

# Objektno Orijentisano Programiranje 1

Programiranje korisničkog interfejsa u Javi

# Literatura

- B. Milosavljević. M. Vidaković: Java i Internet programiranje
- Cay S. Horstmann, Gary Cornell: Core Java™ 2 Volume I - Fundamentals, Seventh Edition
- Cay S. Horstmann, Gary Cornell: Core Java™ 2 Volume II - Advanced Features, Seventh Edition
- Jonathan Knudsen, Java 2D Graphics
- Swing Tutorial: <http://java.sun.com/docs/books/tutorial/>
- JDK™ 5.0 Documentation:  
<http://java.sun.com/j2se/1.5.0/docs/>

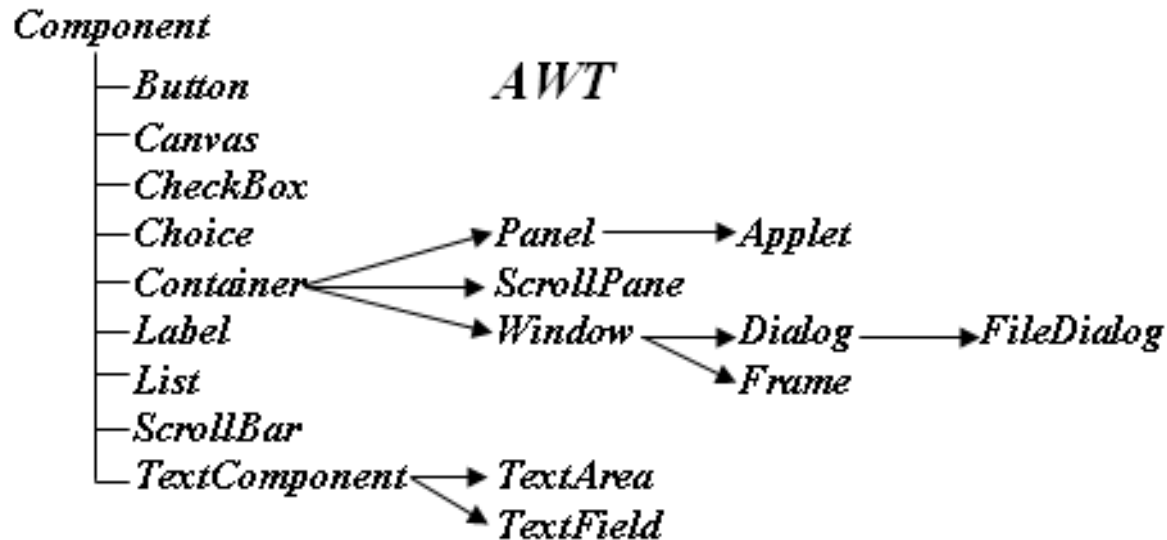
# Uvod

- Korisnički interfejs je upravlján događajima (*Event Driven*).
- U verzijama **1.0** i **1.1** standard je AWT biblioteka (*Abstract Window Toolkit*).
- Od verzije 1.2 standard je Swing biblioteka.

# Abstract Window Toolkit

- AWT: biblioteka koja obezbeđuje upotrebu minimalnog skupa komponenti grafičkog interfejsa, a kojeg poseduju sve platforme koje podržavaju Javu
- izgleda “podjednako osrednje” na svim platformama
- paketi:
  - java.awt
  - java.awt.event
  - java.awt.image
  - java.awt.datatransfer

