
GUI (Graphical User Interface)

Bojan Tomić

Java i GUI

- Java biblioteke sa gotovim grafičkim komponentama
 - AWT (ne koristi se više)
 - SWING (aktuelna) - tutorijeli na Oracle sajtu
- Paket “javax.swing”
- Svaka grafička komponenta je implementirana u vidu klase
- Svaka grafička komponenta je JavaBean
 - get, set i is metode za svaki atribut (property)

Eclipse i GUI

- **Window Builder**
- Eclipse plug-in
- Omogućava grafičko kreiranje prozora umesto ručnog pisanja koda
- **Visual Editor (NAPUŠTEN)**
- Eclipse plug-in
- Omogućava istu stvar

NetBeans i GUI

- **Matisse**
- NetBeans GUI builder
- Omogućava grafičko kreiranje prozora umesto ručnog pisanja koda
- U odnosu na Eclipse WB:
 - Pouzdaniji (za velike, složene projekte)
 - Sporiji
 - Stvara kod koji je složeniji i teži za ručnu izmenu

JFrame

- Klasa koja predstavlja prozor
- Content Pane
 - Centralni deo prozora
 - Na njega se dodaju ostale komponente (dugmići, polja za unos...)
- Neki od atributa (property) klase JFrame su



JFrame

- bounds (x,y,w,h)
 - x i y su koordinate gornjeg levog ćoška prozora
 - w i h su širina (width) i visina (height) prozora
- defaultCloseOperation
 - EXIT - kada se zatvori prozor ceo program se prekida
 - DISPOSE - kada se zatvori prozor, briše se iz memorije
 - HIDE - kada se zatvori prozor, on se više ne vidi ali je i dalje aktivan
 - DO_NOTHING - ništa se ne dešava

JFrame

- enabled
 - true - prozor je aktivan i reaguje na događaje
 - false - prozor nije aktivan i ne reaguje na događaje
- iconImage
 - služi za postavljanje određene ikone (icon) u gornji levi ugao prozora pre naslova
- location (x,y)
 - x i y su koordinate gornjeg levog ćoška prozora

JFrame

- **resizable**
 - true - kada je aktivan, prozor može da se povećava i smanjuje
 - false - prozor ne može da se povećava niti smanjuje
- **size (width, height)**
 - width - širina prozora u pikselima (po X osi)
 - height - visina prozora u pikselima (po Y osi)
- **title**
 - naslov koji se prikazuje na vrhu prozora

JFrame

- visible
 - true - kada se inicijalizuje, prozor je vidljiv
 - false - kada se inicijalizuje, prozor nije vidljiv
- Ako je potrebno postaviti da se prozor pojavljuje na sredini ekrana koristiti poziv metode klase JFrame:

```
setLocationRelativeTo (null) ;
```
- Ovo važi i za sve dijaloge (videti dalje u tekstu)

JFrame

- Svaka ozbiljnija desktop aplikacija se sastoji iz više prozora
- Običaj je da se aplikacija startuje preko glavnog prozora (on mora da ima “main” metodu)
- Svi ostali prozori se kreiraju i aktiviraju po potrebi (najčešće ih kreira glavni prozor)
- Kreiranje i aktiviranje prozora (ovaj kod ne sme da se poziva van EventDispatch thread-a)

```
NazivKlase prozor = new NazivKlase();  
prozor.setVisible(true);
```

JFrame

- Content Pane je centralna površina svakog prozora
- Na nju se postavljaju sve ostale komponente prozora - dugmići, polja za unos...
- Content Pane ima opciju za podešavanje rasporeda komponenti (layout)
- Layout Manager-i su klase koje definišu kako će se komponente sa Content Pane-a rasporediti

Dijalozi (dialog)

- Dialog – mali, obično jednostavan, predefinisani prozor koji ima neku jasnu funkciju:
 - Prikaz neke kratke poruke (informacije, greške..)
 - Prozor za unos kratke odluke (da-ne, ok-cancel)
 - Pronalaženje fajla na disku
 - Izbor boje
- Predefinisane klase za dijaloge
 - JOptionPane
 - JFileChooser
 - JColorChooser

