



# Programiranje I – RIN Računalništvo I – MA

## Dogodkovno gnano programiranje

# Vsebina

- Kaj je to dogodkovno gnano programiranje?
- AWT
- Okvir – Frame
- Komponente
- Risanje
- Dogodki in poslušalci
- Ravnatelj izgleda
- Swing je še boljši!

# Grafični Uporabniški vmesnik

- Lepe slikice, ki se pogovarjajo z uporabnikom
- No, tudi kaj več.

# AWT

- AWT pomeni Abstract Window Toolkit.
- AWT je del Java Foundation Classes (JFC)
- je standardna zbirka klicev (API) za izdelavo grafičnega uporabniškega vmesnika (GUI)
- AWT je v resnici samo zbirka ovitkov (wrapper) okrog standardnih sistemski klicev za delo z okni

# AWT - lastnosti

- AWT je hiter (kliče systemske klice).
- Izgled AWT je odvisen od operacijskega sistema.
- Nimamo popolnega nadzora nad obnašanjem komponent (klici zunanjih metod).

# Okvir – Frame

- Najenostavnejši razred za delo z okni.
- Omogoča izdelavo okenskega programa v nekaj vrsticah.

# Komponente

- Gumb – Button
- Napis – Label
- Vnosno polje – TextField
- Vnosno območje – TextArea
- Kombo – ComboBox
- ...

# Primer–moje prvo javansko okno

```
import java.awt.*;

public class MojePrvoJavanskoOkno extends Frame{
    public MojePrvoJavanskoOkno (int x, int y){
        setSize(x,y);
        show();
    } //MojePrvoJavanskoOkno
    public static void main(String arg[]){
        MojePrvoJavanskoOkno oo = new MojePrvoJavanskoOkno(100, 400);
    }
} //MojePrvoJavanskoOkno
```



# Primer–moje prvo javansko okno

- Primer prevedemo: `javac MojePrvoJavanskoOkno.java`
- Primer izvedemo: `java MojePrvoJavanskoOkno`

# Okvir – ravnatelj izgleda

- Ravnatelji izgleda ali »po domače« **Layout Managers** določajo obnašanje komponent na kontejnerju.
- Poznamo več osnovnih ravnateljev, nove lahko pišemo sami.
- Primeri:
  - **BorderLayout** (osnovni izgled, komponente so razvrščene na vse strani neba ter sredino)
  - **GridLayout** (kontejner je razdeljen na enake dele – mreža. Vsak del dobi svojo komponento)
  - **GridbagLayout** (najzahtevneši izgled, s pomočjo sider ter uteži se natančno določi položaj vsake komponente)
  - **FlowLayout** (komponente so nizane prosto od leve proti desni)

# Primer – okno s komponentami

```
import java.awt.*;
public class NovoOkno extends Frame{
    public NovoOkno(int x, int y){
        setSize(x,y);
        setLayout(new BorderLayout());
        Panel pp = new Panel ();
        pp.setLayout(new GridLayout(3, 4));
        pp.add(new Button("panel5"));
        pp.add(new Button("panel4"));
        pp.add(new Button("panel3"));
        pp.add(new Button("panel2"));
        pp.add(new Button("panel1"));
        add(new Button("Okay"), BorderLayout.NORTH);
        add(new Button("Okay"), BorderLayout.SOUTH);
        add(pp, BorderLayout.CENTER);
        add(new Button("Okay"), BorderLayout.EAST);
        add(new Button("Okay"), BorderLayout.WEST);
        show();
    } //NovoOkno
} //NovoOkno
predavanje
```

# Primer - okno s komponentami

- Primer prevedemo: javac Okno1.java
- Primer izvedemo: java Okno1

# Dogodki in poslušalci

- Dogodek v AWT je odziv na uporabnikovo akcijo oziroma sprememba stanja določene komponente.
- Dogodki so: klik miške, vnos v določeno polje, prenos komponente...
- Dogodek se prenese iz izvornega objekta na t.i. poslušalca (Listener).
- Izvorni objekt je objekt, s katerim je uporabnik interaktiral, poslušalec pa je objekt, ki je registriran, da posluša in prestreže določen tip dogodka.
- Vsaki komponenti lahko za vsak tip dogodka dodamo poslušalca, ki se bo izvedel ob nastopu dogodka.

