

Materiał pomocniczy do kursu „Podstawy programowania”

Autor: Grzegorz Góralski

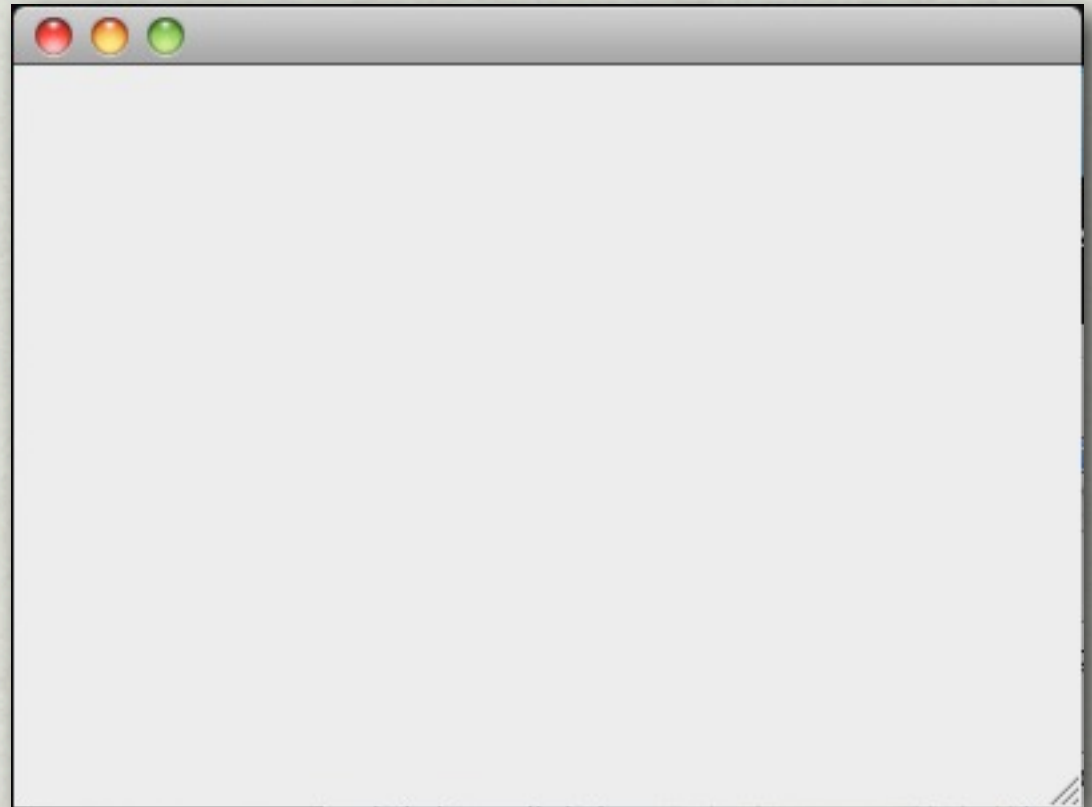
ggoralski.com

GUI-Swing

Wstęp do tworzenia prostych aplikacji z interfejsem graficznym (GUI) przy pomocy Swing, rysowanie prostych elementów graficznych

Tworzymy pierwsze okno

```
import javax.swing.*;
public class PierwszeGUI extends JFrame {
    public static void main(String[] args) {
        JFrame ramka = new JFrame();
        ramka.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        ramka.setSize(400, 300);
        ramka.setVisible(true);
    }
}
```

