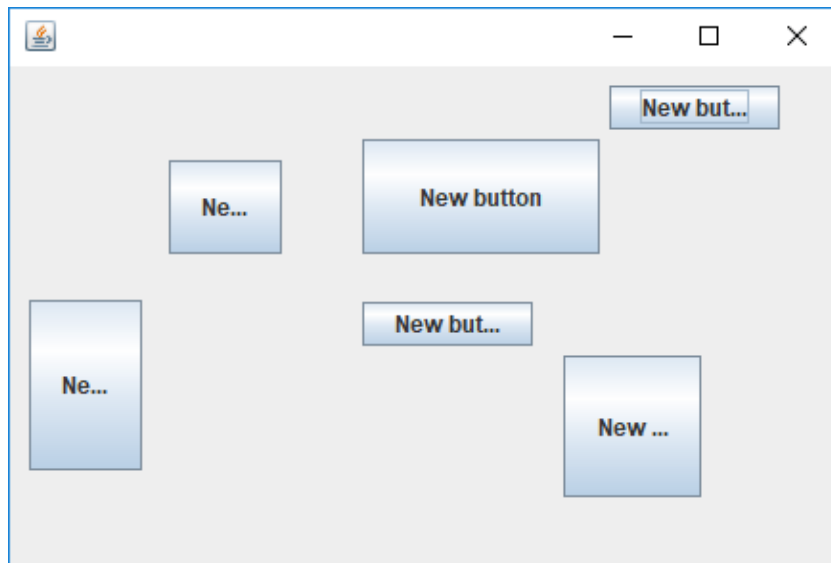
Ostatni parametr to kolor który po otwarciu okna jest domyślny.



Zarządzanie rozkładem

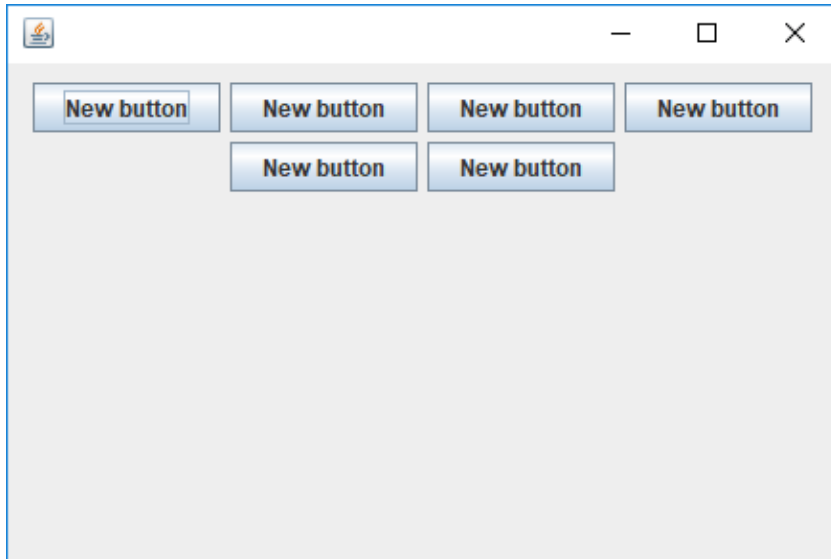
Absolute Layout:

```
pane.setLayout(null);
```



Flow layout:

```
pane.setLayout(new FlowLayout(FlowLayout.CENTER, hgap, vgap));
```