# Dogodki in poslušalci, naprej

- Poznamo več vrst dogodkov, pomembnejši so:
  - ActionEvent se zgodi, ko npr. kliknemo na gumb uporabniškega vmesnika, izberemo opcijo v menuju,...
  - AdjustmentEvent se zgodi, ko premaknemo drsnik (Scrollbar);
  - FocusEvent se zgodi, ko komponenta dobi ali izgubi vnosni fokus;
  - KeyEvent se zgodi, ko uporabnik pritisne na tipko;
  - MouseEvent se zgodi, ko uporabnik pritisne na mišji gumb, ali pa miško premakne;
  - WindowEvent se zgodi, ko okno odpremo, minimiziramo, zapremo,...
  - ItemEvent se zgodi, ko npr. izberemo opcijo v seznamu, pritisnemo na gumb tipa CheckBox,...
  - TextEvent se zgodi, ko v TextArea ali TextField komponentah spremeni tekst

# Dogodki in poslušalci, poslušalec

- Poslušalec je razred, ki ga izvedemo iz primerne adapter razreda (Adapter) ali naredimo z implementacijo primerne Listener vmesnika.
- Tako lahko za poslušanje dogodkom MouseEvent naredimo razred, ki je izpeljan iz razreda MouseAdapter ali razred, ki implementira vmesnik MouseListener.
- MouseAdapter in MouseListener imata pet funkcij (mouseClicked, mouseEntered, mouseExited, mousePressed, mouseReleased), ki jih lahko prekrijemo, odvisno od tega, kakšen tip dogodkov želimo spremljati.

# Primer – dogodki

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

public class Poslusa extends Frame{
    public Poslusa () {
        Label lbl;
        Button btn;
        Poslusalec psl;
        psl = new Poslusalec();
        lbl = new Label("Ob pritisku na gumb se bo odprlo novo okno!");
        add(lbl);
        btn = new Button("Novo okno");
        btn.addActionListener(psl);
        add(btn);
        System.out.println("dsdaf");
    } //Poslusa
} //Poslusa

class Poslusalec implements ActionListener {
    public void actionPerformed(ActionEvent e) {
        System.out.println("AKCIJA!!!");
    } //actionPerformed
} //Poslusalec
predavanje
```



# Primer - dogodki

- Primer prevedemo: `javac Start.java`
- Primer izvedemo: `java Start`

# Primer – risanje

```
import java.awt.*;
public class Avto extends Panel{
    boolean prizgana = false;
    public void luc(boolean p){
        prizgana = p;
        repaint();
    }
    public void paint (Graphics g){
        g.drawLine(400,300,300,250);
        g.setColor(Color.red);
        g.fillRect(230,300,200,100);
        g.fillRect(200,400,400,100);
        g.setColor(Color.white);
        ...
        if(prizgana){
            g.setColor(Color.yellow);
            g.drawLine(600, 450, 620, 430);
            g.drawLine(600, 450, 620, 470);
            ...
        }
    } //paint
} // Avto
```

# Primer - risanje

- Primer prevedemo: `javac Test.java`
- Primer izvedemo: `java Test`

# Primer – avto in garaža in še kaj

```
..
public class Garaza extends Frame{
    Avto avto;
    boolean prizgana = false;
    //poslusalci ("", "1")...
    public Garaza(int x, int y){
        avto = new Avto();
        Button b1 = new Button("skrije zaslon");
        Button b2 = new Button("Akcija");
        Poslusalec ps11 = new Poslusalec();
        Poslusalec1 ps12 = new Poslusalec1();
        b1.addActionListener(ps11);
        b2.addActionListener(ps12);
        setSize(x,y);
        setLayout(new BorderLayout());
        add(b1, BorderLayout.NORTH);
        add(b2, BorderLayout.SOUTH);
        add(avto, BorderLayout.CENTER);
        show();
    } //Garaza
} //Garaza
```

# Primer – avto in garaža in še kaj

- Primer prevedemo: `javac OknoGaraze.java`
- Primer izvedemo: `java OknoGaraze`

# Swing je še boljši!

- **Swing si bomo podrobneje ogledali v naslednji učni enoti.**
- **Nastal je kot posledica negodovanja programerjev na okorno uporabo AWT.**
- **Swing je zelo počasen!**

# Swing - pregled

- Zakaj, kako in kdo
- Uporaba
- Primer uporabe
- Razlike med Swing in AWT
- Trije priloženi izgledi

# Swing - zakaj, kako in kdo

- Sun in zavezniki
  - (Sun je umrl šmrc)
    - Oracle in zavezniki
- JFC
  - AWT
  - Java 2D
  - Accessibility
  - Drag and Drop
  - Swing



# Swing Uporaba

- Analogna uporabi AWT komponent
- Dodamo črko J
- Dodatne komponente

# Swing - primer uporabe