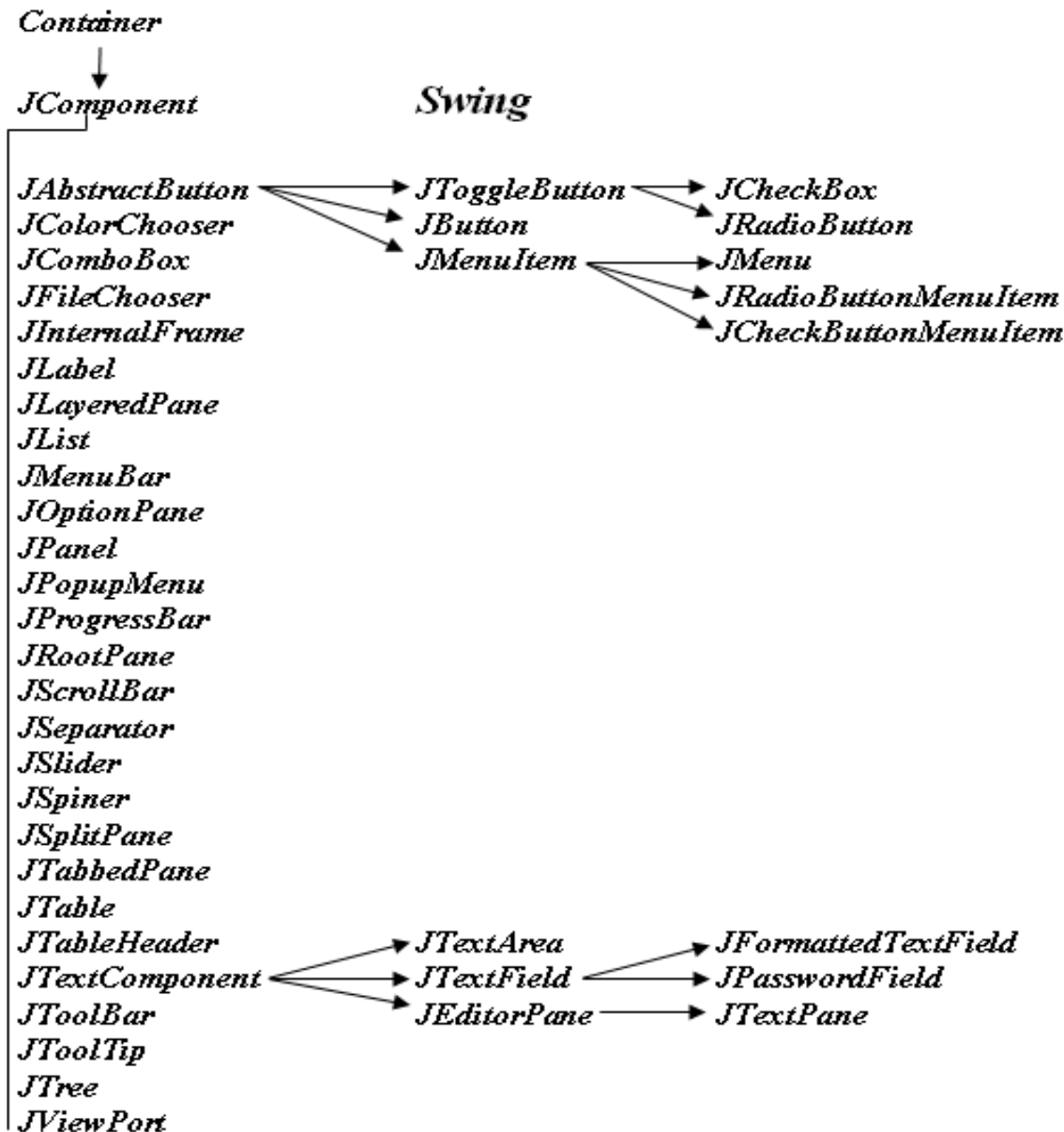
# Delimična hijerarhija AWT klasa



# Swing

- napisan “od nule”, u Javi
- ne oslanja se na operativni sistem
- jednako izgleda i radi na svim platformama
- pluggable look-and-feel (Java, Windows, GTK, MacOS, ...)
- veliki broj komponenti
- drag&drop, rad sa clipboard-om
- kompletna podrška za Unicode standard
- paketi javax.swing.\*
- nazivi klasa počinju sa Jxxx

# Delimična hijerarhija Swing



# *Event Driven* model

- Program se ne izvršava linearno (od gore prema dole).
- Pišu se procedure koje se izvršavaju po pojavi nekog događaja korisničkog interfejsa (klik mišem, pritisak tastera i sl.).
- Program ima inicijalizacioni blok i blokove koda koji reaguju na događaje korisničkog interfejsa.



# *Event Driven* model

- Svaka akcija nad komponentama korisničkog interfejsa izaziva generisanje objekata klasa naslednica *Event* klase.
- Ti objekti se prosleđuju objektima klasa naslednica *EventListener* klase, koje “osluškuju” događaje.

# Osnovna struktura GUI aplikacije

- Izvršavanje programa počinje `main()` funkcijom.
- Najčešće se u `main()` funkciji otvara prvi prozor (**JFrame**).

# Razlika u konstrukciji GUI-ja za Windows i Java aplikacije

- Windows aplikacije koriste tzv. *resurse* za definisanje izgleda korisničkog interfejsa
- Java aplikacije svoj korisnički interfejs grade programski, najčešće u toku inicijalizacije prozora

# Dodavanje komponenti u prozor

- prilikom konstrukcije objekta klase naslednice `JFrame` dodaju se komponente na formu
- ako se komponenta samo kreira (`new JButton()`), to nije dovoljno
- komponenta se mora dodati na formu metodom:  
`getContentPane().add(new JButton("asdf"))`