JOptionPane

- Klasa za kreiranje dijaloga za prikaz neke
 - Poruke
 - Upozorenja
 - Greške...
- Ima svoj skup ikonica koje se prikazuju na prozoru
- Dijalog se može prilagoditi u smislu izgleda
- Dijalog se kreira pozicijanjem statičke metode **showMessageDialog**

JOptionPane

- Primer dijaloga poruke

```
JOptionPane.showMessageDialog(frame,  
    "Neka poruka", "Naslov prozora")
```

- Primer dijaloga upozorenja

```
JOptionPane.showMessageDialog(frame,  
    "Tekst upozorenja", "Naslov prozora",  
    JOptionPane.WARNING_MESSAGE);
```

- Primer dijaloga greške

```
JOptionPane.showMessageDialog(frame,  
    "Tekst greške", "Naslov prozora",  
    JOptionPane.ERROR_MESSAGE);
```

JOptionPane

- Može i da prikaže dijalog za neku kratku odluku (npr. da-ne)
- Statička metoda
`showConfirmDialog`
- Primer

```
int opcija =  
JOptionPane.showConfirmDialog(frame,  
    "Pitanje", "Naslov prozora",  
    JOptionPane.YES_NO_OPTION);
```

JFileChooser

- Klasa za prikaz dijaloga za pronalaženje fajla ili lokacije na disku
- Dijalozi za otvaranje i čuvanje fajla
- **NE RADI** otvaranje ili čuvanje fajla već samo služi za navigaciju do lokacije
- Metode (nisu statičke) za prikaz dijaloga:
 - `showOpenDialog`
 - `showSaveDialog`

JFileChooser

- Primer dijaloga za otvaranje fajla (open)

```
JFileChooser fc = new JFileChooser();  
int returnVal = fc.showOpenDialog(frame);  
  
if (returnVal == JFileChooser.APPROVE_OPTION)  
    File file = fc.getSelectedFile();
```

- Primer dijaloga za čuvanje fajla (close)

```
JFileChooser fc = new JFileChooser();  
int returnVal = fc.showCloseDialog(frame);  
  
if (returnVal == JFileChooser.APPROVE_OPTION)  
    File file = fc.getSelectedFile();
```

Navigacija prozora i kontrola izvršavanja programa

- Skoro svaka GUI aplikacija se sastoji iz više prozora. Najčešće su to:
- Glavni prozor
 - Njime se pokreće aplikacija (ima main metodu)
 - Pokreće ostale (pomoćne) prozore
 - Kad se zatvori, prekida se aplikacija
- Pomoćni prozori
 - Nemaju main metodu
 - Kada se zatvore, samo se obrišu a aplikacija nastavlja sa radom

Navigacija prozora i kontrola izvršavanja programa

- Pokretanje i prikazivanje novog prozora

```
new NazivKlase().setVisible(true);
```

