

Języki i metody programowania

– Java

Obsługa zdarzeń - przykłady

wg

<https://docs.oracle.com/javase/tutorial/uiswing/components/index.html>

Autor

Dr inż. Zofia Kruczkiewicz

Rodzaje słuchaczy zdarzeń

Wydarzenia można podzielić na dwie grupy:

- zdarzenia o niskim poziomie – zdarzenia myszy i klawiatury
- zdarzenia semantyczne – zdarzenia i akcje komponentów Swing (działania użytkownika lub innych programów np bazy danych)

Zaleca się używać zdarzeń semantycznych, bo sprzyja to przenośności programu

Lista komponentów Swing z wykazem obsługujących ich słuchaczy zdarzeń (zdarzenia semantyczne)

Component	Action Listener	Caret Listener	Change Listener	Document Listener, Undoable Edit Listener	Item Listener	List Selection Listener	Window Listener	Other Types of Listeners
button	✓		✓		✓			
check box	✓		✓		✓			
color chooser			✓					
combo box	✓				✓			
dialog							✓	
editor pane		✓		✓				hyperlink
file chooser	✓							
formatted text field	✓	✓		✓				

(cd) Lista komponentów Swing z wykazem obsługujących ich słuchaczy zdarzeń

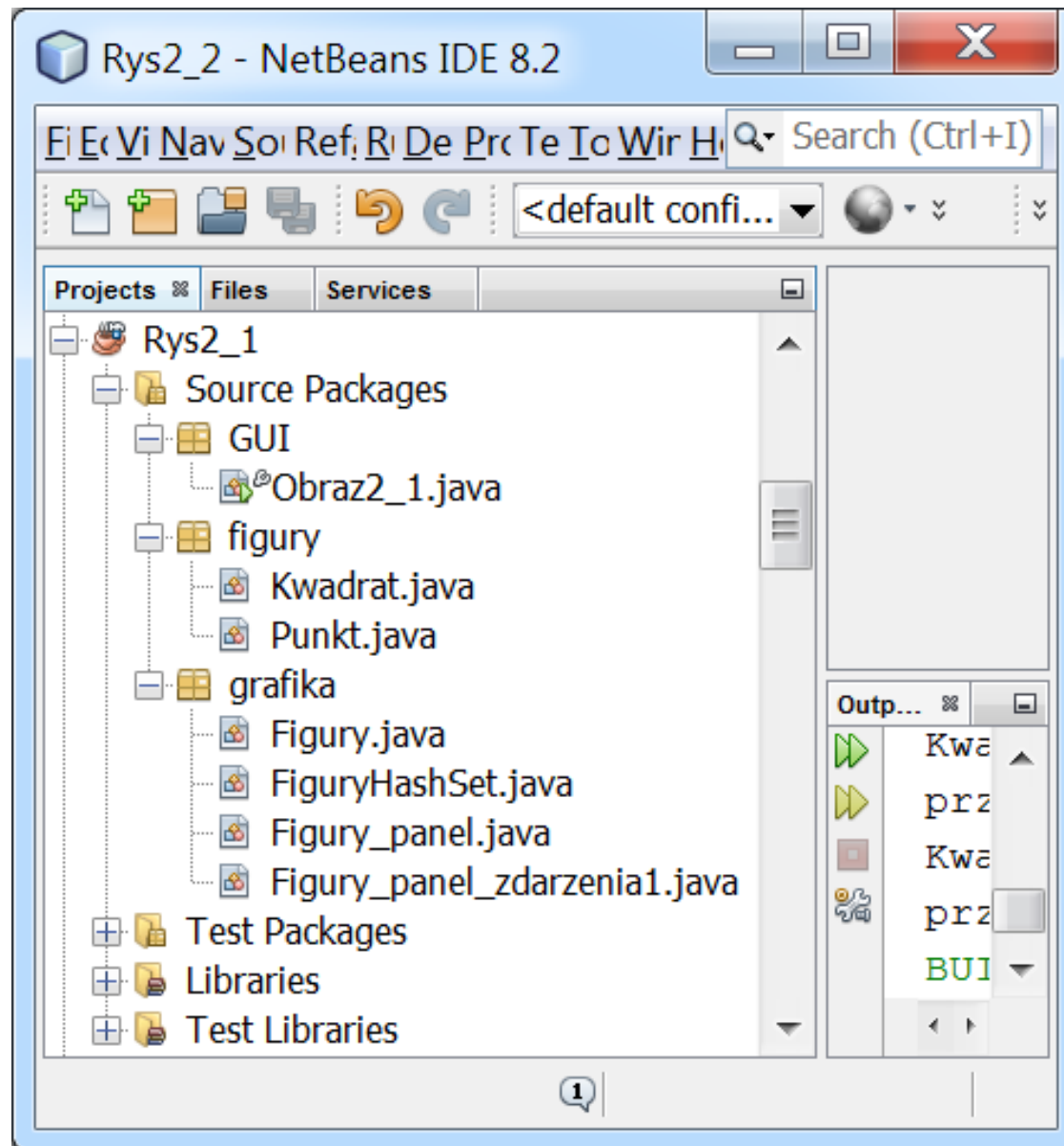
progress bar			✓					
radio button	✓		✓		✓			
slider			✓					
spinner			✓					
tabbed pane			✓					
table						✓		table model table column model cell editor