```
tree = new JTree();  
list = new JList(items);  
button = new JButton("Pritisni");  
button.addActionListener(new  
ButtonListener());
```

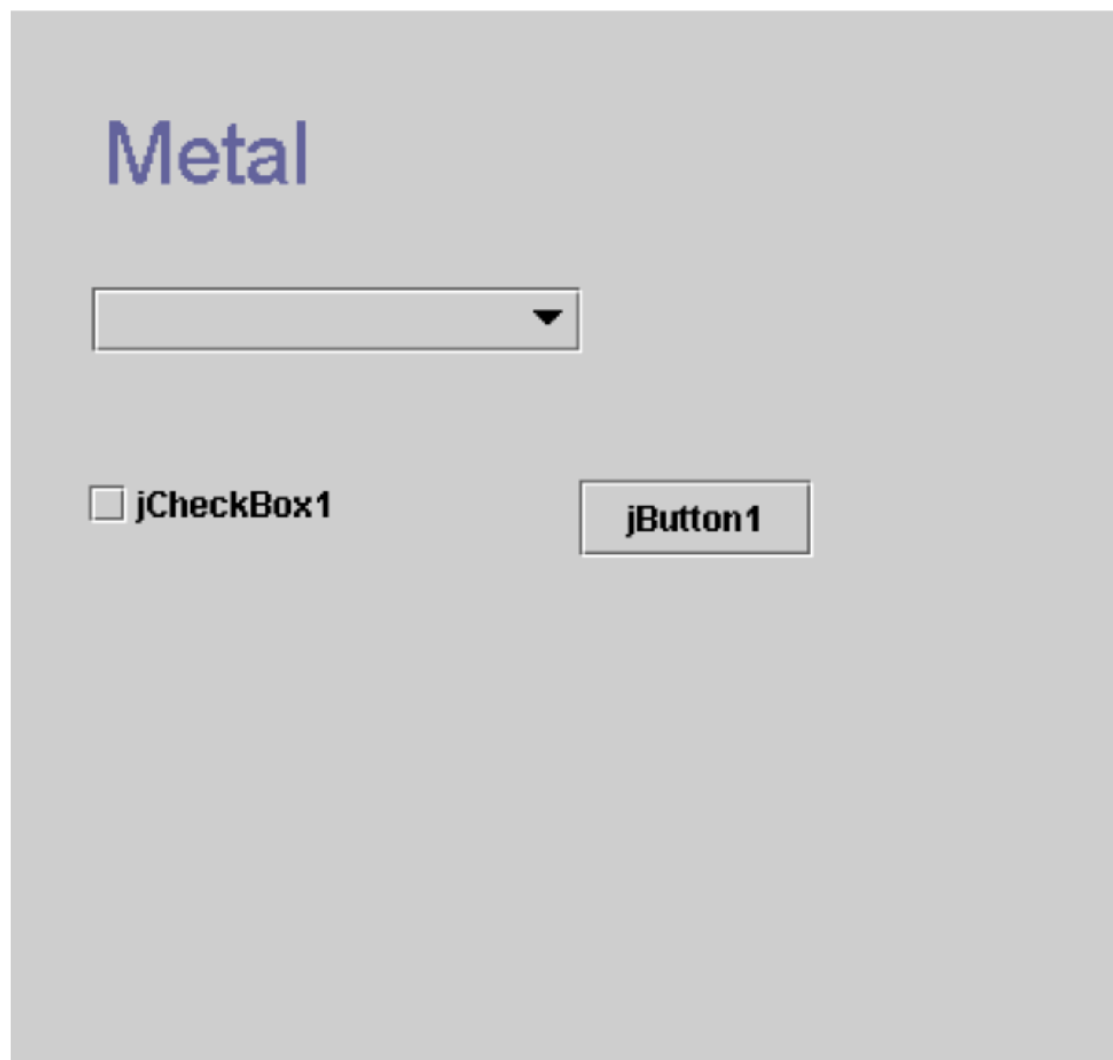
# Swing - Razlike med Swing in AWT

- Vse komponente so napisane v Javi
  - AWT komponente so samo ovojnica