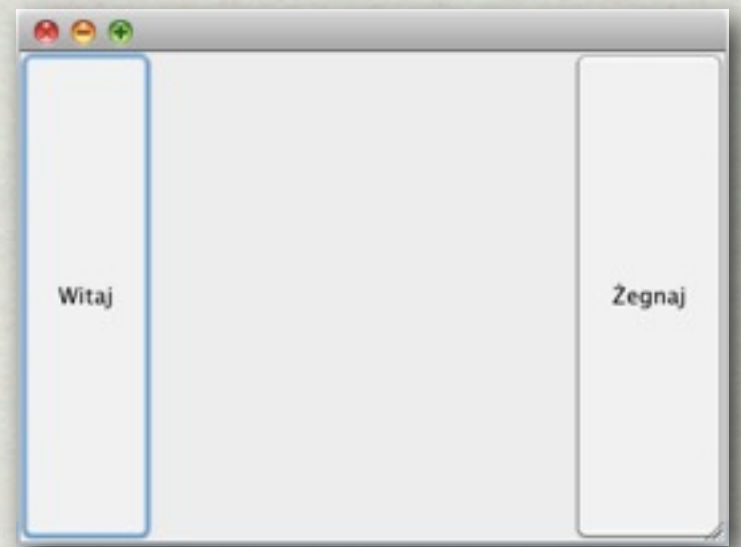


Tworzymy okno z przyciskami

```
import java.awt.BorderLayout;
import javax.swing.JButton;
import javax.swing.JFrame;

public class Przyciski {

    public static void main(String[] args) {
        // Tworzymy okno aplikacji
        JFrame ramka = new JFrame();
        // Po zamknięciu okna, aplikacja ma zakończyć działanie
        ramka.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        // Ustalamy rozmiar okna
        ramka.setSize(400, 300);
        // Tworzymy przycisk
        JButton witajButton = new JButton();
        // Ustawiamy jaki tekst ma się wyświetlać na przycisku
        witajButton.setText("Witaj");
        // Można od razu utworzyć przycisk i ustawić tekst
        JButton zegnajButton = new JButton("Żegnaj");
        // Wstawiamy przycisk, umieszczając go po lewej
        ramka.getContentPane().add(BorderLayout.WEST, witajButton);
        // Wstawiamy przycisk, umieszczając go po prawej
        ramka.getContentPane().add(BorderLayout.EAST, zegnajButton);
        // Okno się wyświetla
        ramka.setVisible(true);
    }
}
```



Tworzymy okno z przyciskami które coś robią

```
import java.awt.BorderLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.*;

public class PrzyciskiZAkcja implements ActionListener {
    JFrame ramka;
    JButton witajButton;
    JButton zegnajButton;

    public static void main(String[] args) {
        PrzyciskiZAkcja przyciski = new PrzyciskiZAkcja();
        przyciski.zbudujGUI();
    }
    ....
}
```

Tworzymy okno z przyciskami które coś robią

....

```
public void zbudujGUI() {  
    ramka = new JFrame();  
    ramka.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
    ramka.setSize(400, 300);  
    witajButton = new JButton("Witaj");  
    zegnajButton = new JButton("Żegnaj");  
  
    // dodajemy do przycisków ActionListener  
    witajButton.addActionListener(this);  
    zegnajButton.addActionListener(this);  
  
    ramka.getContentPane().add(BorderLayout.WEST, witajButton);  
    ramka.getContentPane().add(BorderLayout.EAST, zegnajButton);  
    ramka.setVisible(true);  
}
```

....

Tworzymy okno z przyciskami które coś robią

....

```
public void actionPerformed(ActionEvent akcja) {  
    // metoda obsługująca akcje  
    if (akcja.getSource() == witajButton)  
        System.out.println("Witaj robaczku!");  
    else if (akcja.getSource() == zegnajButton)  
        System.out.println("Żegnaj robaczku!");  
    else  
        // to się nie powinno wydarzyć ;-)  
        System.out.println("Skontaktowali się ze mną kosmici");  
}
```

```
}
```

Tworzymy okno z przyciskami które coś robią, z tekstem w naszym GUI

```
import java.awt.BorderLayout;
import java.awt.FlowLayout;
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.*;

public class PrzyciskiITekst implements ActionListener{
    JFrame ramka;
    JButton witajButton;
    JButton zegnajButton;
    JTextField textF;
    JLabel rezultatJL;

    public static void main(String[] args) {
        PrzyciskiITekst przyciski = new PrzyciskiITekst();
        przyciski.zbudujGUI();
    }
    ....
}
```

Tworzymy okno z przyciskami które coś robią, z tekstem w naszym GUI

....