(cd) Lista komponentów Swing z wykazem obsługujących ich słuchaczy zdarzeń

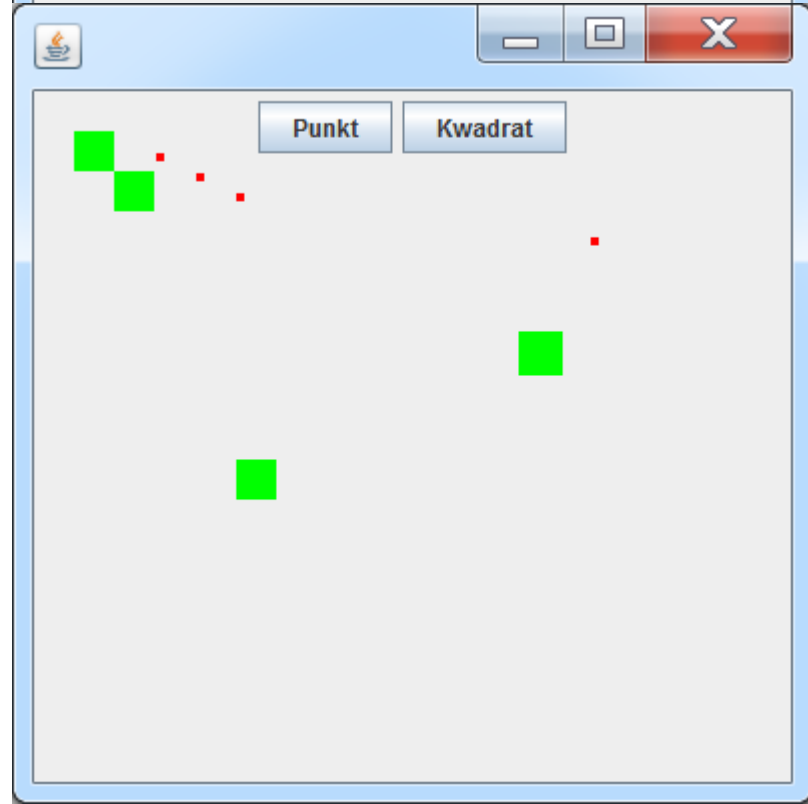
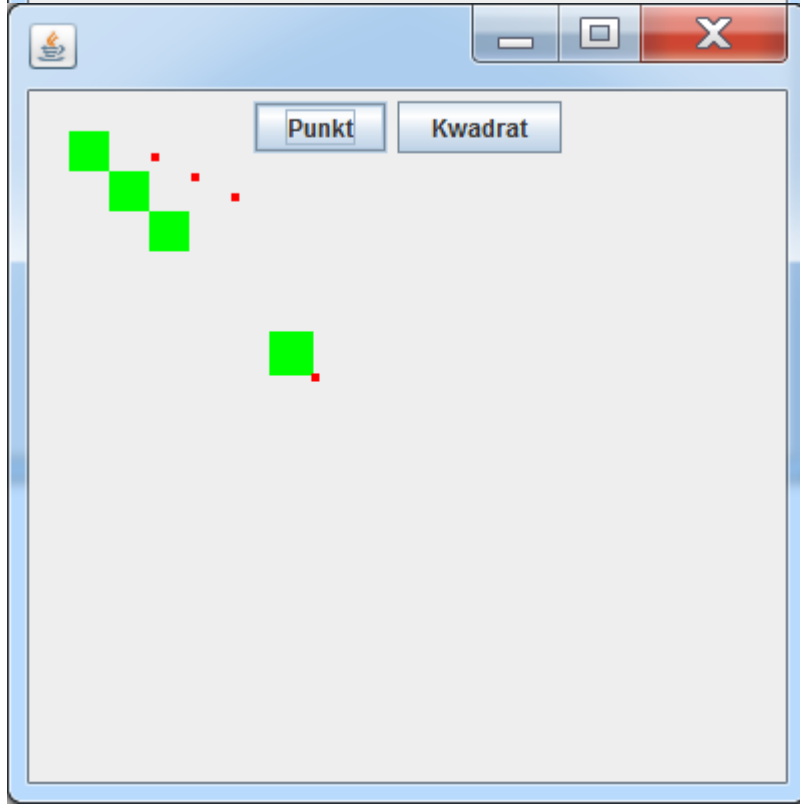
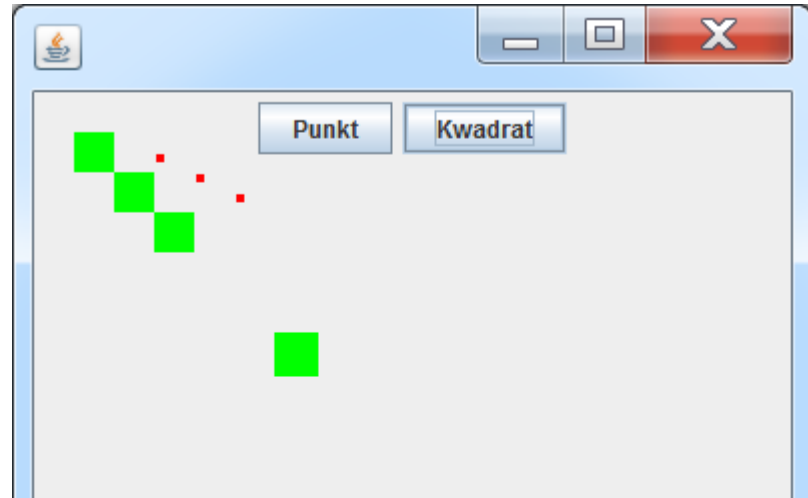
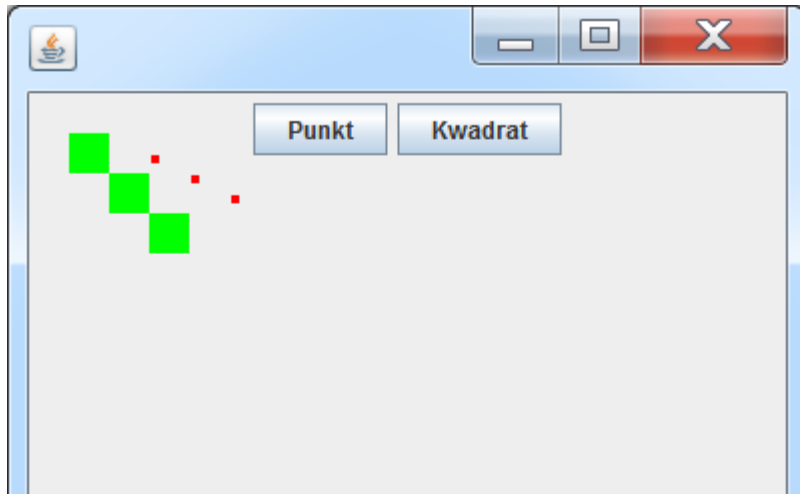
text area		✓		✓				
text field	✓	✓		✓				
text pane		✓		✓				hyperlink
toggle button	✓		✓		✓			
tree								tree expansion tree will expand tree model tree selection
viewport (used by scrollpane)			✓					