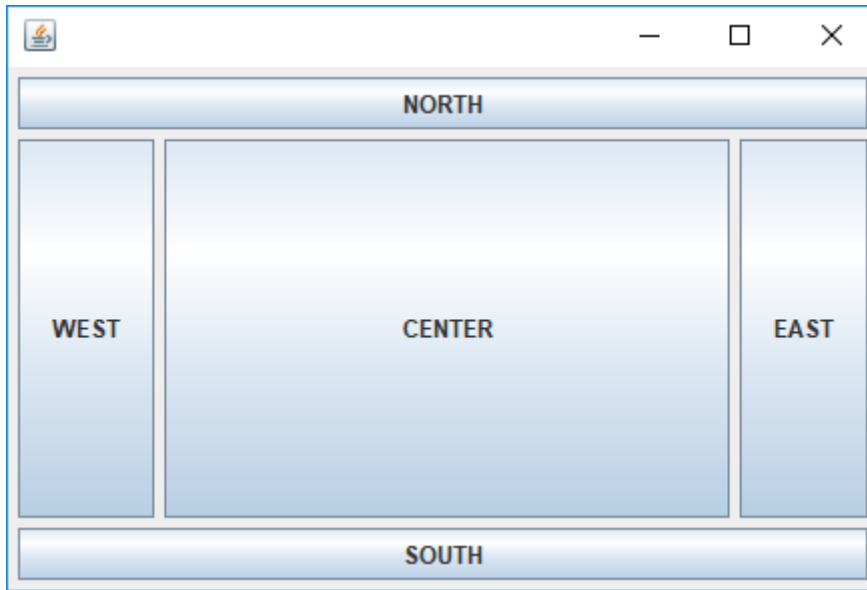


FlowLayout.CENTER - Wyrównanie do środka. Może być również RIGHT lub LEFT.

hgap, vgap – odstęp pomiędzy kolejnymi komponentami w rzędach i kolumnach

Border Layout:

```
pane.setLayout(new BorderLayout(hgap, vgap));
```



`hgap, vgap` – odstęp pomiędzy kolejnymi komponentami w rzędach i kolumnach

```
pane.add(btnNewButton_1, BorderLayout.WEST);
```

Każdy przycisk musi mieć ustawiony brzeg.

Grid Layout:

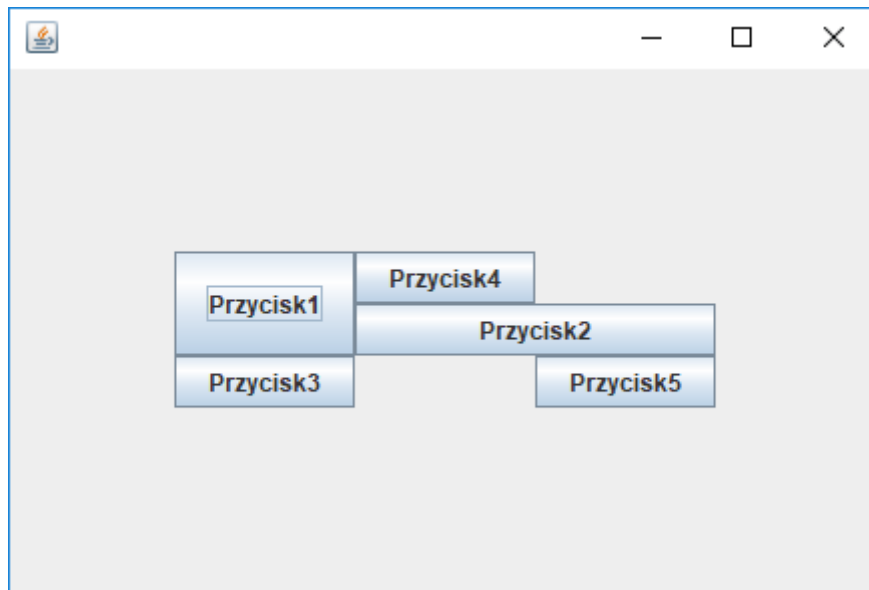
```
pane.setLayout(new GridLayout(row, columns, hgap, vgap));
```



hgap, vgap – odstęp pomiędzy kolejnymi komponentami w rzędach i kolumnach

row, columns -ilość rzędów i kolumn

GridBagLayout:



```

public Okno() {
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 450, 300);
    contentPane = new JPanel();
    setContentPane(contentPane);
    contentPane.setLayout(new GridBagLayout());

    b1 = new JButton("Przycisk1");
    gbc.gridx = 0;
    gbc.gridy = 0;
    gbc.gridheight = 2;
    gbc.fill= GridBagConstraints.VERTICAL;
    add(b1,gbc);

    b2 = new JButton("Przycisk2");
    gbc.gridx = 1;
    gbc.gridy = 1;
    gbc.gridwidth = 3;
    gbc.gridheight = 1;
    gbc.fill= GridBagConstraints.HORIZONTAL;
    add(b2,gbc);

    b3 = new JButton("Przycisk3");
    gbc.gridwidth = 1;
    gbc.gridx = 0;
    gbc.gridy = 2;
    add(b3,gbc);

    b4 = new JButton("Przycisk4");
    gbc.gridx = 1;
    gbc.gridy = 0;
    add(b4,gbc);

    b5 = new JButton("Przycisk5");
    gbc.gridx = 2;
    gbc.gridy = 2;
    add(b5,gbc);
}

```