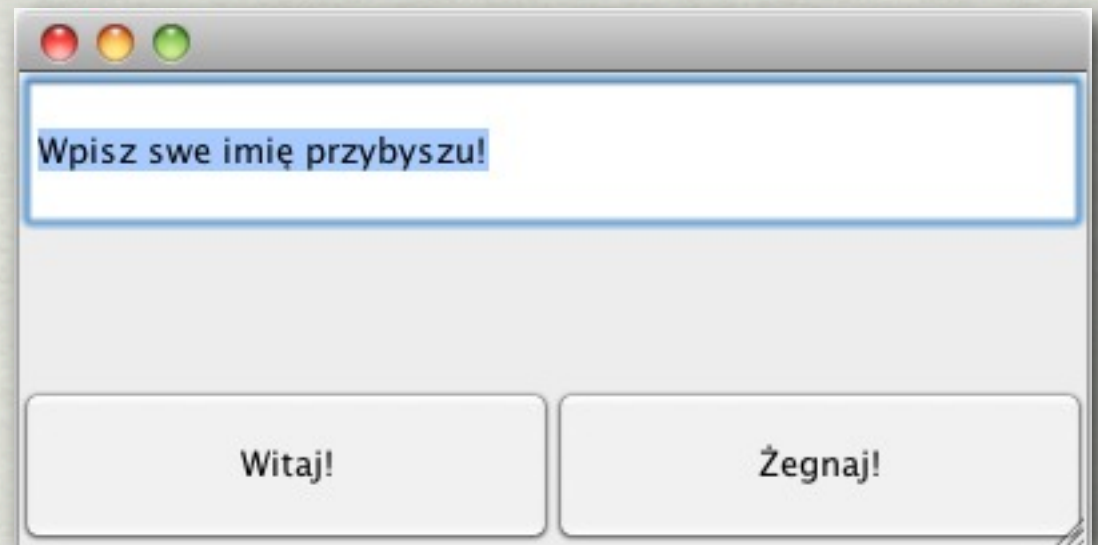
```
public void zbudujGUI() {
    ramka = new JFrame();
    ramka.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    ramka.setSize(400, 200);
    witajButton = new JButton();
    witajButton.setText("Witaj!");
    zegnajButton = new JButton("Żegnaj!");
    witajButton.addActionListener(this);
    zegnajButton.addActionListener(this);
    JPanel jp = new JPanel();
    jp.setLayout(new GridLayout(1,2));
    jp.add(witajButton);
    jp.add(zegnajButton);
    JPanel jp2 = new JPanel();
    jp2.setLayout(new GridLayout(3,1));
    // tworzymy pole tekstowe z domyślnym tekstem
    textF = new JTextField("Wpisz swe imię przybyszu!");
    // tworzymy pole na którym będzie wypisany tekst
    rezultatJL = new JLabel();
    rezultatJL.setSize(400,20);
    jp2.add(textF);
    jp2.add(rezultatJL);
    jp2.add(jp);
    ramka.add(jp2);
    ramka.setVisible(true);
}
```

....

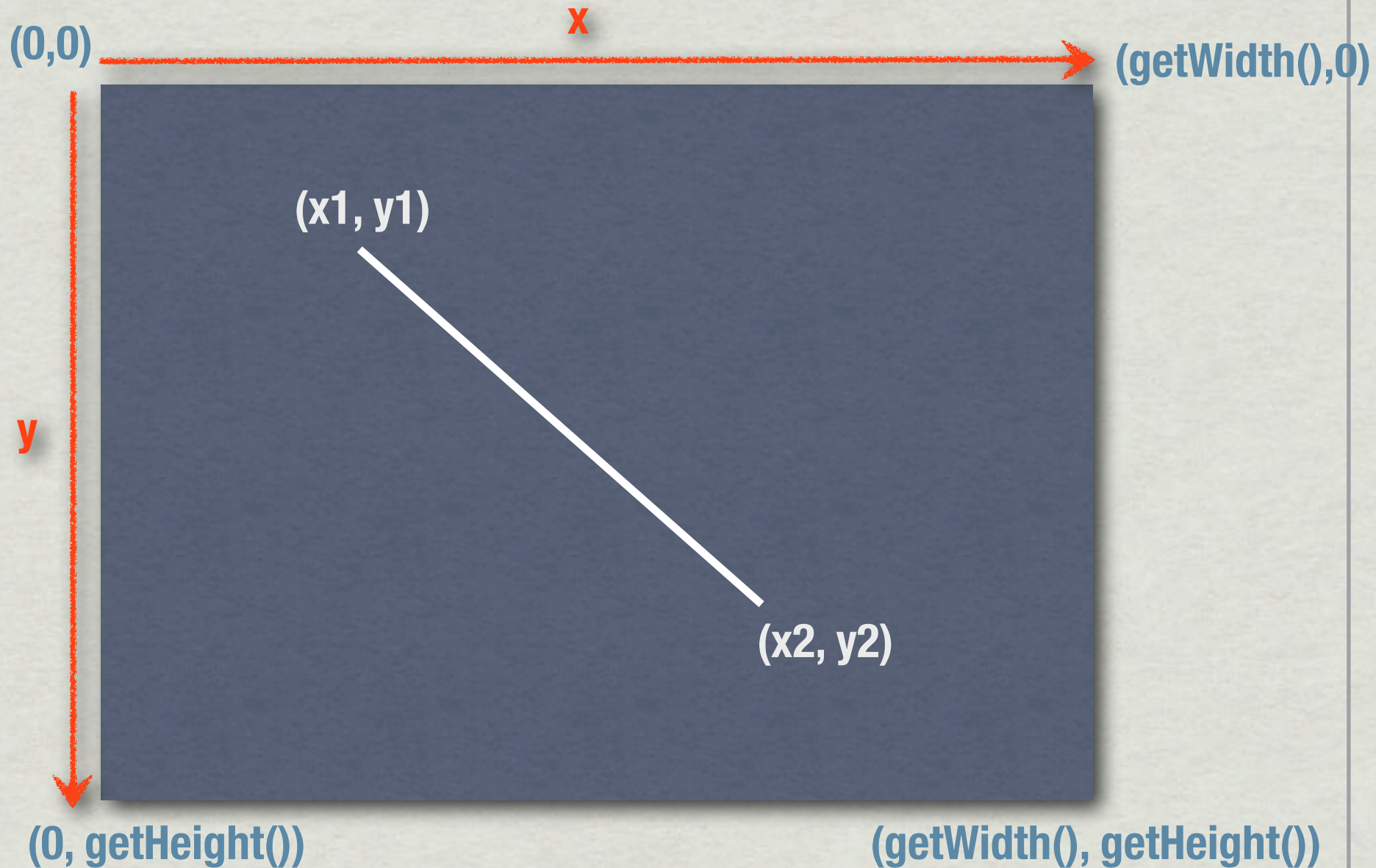
Tworzymy okno z przyciskami które coś robią, z tekstem w naszym GUI