- [How to Write an Action Listener](#)
- [How to Write a Caret Listener](#)
- [How to Write a Change Listener](#)
- [How to Write a Component Listener](#)
- [How to Write a Container Listener](#)
- [How to Write a Document Listener](#)
- [How to Write a Focus Listener](#)
- [How to Write an Internal Frame Listener](#)
- [How to Write an Item Listener](#)
- [How to Write a Key Listener](#)
- [How to Write a List Data Listener](#)
- [How to Write a List Selection Listener](#)
- [How to Write a Mouse Listener](#)
- [How to Write a Mouse-Motion Listener](#)
- [How to Write a Mouse-Wheel Listener](#)
- [How to Write a Property Change Listener](#)
- [How to Write a Table Model Listener](#)
- [How to Write a Tree Expansion Listener](#)
- [How to Write a Tree Model Listener](#)
- [How to Write a Tree Selection Listener](#)
- [How to Write a Tree-Will-Expand Listener](#)
- [How to Write an Undoable Edit Listener](#)
- [How to Write Window Listeners](#)

Przykład 1 obsługi zdarzeń – projekt Rys2_1



(cd) Przykład 1 obsługi zdarzeń – działanie programu



(cd) Przykład 1 obsługi zdarzeń – projekt Rys2_1

```
package figury;

import java.awt.Color;
import java.awt.Graphics;
import java.awt.Graphics2D;
public class Punkt implements Comparable {
    protected int x, y;

    public Punkt(int wspX, int wspY) {
        x = wspX;
        y = wspY; }

    @Override
    public int compareTo(Object o) {
        Punkt p = (Punkt) o;
        if ((x == p.x) && (y == p.y)) {
            return 0;
        } else if ((x < p.x) && (y < p.y)) {
            return -1;
        }
        return 1; }
}
```

```
public int getX() { return x; }

public int getY() { return y; }

@Override
public int hashCode() {
    int hash = 7;
    hash = 17 * hash + this.x;
    hash = 17 * hash + this.y;
    return hash; }

@Override
public boolean equals(Object obj) {
    if (this == obj) {
        return true; }
    if (obj == null) {
        return false; }
    if (getClass() != obj.getClass()) {
        return false; }
    return this.compareTo(obj) == 0;
}
```

(cd) Przykład 1 obsługi zdarzeń – projekt Rys2_1

```
public double odleglosc(Punkt p) {  
    return Math.sqrt((x - p.x) * (x - p.x) + (y - p.y) * (y - p.y)); }  
}
```

@Override

```
public String toString() {  
    return "Punkt{" + "x=" + x + ", y=" + y + '}'; }  
}
```

```
public int getDI() {    return 5; }  
}
```

```
public void przesun(int dx, int dy, int a, int b) {  
    x += dx;  
    y += dy;  
    if (x > a || x < 1)  
        x = 5;  
    if (y > b || y < 1)  
        y = 2;  
}  
}
```

```
public void rysuj(Graphics g) {  
    Graphics2D g2D = (Graphics2D) g;  
    Color pedzel = new Color(255, 0, 0);  
    g2D.setColor(pedzel);  
    g2D.fillOval(x, y, 5, 5);}  
}
```

