

Swing (3. del)

Programiranje grafičnega vmesnika

Aplikacija Slikar



Vpravitelj postavitev (layout managers)

Organiziramo uporabniški vmesnik z uporabo **layout manager** (**vpravitelj postavitev**).

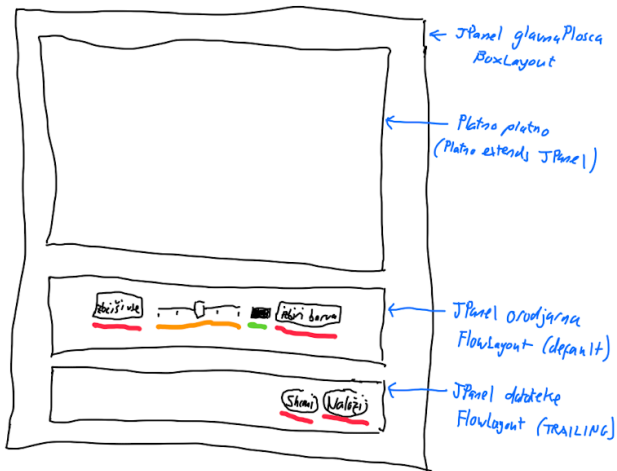
Java AWT (Abstract Windows Toolkit), ki ga uporablja Swing, ima vmesnik `LayoutManager`.

Uporabljamo razrede, ki implementirajo `LayoutManager`, postaviti Containers, na primer `JPanels`.

— JButton

— JSlider

— JLabel



```
public class Okno extends JFrame implements ... {

// polja tukaj

public Okno(int width, int height) {
    setTitle("Slikar");

    JPanel glavnaPlosca = new JPanel();
    glavnaPlosca.setLayout(new BoxLayout(glavnaPlosca,
                                         BoxLayout.Y_AXIS));
    this.add(glavnaPlosca);

    platno = new Platno(width, height);
    glavnaPlosca.add(platno);

    JPanel orodjarna = new JPanel();
    // dodamo elemente tukaj
    glavnaPlosca.add(orodjarna);

    JPanel datoteke = new JPanel();
    datoteke.setLayout(new FlowLayout(FlowLayout.TRAILING));
    // dodamo elemente tukaj
    glavnaPlosca.add(datoteke);
}
// metode tukaj
}
```

Polja razreda Okno

```
private Platno platno;  
  
private JButton zbrisiVse;  
private JSlider sirinaSlider;  
private JButton izberiBarvo;  
private JLabel aktivnaBarvaLabel;  
private JButton shrani;  
private JButton nalozi;  
  
private final JFileChooser FC = new JFileChooser ();
```

JPanel za orodja

```
JPanel orodjarna = new JPanel();

zbrisiVse = new JButton("Zbriši vse");
orodjarna.add(zbrisiVse);
zbrisiVse.addActionListener(this);

sirinaSlider = new JSlider(JSlider.HORIZONTAL, 1, 21, 11);
sirinaSlider.setMajorTickSpacing(5);
sirinaSlider.setPaintTicks(true);
orodjarna.add(sirinaSlider);
sirinaSlider.addChangeListener(this);

Color aktivnaBarva = Color.BLACK;
aktivnaBarvaLabel = new JLabel("      ");
aktivnaBarvaLabel.setOpaque(true);
aktivnaBarvaLabel.setBackground(aktivnaBarva);
platno.setBarva(aktivnaBarva);
izberiBarvo = new JButton("Izberi barvo");
izberiBarvo.addActionListener(this);
orodjarna.add(aktivnaBarvaLabel);
orodjarna.add(izberiBarvo);

glavnaPlosca.add(orodjarna);
```

JPanel za datoteke

```
JPanel datoteke = new JPanel();
datoteke.setLayout(new FlowLayout(FlowLayout.TRAILING));

shrani = new JButton("Shrani");
datoteke.add(shrani);
shrani.addActionListener(this);

nalozi = new JButton("Naloži");
datoteke.add(nalozi);
nalozi.addActionListener(this);

glavnaPlosca.add(datoteke);
```


Vmesnik ActionListener:

void actionPerformed(ActionEvent e) *Invoked when an action occurs.*

Vmesnik ChangeListener:

void stateChanged(ChangeEvent e) *Invoked when its target has changed state.*

```
public class Okno extends JFrame implements
    ActionListener, ChangeListener {

    @Override
    public void stateChanged(ChangeEvent e) {
        platno.setSirina(sirinaSlider.getValue());
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == zbrisiVse) platno.zbrisiVse();
        else if (e.getSource() == izberiBarvo) {
            Color novaBarva = JColorChooser.showDialog(this,
                "Izberite barvo", platno.getBarva());
            if (novaBarva != null) {
                platno.setBarva(novaBarva);
                aktivnaBarvaLabel.setBackground(novaBarva);
            }
        }
    }
}
```

```

    } else if (e.getSource() == shrani) {
        int returnVal = FC.showSaveDialog(this);
        if (returnVal == JFileChooser.APPROVE_OPTION) {
            File file = FC.getSelectedFile();
            try (FileOutputStream fileOut = new FileOutputStream(file);
                 ObjectOutputStream objectOut =
                     new ObjectOutputStream (fileOut)); {
                objectOut.writeObject(platno.getPoteze());
            } catch (IOException exn) { exn.printStackTrace(); }
        }
    } else if (e.getSource() == nalozi) {
        int returnVal = FC.showOpenDialog(this);
        if (returnVal == JFileChooser.APPROVE_OPTION) {
            File file = FC.getSelectedFile();
            try (FileInputStream fileIn = new FileInputStream(file));
                 ObjectInputStream objectIn =
                     new ObjectInputStream (fileIn)); {
                LinkedList<Poteza> poteze =
                    (LinkedList<Poteza>) objectIn.readObject();
                platno.setPoteze(poteze);
            } catch (Exception exn) { exn.printStackTrace();}
        }
    }
}
}
}