....

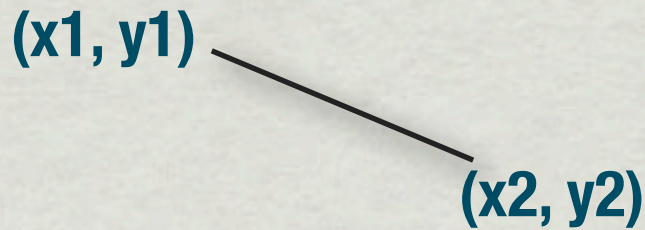
```
public void actionPerformed(ActionEvent akcja) {  
    if (akcja.getSource() == witajButton)  
        // modyfikujemy komponent, zmieniając tekst  
        rezultatJL.setText(" Witaj "+textField.getText());  
    else if (akcja.getSource() == zegnajButton)  
        // modyfikujemy komponent, zmieniając tekst  
        rezultatJL.setText(" Żegnaj "+textField.getText());  
  
    else  
        System.out.println("Skontaktowali się ze mną kosmici");  
}
```



Grafika - układ współrzędnych



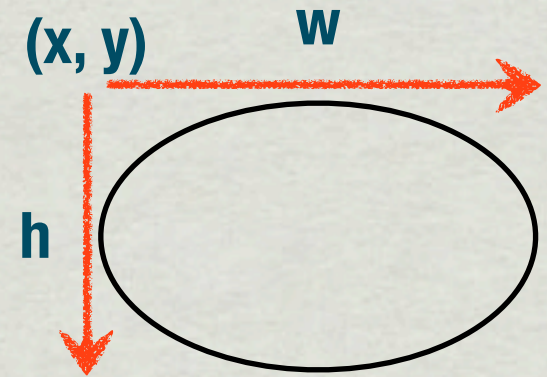
Grafika - proste figury



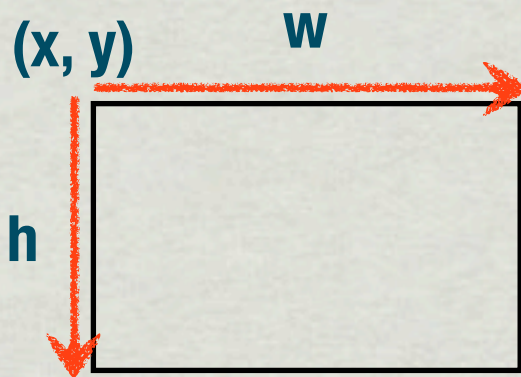
(x, y) string

`drawLine(x1, y1, x2, y2)`

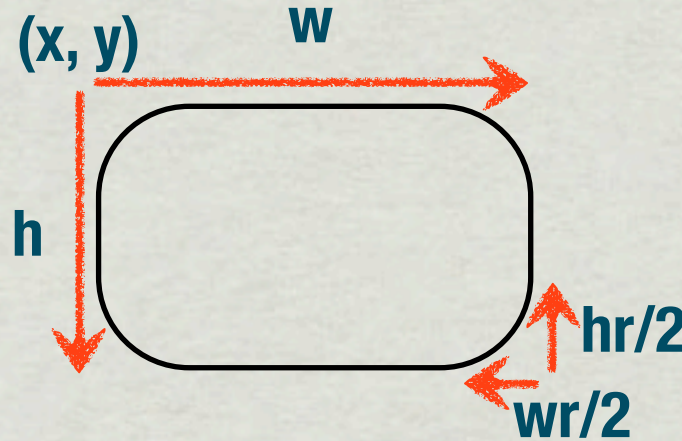
`drawString(string, x, y)`



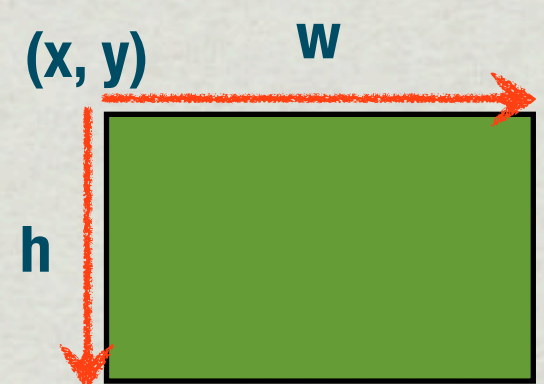
`drawOval(x, y, w, h)`



`drawRect(x, y, w, h)`



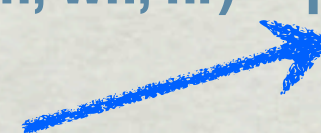
`drawRoundRect(x, y, w, h, wh, hr)`



`fillRect(x, y, w, h)`

analogicznie: `fillRoundRect(...)`, `fillOval(...)` itd.

Uwaga: wcześniej trzeba ustawić kolor wypełnienia (patrz przykład poniżej)



Aplikacja z panelem rysująca figury geometryczne

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

public class Rysunek implements ActionListener {
    JFrame ramka;
    JButton liniaButton;
    JButton kwadratButton;
    JButton kołoButton;
    JButton wzorekButton;
    PanelRysunkowy rysunkowyP;

    public static void main(String[] args) {
        Rysunek rysunek = new Rysunek();
        rysunek.zbudujGUI();
    }
}
```

```

public void zbudujGUI() {
    ramka = new JFrame();
    ramka.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    ramka.setSize(400, 400);

    liniaButton = new JButton("Linia");
    kwadratButton = new JButton("Kwadrat");
    koloButton = new JButton("Kolo");
    wzorekButton = new JButton("Wzorek");

    liniaButton.addActionListener(this);
    kwadratButton.addActionListener(this);