# Kontejneri najvišeg nivoa (top-level) u Swing-u

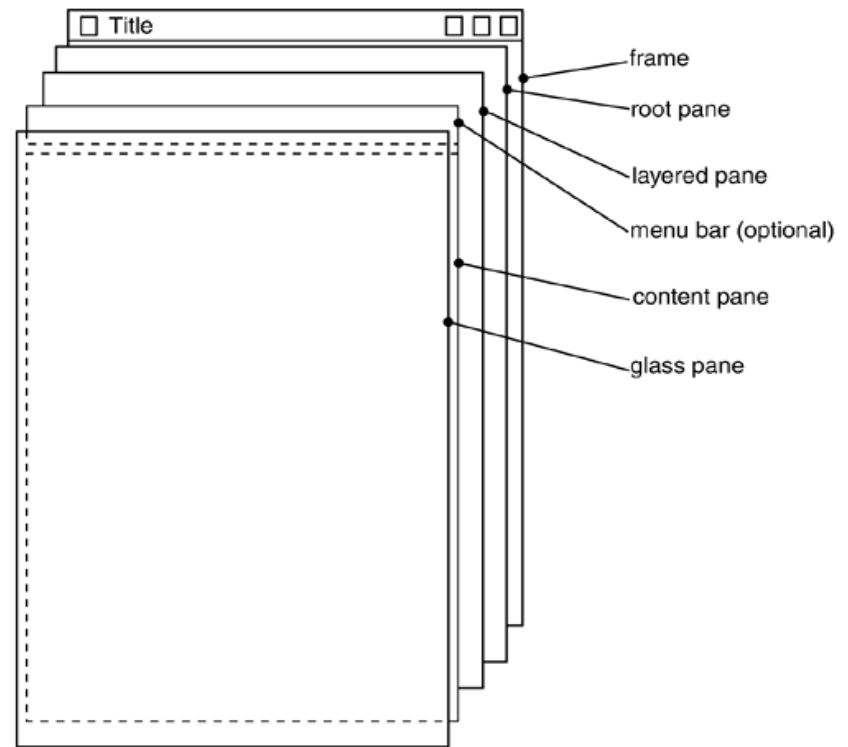
- JFrame - koristi se za implementaciju glavnog prozora aplikacije. Može imati menije, toolbarove itd.
- JDialog - Nasleđuje se iz java.awt.Dialog i koristi se za kreiranje dijaloga aplikacije. Za kreiranje standardnih dijaloga koristi se klasa JOptionPane.
- JWindow - prozor koji za razliku od JFrame-a nema naslov (titlebar) i sistemske akcije za manipulaciju prozorima
- JApplet - klasa za kreiranje apleta u swingu

# Paneli u okviru kontejnera

- Svi kontejneri najvišeg nivoa implementiraju interfejs *RootPaneContainer* koji deklariše metode za manipulaciju sa panelima. Kontejneri najvišeg nivoa poseduju sledeće panele:
  - rootPane (JRootPane) – jedino se on nalazi u kontejneru najvišeg nivoa i sadrži preostale panele. Definiše metode za manipulaciju ostalim panelima
  - layeredPane (JLayeredPane) - sadrži u sebi meni i contentPane i služi za njihov prostorni raspored (layout). Raspoređuje komponente u nivoe (layers) čiji Z-order je predefinisani. Služi za organizaciju preklapanja komponenti, iscrtavanje dijaloga, popupa, tooltipova itd
  - contentPane (Container) - po defaultu JPanel. Predstavlja sadržaj prozora bez menija i title-bara
  - glassPane (Component) - po defaultu je JPanel. Providan je i nevidljiv i nalazi se preko celog rootPane. Koristi se u situacijama kada je potrebno uhvatiti događaje miša ili tastature ili prikazati nešto ispred svih ostalih komponenti

# Dodavanje komponenti u kontejner

- prilikom konstrukcije prozora komponente se dodaju u `ContentPane`.
- Podrazumevani layout manager za `ContentPane` je `BorderLayout`.



# JFrame

- JFrame je osnovna komponenta korisničkog interfejsa.
- Reprezentuje prozor.
- Metoda `setVisible(boolean mode)` postavlja prozor ili ga sakriva.
- Metoda `setSize(širina, visina)` podešava veličinu prozora.
- Metoda `setTitle("tekst")` postavlja naslov prozora.
- Tri načina gašenja aplikacije kada se ugasi prozor:
  - `processWindowEvent(WindowEvent)` – JFrame1
  - `WindowListener` – JFrame2
  - `setDefaultCloseOperation(int operation)` – JFrame3

primer3 JFrame1

primer4 JFrame2

primer5 JFrame3



# Rad sa prozorima – korisne metode 1/5

- Nasleđeno od klase `java.awt.Component`:

- **`boolean isVisible()`**

Vraća `true` ili `false` u zavisnosti da li je komponenta vidljiva ili ne, respektivno

- **`void setVisible(boolean b)`**

Prikazuje ili sakriva komponentu, u zavisnosti o vrednosti parametra `b`

- **`boolean isEnabled()`**

Proverava da li je komponenta omogućena za unos (tj. da li može da primi fokus)