(cd) Przykład 1 obsługi zdarzeń – projekt Rys2_1

```
package figury;

import java.awt.Color;
import java.awt.Graphics;
import java.awt.Graphics2D;

public class Kwadrat extends Punkt {

    protected int dlugosc;

    public Kwadrat(int wspX, int wspY, int dlugosc_) {
        super(wspX, wspY);
        dlugosc = dlugosc_; }

    public double odlegosc() {
        return Math.sqrt(x * x + y * y);
    }

    @Override
    public double odlegosc(Punkt p) {
        return odlegosc() + super.odlegosc(p); }

    public int getDI() {
        return dlugosc; }
```

```
@Override
public int hashCode() {
    int hash = 5;
    hash = 97 * hash + this.dlugosc;
    return hash; }

@Override
public boolean equals(Object obj) {
    if (this == obj) {
        return true; }
    if (obj == null) {
        return false; }
    if (getClass() != obj.getClass()) {
        return false;
    }
    final Kwadrat other = (Kwadrat) obj;
    if (this.dlugosc != other.dlugosc) {
        return false; }
    return this.compareTo(obj)==0;
}
```

(cd) Przykład 1 obsługi zdarzeń – projekt Rys2_1

@Override

```
public String toString() {  
    String s = super.toString();  
    return "Kwadrat{" + "dlugosc=" + dlugosc + '}' + " i dziedzicze od " + s;  
}
```

```
public void rysuj(Graphics g) {  
    Graphics2D g2D = (Graphics2D) g;  
    Color pedzel = new Color(0, 255, 0);  
    g2D.setColor(pedzel);  
    g2D.fillRect(x, y, dlugosc, dlugosc);  
}  
}
```

(cd) Przykład 1 obsługi zdarzeń – projekt Rys2_1

```
package grafika;

import figury.Kwadrat;
import figury.Punkt;
import java.awt.Graphics;
import java.util.Collection;

public class Figury {

    protected int N = 3;
    public Collection<Punkt> figura;

    protected Punkt biezacy;

    public void polozenie() {
        for (Punkt figura : figura) {
            boolean p = figura instanceof Kwadrat;
            System.out.println( p +
                ", ze jestem kwadratem, bo jestem " +
                figura.toString()
                + ", X=" + figura.getX()
                + ", Y=" + figura.getY()
                + ", odleglosc=" + figura.odleglosc(figura)); }
    }

    public Punkt getBiezacy() { return biezacy; }
```

```
public boolean Clicked(int x_, int y_) {
    for (Punkt figura : figura) {
        if (figura.getX() + figura.getDI() >= x_
            && figura.getX() <= x_
            && figura.getY() + figura.getDI() >= y_
            && figura.getY() <= y_) {
            /*if(figura.lezy_na(x_, y_)) */
            biezacy = figura;
            return true; } }
        return false; }

    public void rysuj_figury(Graphics g) {
        for (Punkt figura : figura) {
            figura.rysuj(g); }
    }

    public boolean przesun(int x, int y, int dl,
        int szer) {

        if (biezacy != null) {
            biezacy.przesun(x, y, dl, szer);
            return true; }
        return false; }
}
```

(cd) Przykład 1 obsługi zdarzeń – projekt Rys2_1

```
package grafika;

import figury.Kwadrat;
import figury.Punkt;
import java.util.ArrayList;

public class FiguryHashSet extends Figury {

    public void pojemnik() {
        figury = new HashSet(); }

    public boolean wyszukaj(Punkt p) {
        return figury.contains(p);
    }

    public boolean wyszukaj(int dane[]) {
        Punkt p;
        if (dane[0] == 0) {
            p = new Punkt(dane[1], dane[2]);
        } else {
            p = new Kwadrat(dane[1], dane[2], dane[3]);
        }
        return wyszukaj(p); }
}
```

```
public void wypelnij() {
    for (int i = 0; i < N; i++) {
        figury.add(new Punkt(20 * (N + i),
            10 * (N + i)));
        figury.add(new Kwadrat((i + 1) * 20,
            (i + 1) * 20, 20)); }
    }

    public void Dodaj_punkt() {
        int a = figury.size();
        figury.add(new Punkt(a * 20, a * 20)); }

    public void Dodaj_kwadrat() {
        int a = figury.size();
        figury.add(new Kwadrat(a * 20, a * 20, 22));
    }

    public void init() {
        pojemnik();
        wypelnij();
        polozenie(); }
}
```