    koloButton.addActionListener(this);
    wzorekButton.addActionListener(this);

    JPanel jp = new JPanel();
    jp.setLayout(new BorderLayout());

    rysunkowyP = new PanelRysunkowy();
    JToolBar tb = new JToolBar();

    tb.add(liniaButton);
    tb.add(kwadratButton);
    tb.add(koloButton);
    tb.add(wzorekButton);
    jp.add("Center", rysunkowyP);
    jp.add("North", tb);
    ramka.add(jp);
    ramka.setVisible(true);
}

```

```
public void actionPerformed(ActionEvent akcja) {
    Graphics gr = rysunkowyP.getGraphics();
    gr.clearRect(0, 0, 400, 400);
    if (akcja.getSource() == liniaButton)
        gr.drawLine(20, 20, 200, 200);
    else if (akcja.getSource() == kwadratButton) {
        gr.drawRect(20, 20, 220, 220);
        gr.setColor(Color.BLUE);
        gr.fillRoundRect(50, 50, 100, 100, 40, 40);
    }
    else if (akcja.getSource() == koloButton) {
        gr.setColor(Color.GREEN);
        gr.fillOval(20, 20, 200, 200);
    }
    else if (akcja.getSource() == wzorekButton)
        malujWzorek(rysunkowyP, (int) rysunkowyP.getWidth() / 2,
                    (int) rysunkowyP.getHeight() / 2, 200);
    else
        System.out.println("Skontaktowali się ze mną kosmici");
}
```

Ustawiamy kolor wypełnienia

```
public void malujWzorek(PanelRysunkowy pr, int x, int y, int r) {  
    rysunkowyP.getGraphics().drawOval(x - r / 2, y - r / 2, r, r);  
  
    if (r > 1) {  
        malujWzorek(pr, x + r / 2, y, r / 2);  
        malujWzorek(pr, x - r / 2, y, r / 2);  
    }  
}  
}
```

Druga klasa w tym samym pliku!

Tak, jest to możliwe.

```
class PanelRysunkowy extends JPanel {  
    public void paintComponent(Graphics g) {  
        super.paintComponents(g);  
    }  
}
```



Co dalej?

Swing

<http://docs.oracle.com/javase/tutorial/uiswing/>

JavaFX

<http://www.oracle.com/technetwork/java/javafx/overview/index.html>

<http://docs.oracle.com/javafx/>