```

Vsi uvozi!

```
import java.awt.Color;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.FlowLayout;

import java.util.LinkedList;
import java.io.File;
import java.io.FileOutputStream;
import java.io.ObjectOutputStream;
import java.io.ObjectInputStream;
import java.io.InputStream;
import java.io.ObjectInputStream;
import java.io.IOException;

import javax.swing.BoxLayout;
import javax.swing.JButton;
import javax.swing.JColorChooser;
import javax.swing.JFileChooser;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JSlider;
import javax.swing.event.*;
```

Platno stran 1

```
public class Platno extends JPanel
    implements MouseListener, MouseMotionListener {

    private LinkedList<Poteza> poteze;
    private Poteza aktivnaPoteza;

    private Color aktivnaBarva;
    private int aktivnaSirina;

    public Platno(int sirina, int visina) {
        this.setPreferredSize(new Dimension(sirina,visina));
        this.setBackground(Color.WHITE);
        aktivnaBarva = Color.BLACK;
        aktivnaSirina = 11;
        this.addMouseListener(this);
        this.addMouseMotionListener(this);
        this.poteze = new LinkedList<Poteza> ();
    }
```

Platno stran 2

```
public LinkedList<Poteza> getPoteze () {  
    return poteze;  
}  
  
public void setPoteze (LinkedList<Poteza> poteze) {  
    this.poteze = poteze;  
    repaint();  
}  
  
public void setBarva (Color barva) {  
    this.aktivnaBarva = barva;  
}  
  
public Color getBarva () {  
    return aktivnaBarva;  
}  
  
public void setSirina (int sirina) {  
    this.aktivnaSirina = sirina;  
}  
  
public void zbrisiVse () {  
    poteze = new LinkedList<Poteza>();  
    repaint();  
}
```

Platno stran 3

```
@Override
protected void paintComponent(Graphics g) {
    super.paintComponent(g);
    Graphics2D g2 = (Graphics2D)g;
    for (Poteza p : poteze) {
        p.narisi(g2);
    }
    if (aktivnaPoteza != null) {
        aktivnaPoteza.narisi(g2);
    }
}
```

```
@Override
public void mouseClicked(MouseEvent e) { }
```

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

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

```
@Override
public void mouseMoved(MouseEvent e) { }
```

Platno stran 4

```
@Override
public void mousePressed(MouseEvent e) {
    aktivnaPoteza = new Poteza(aktivnaBarva, aktivnaSirina,
                                e.getX(), e.getY());

    this.repaint();
}
```

```
@Override
public void mouseReleased(MouseEvent e) {
    aktivnaPoteza.dodajTocko(e.getX(), e.getY());
    poteze.addLast(aktivnaPoteza);
    aktivnaPoteza = null;
    this.repaint();
}
```

```
@Override
public void mouseDragged(MouseEvent e) {
    aktivnaPoteza.dodajTocko(e.getX(), e.getY());
    this.repaint();
}
```

```
}
```

```
public class Poteza implements Serializable {

    static final long serialVersionUID = 1L;

    private Color barva;
    private int sirina;
    private GeneralPath pot;

    public Poteza(Color barva, int sirina, int x, int y) {
        this.barva = barva;
        this.sirina = sirina;
        this.pot = new GeneralPath();
        pot.moveTo(x, y);
    }

    public void dodajTocko(int x, int y) {
        pot.lineTo(x, y);
    }

    public void narisi(Graphics2D g) {
        g.setColor(barva);
        g.setStroke(new BasicStroke(sirina, BasicStroke.CAP_ROUND,
                                     BasicStroke.JOIN_ROUND));
        g.draw(pot);
    }
}
```



```
public class Slikar {  
  
    public static void main(String[] args) {  
        Okno glavnoOkno = new Okno(750, 750);  
        glavnoOkno.pack();  
        glavnoOkno.setVisible(true);  
    }  
}
```

Alternativna pot

Lahko lažje sestavimo grafične vmesnike z uporabo IDE, ki ima “GUI Builder”, na primer **NetBeans**.

V tem predmetu grafične vmesnike pišemo ročno, da bi bolj razumeli kako delajo.

Novica!

Nova verzija jezika:

Java SE 18

je izšla v torek 22. marca 2022!