(cd) Przykład 1 obsługi zdarzeń – projekt Rys2_1

```
package grafika;

import java.awt.Graphics;
import javax.swing.JPanel;

public class Figury_panel extends JPanel{
    FiguryHashSet kontroler;

    @Override
    protected void paintComponent(Graphics g) {
        super.paintComponent(g);
        kontroler.rysuj_figury(g);
    }

    public void init()
    { kontroler=new FiguryArrayList();
      kontroler.pojemnik();
      kontroler.wypelnij();
    }

    public FiguryArrayList getKontroler() {
        return kontroler; }
}
```


(cd) Przykład 1 obsługi zdarzeń – projekt Rys2_1

```
package grafika;

import java.awt.FlowLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;
import java.awt.event.MouseEvent;
import java.awt.event.MouseListener;
import java.util.Random;
import javax.swing.JButton;

public class Figury_panel_zdarzenia1 extends Figury_panel
implements KeyListener, MouseListener, ActionListener{

    protected JButton punkt = new JButton("Punkt");
    protected JButton kwadrat = new JButton("Kwadrat");
```

@Override

```
public void mouseClicked(MouseEvent e)
{ int x_ = e.getX();
  int y_ = e.getY();
  kontroler.Clicked(x_, y_);
  requestFocus();
  repaint(); }
```

```
public void mouseEntered(
    MouseEvent e)
{ }
```

```
public void mouseExited(
    MouseEvent e)
{ }
```

```
public void mousePressed(
    MouseEvent e)
{ }
```

```
public void mouseReleased(
    MouseEvent e)
{ }
```

(cd) Przykład 1 obsługi zdarzeń – projekt Rys2_1

@Override

```
public void keyPressed(KeyEvent evt) {
    int dx = 0, dy = 0, zmiana;
    Random r = new Random();
    if (evt.isShiftDown()) {
        zmiana = r.nextInt(100);
    } else {
        zmiana = r.nextInt(10);
    }
    switch (evt.getKeyCode()) {
        case KeyEvent.VK_LEFT:
            dx -= zmiana; break;
        case KeyEvent.VK_RIGHT:
            dx += zmiana; break;
        case KeyEvent.VK_UP:
            dy -= zmiana; break;
        case KeyEvent.VK_DOWN:
            dy += zmiana; break;
    }
    kontroler.przesun(dx, dy,
        getWidth(), getHeight());
    repaint();
}
```

@Override

```
public void keyReleased(KeyEvent evt) { }
```

@Override

```
public void keyTyped(KeyEvent evt)
    /*obsługa klawiszy (a, A, #, ...)*/ { }
```

@Override

```
public void actionPerformed(ActionEvent e)
{
    Object zrodlo = e.getSource();
    if (zrodlo == punkt) {
        kontroler.Dodaj_punkt();
    } else if (zrodlo == kwadrat) {
        kontroler.Dodaj_kwadrat();
    }
    repaint();
}
```

(cd) Przykład 1 obsługi zdarzeń – projekt Rys2_1

```
@Override
public void init() {
    super.init();
    setLayout(new FlowLayout());
    addKeyListener(this);
    setFocusable(true);
    addMouseListener(this);
    punkt.addActionListener(this);
    kwadrat.addActionListener(this);
    add(punkt);
    add(kwadrat);
}
}
```