- Enoten izgled na vseh platformah
  - Glej gor
- Večji nadzor nad komponentami
  - Pluggable Look and Feel
- Počasna in požrešna

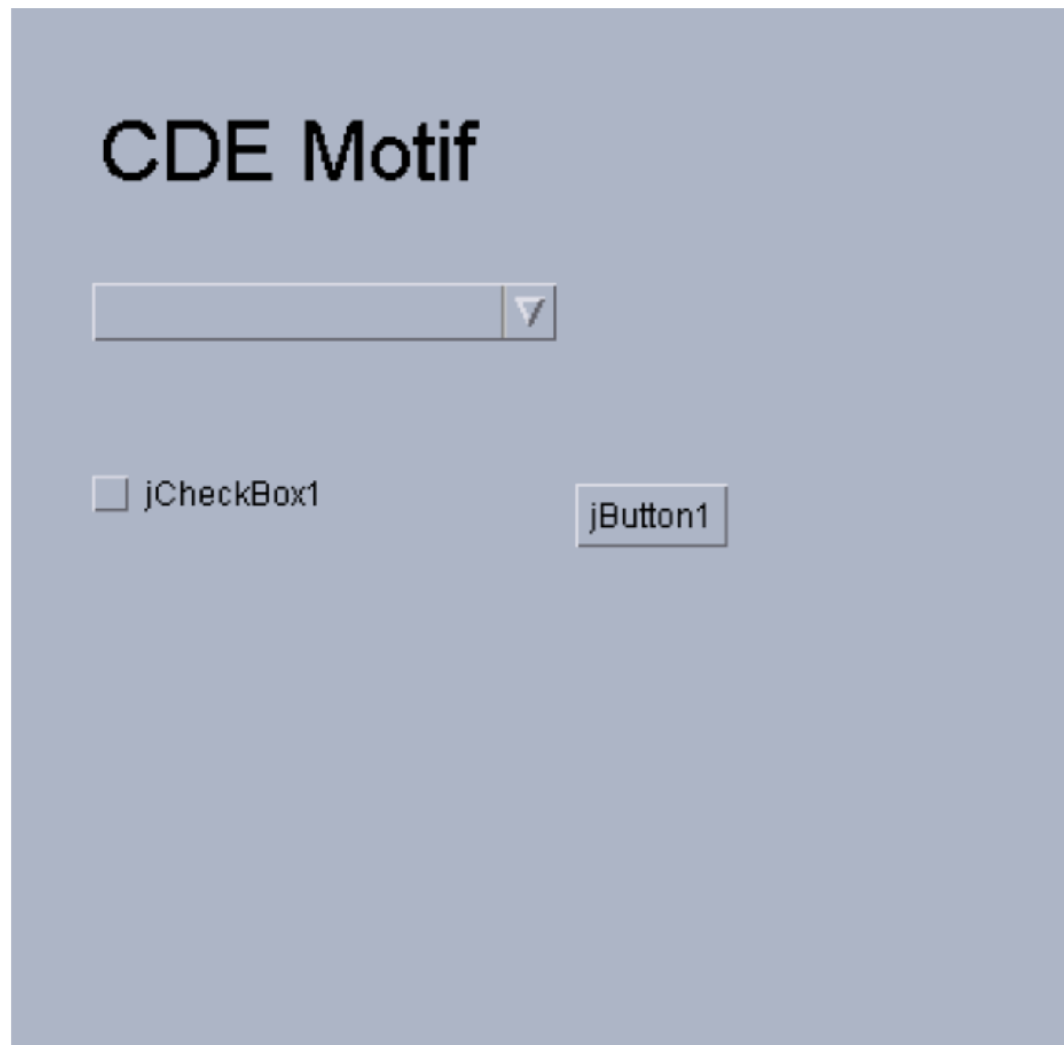
# Swing - Windows izgled



# Swing metal (Java) izgled



# Swing Motif izgled



# Swing - naloga

- Oglejmo si kodo: `SwingApplet.html`