- **`void setEnabled(boolean b)`**

Omogućava ili onemogućava komponentu za unos, u zavisnosti od vrednosti parametra `b`

- **`Point getLocation()`**

Vraća poziciju gornjeg levog ugla komponente relativno u odnosu na gornji levi ugao “vlasnika”

- **`Point getLocationOnScreen()`**

Vraća poziciju gornjeg levog ugla komponente u odnosu na koordinate ekrana

# Rad sa prozorima – korisne metode 2/5

- **void setBounds(int x, int y, int width, int height)**

Postavlja komponentu na poziciju (x, y) realativno u odnosu na vlasnika i definiše joj dimenzije

- **void setLocation(int x, int y)**

Postavlja komponentu na zadate koordinate

- **void setLocation(Point p)**

Postavlja komponentu na zadatu tačku

- **Dimension getSize()**

Vraća dimenzije komponente (**Dimension** poseduje attribute **width** i **height**)

- **void setSize(int width, int height)**

Postavlja dimenzije komponente zadavanjem širine i visine

- **void setSize(Dimension d)**

Postavlja dimenzije komponente korišćenjem **Dimension** parametra

# Rad sa prozorima – korisne metode 3/5

- Nasleđeno od klase `java.awt.Window`

- **`void toFront()`**

Premešta prozor “u prvi plan” (ispred svih ostalih prozora)

- **`void toBack()`**

Sklanja prozor iza ostalih prozora

# Rad sa prozorima – korisne metode 4/5

- Nasleđeno od klase `java.awt.Frame`

- **`void setResizable(boolean b)`**

Definiše da li korisnik može menjati dimenzije prozora, u zavisnosti od vrednosti parametra `b`

- **`void setTitle(String s)`**

Definiše naslov prozora

- **`void setIconImage(Image image)`**

Dodeljuje ikonu prozoru (ikona mora biti prethodno konstruisana od strane klase `Toolkit`, videti u nastavku)

- **`void setUndecorated(boolean b)`**

Sklanja i vraća “dekoraciju” prozora (dugmad za minimizovanje i maksimizovanje) u zavisnosti od vrednosti parametra

- **`boolean isUndecorated()`**

vraća `true` ukoliko prozor nema dekoraciju

# Rad sa prozorima – korisne metode 5/5

- **`int getExtendedState()`**

Vraća stanje prozora

- **`void setExtendedState(int state)`**

Definiše stanje prozora pri pokretanju. Moguće vrednosti su:

- **`Frame.NORMAL`**
- **`Frame.ICONIFIED`**
- **`Frame.MAXIMIZED_HORIZ`**
- **`Frame.MAXIMIZED_VERT`**
- **`Frame.MAXIMIZED_BOTH`**

# Pomoćna klasa `java.awt.Toolkit`

- Dobija se pomoću statičke metode:

**`static Toolkit getDefaultToolkit()`**

vraća objekat sa tekućim podešavanjima grafičkog okruženja

- **`Dimension getScreenSize()`**

vraća tekuće dimenzije ekrana

- **`Image getImage(String filename)`**

Učitava sliku iz zadate datoteke. Širina i visina slike (`width` i `height`) su `(-1, -1)` ukoliko nije uspelo učitavanje

# Primer – rad sa klasom Toolkit

```
class MainFrame extends JFrame {  
    public MainFrame() {  
        // Preuzimamo dimenzije ekrana  
        Toolkit kit = Toolkit.getDefaultToolkit();  
        Dimension screenSize = kit.getScreenSize();  
        int screenHeight = screenSize.height;  
        int screenWidth = screenSize.width;  
  
        // Podesavamo dimenzije prozora na polovinu dimenzija ekrana  
        setSize(screenWidth / 2, screenHeight / 2);  
  
        // Dodeljujemo ikonu  
        Image img = kit.getImage("icon.gif");  
        setIconImage(img);  
        //Podesavamo naslov  
        setTitle("My Second GUI App");  
    }  
}
```

# Komponente

<b>JButton</b>	<b>Dugme</b>
<b>TextField</b>	<b>Jednolinijsko Edit polje</b>
<b>TextArea</b>	<b>Višelinijsko Edit polje (Memo)</b>
<b>Label</b>	<b>Labela</b>
<b>CheckBox</b>	<b>CheckBox</b>
<b>ButtonGroup</b>	<b>Kontejner za Radio Button-e</b>
<b>JRadioButton</b>	<b>Radio Button</b>
<b>JComboBox</b>	<b>ComboBox</b>
<b>JList</b>	<b>ListBox</b>
<b>JTabbedPane</b>	<b>Kartice</b>
<b>JOptionPane</b>	<b>MessageBox</b>
<b>JMenu, JMenuItem</b>	<b>Meniji</b>
<b>JDialog</b>	<b>DialogBox</b>
<b>JFrame</b>	<b>Prozor</b>