(cd) Przykład 1 obsługi zdarzeń – projekt Rys2_1

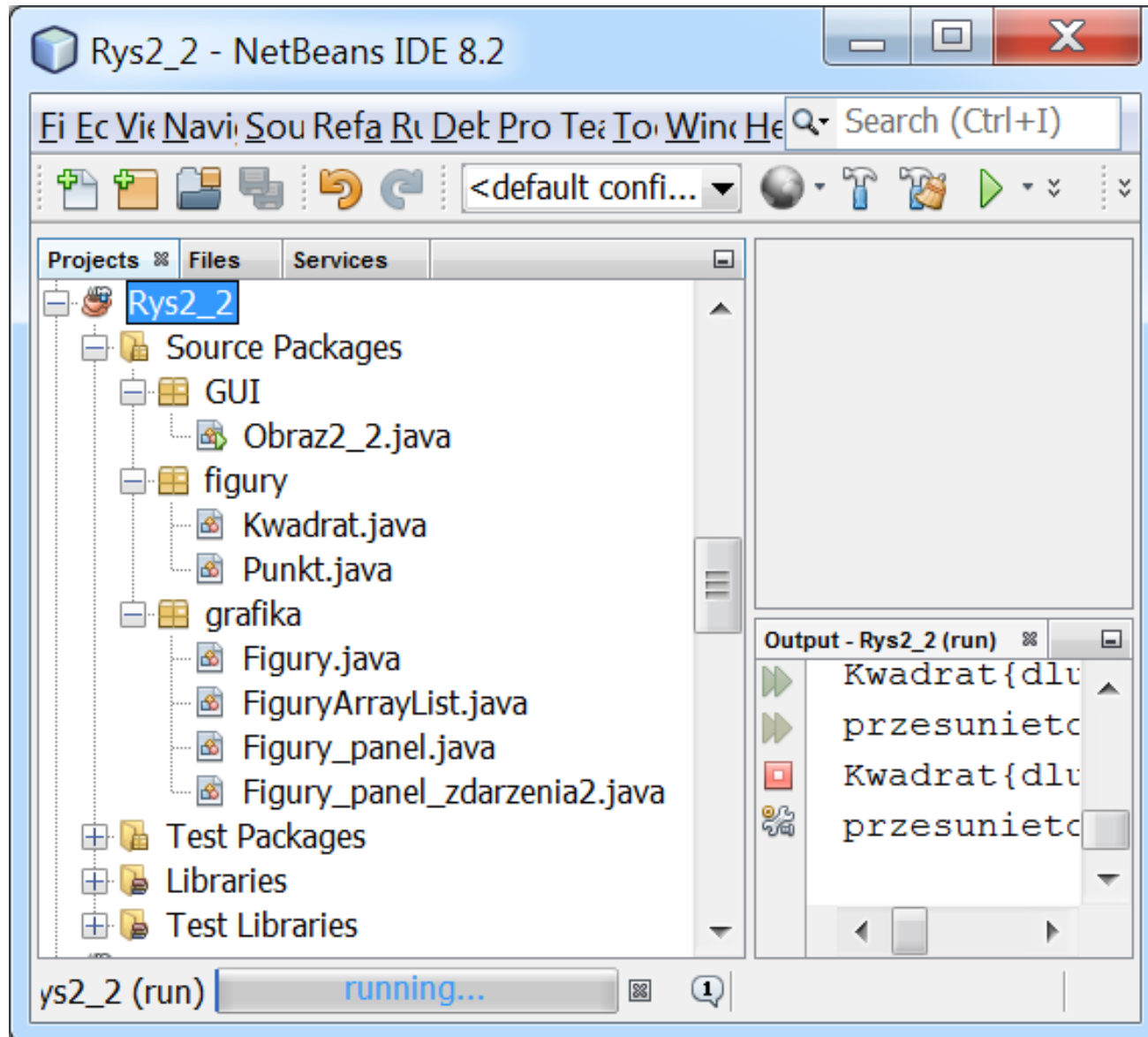
```
package GUI;

import grafika.FiguryArrayList;
import grafika.Figury_panel_zdarzenia1;
import javax.swing.JFrame;

public class Obraz2_1 {
    void rysunek_Swing() {
        JFrame ramka = new JFrame();
        Figury_panel_zdarzenia1 panel = new Figury_panel_zdarzenia1();
        panel.init();
        ramka.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        ramka.setSize(400, 400);
        ramka.setContentPane(panel);
        ramka.setVisible(true); }

    public static void main(String args[]) {
        Obraz2_1 obraz = new Obraz2_1();
        java.awt.EventQueue.invokeLater(new Runnable() {
            @Override
            public void run() {
                obraz.rysunek_Swing(); }
        });
    }
}
```

Przykład 2 obsługi zdarzeń – projekt Rys2_2



(cd) Przykład 2 obsługi zdarzeń – projekt Rys2_2

```
package grafika;

import java.awt.FlowLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.util.Random;
import javax.swing.JButton;

public class Figury_panel_zdarzenia2
    extends Figury_panel implements ActionListener{

protected JButton punkt = new JButton("Punkt");
protected JButton kwadrat = new JButton("Kwadrat");

public void mouseClicked_(MouseEvent e) {
    int x_ = e.getX();
    int y_ = e.getY();
    kontroler.Clicked(x_, y_);
    requestFocus();
    repaint(); }
}
```

```
public void keyPressed_(KeyEvent evt)
{
    int dx = 0, dy = 0, zmiana;
    Random r = new Random();
    if (evt.isShiftDown()) {
        zmiana = r.nextInt(100);
    } else {
        zmiana = r.nextInt(10); }
    switch (evt.getKeyCode()) {
        case KeyEvent.VK_LEFT:
            dx -= zmiana;    break;
        case KeyEvent.VK_RIGHT:
            dx += zmiana;    break;
        case KeyEvent.VK_UP:
            dy -= zmiana;    break;
        case KeyEvent.VK_DOWN:
            dy += zmiana;    break;
    }
    kontroler.przesun(dx, dy,
        getWidth(), getHeight());
    repaint(); }
}
```

(cd) Przykład 2 obsługi zdarzeń – projekt Rys2_2

@Override

```
public void init() {  
    super.init();  
    setLayout(new FlowLayout());  
    addKeyListener(new KeyAdapter() {  
        @Override  
        public void keyPressed(KeyEvent evt) {  
            keyPressed_(evt);  
        }  
    });  
    setFocusable(true);  
    addMouseListener(new MouseAdapter() {  
        @Override  
        public void mouseClicked(MouseEvent e) {  
            mouseClicked_(e);  
        }  
    });  
    punkt.addActionListener(this);  
    kwadrat.addActionListener(this);  
    add(punkt);  
    add(kwadrat);  
}
```

@Override

```
public void actionPerformed(ActionEvent e) {  
    Object zrodlo = e.getSource();  
    if (zrodlo == punkt) {  
        kontroler.Dodaj_punkt();  
    } else if (zrodlo == kwadrat) {  
        kontroler.Dodaj_kwadrat();  
    }  
    repaint();  
}
```

(cd) Przykład 2 obsługi zdarzeń – projekt Rys2_2

```
package GUI;

import grafika.FiguryArrayList;
import grafika.Figury_panel_zdarzenia2;
import javax.swing.JFrame;

public class Obraz2_2 {
    void rysunek_Swing() {
        JFrame ramka = new JFrame();
        Figury_panel_zdarzenia2 panel = new Figury_panel_zdarzenia2();
        panel.init();
        ramka.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        ramka.setSize(400, 400);
        ramka.setContentPane(panel);
        ramka.setVisible(true); }

    public static void main(String args[]) {
        Obraz2_2 obraz = new Obraz2_2();
        java.awt.EventQueue.invokeLater(new Runnable() {
            @Override
            public void run() {
                obraz.rysunek_Swing(); }
        });
    }
}
```