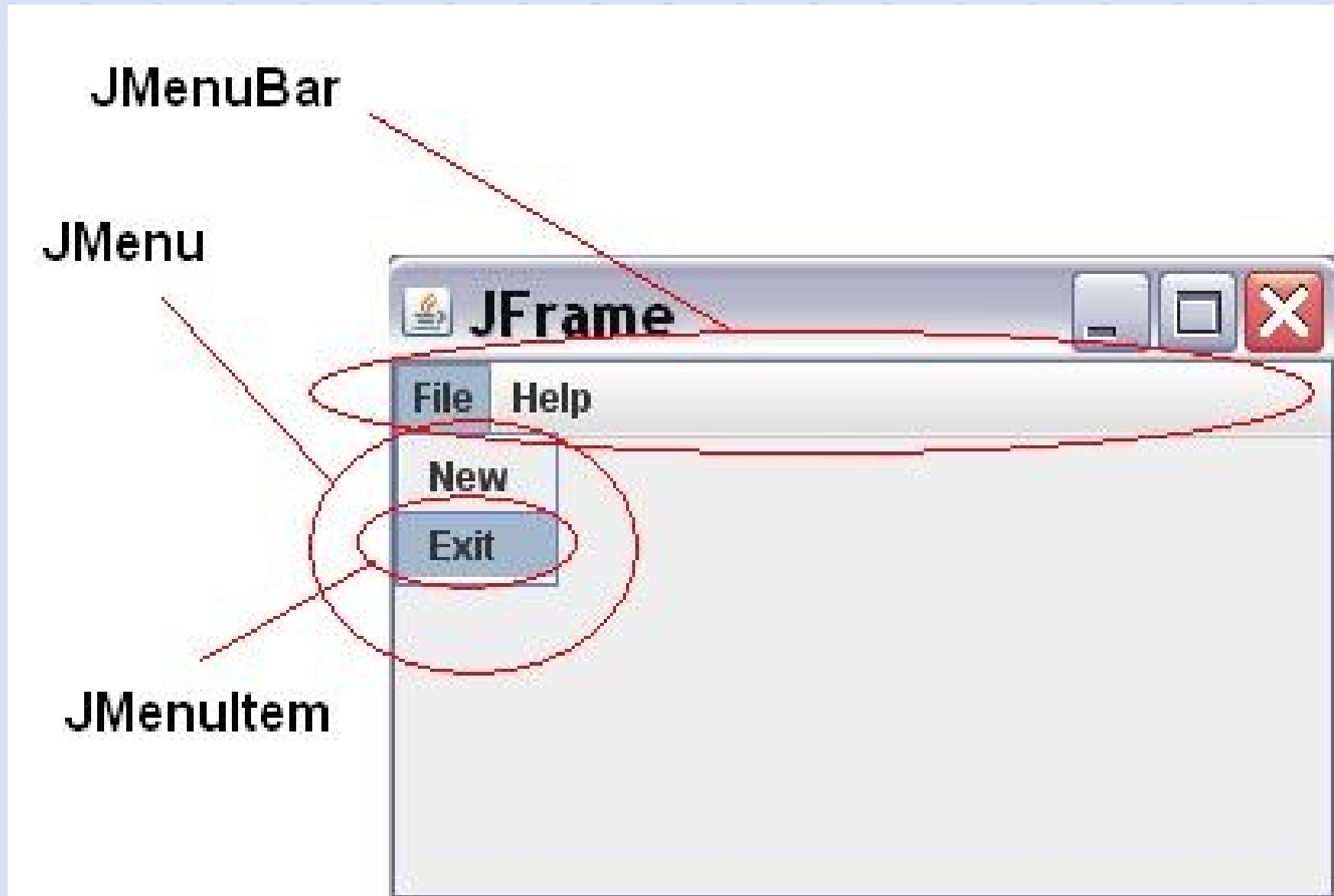
- Komanda za prekidanje rada celokupne aplikacije

```
System.exit(0);
```

- Komanda za zatvaranje aktivnog prozora (aplikacija nastavlja da radi ako to nije jedini aktivni prozor)

```
dispose();
```

Glavni meni (JMenuBar, JMenu, JMenuItem)



JMenuBar

- Klasa koja predstavlja pravougaonik pri vrhu prozora u kome se nalazi glavni meni
- Objekat ove klase pripada direktno prozoru i ne stavlja se na ContentPane
- Prozor može da ima samo jedan JMenuBar
- Na JMenuBar se dodaju meniji (JMenu)

JMenu

- Ova klasa predstavlja jedan padajući meni u okviru glavnog menija
- JMenuBar može da ima više JMenu komponenti (npr. “File”, “Edit”, “View”...)
- Svaki JMenu može sadržati jednu ili više stavki (klasa JMenuItem)

JMenuItem

- Stavka u okviru padajućeg menija (npr. “New”, “Open”, “Exit”...)
- Sve tri komponente mogu da sadrže tekst i, po potrebi, ikonu

JPopupMenu

- Slično kao JMenu, samo što se pojavljuje kada se na neku komponentu klikne desnim tasterom miša.
- Može imati stavke JMenuItem, podstavke i sve što običam meni može da ima.

Prečice u menijima

- Mnemonics - aktiviraju se kad je meni vidljiv, pozivanje jednim slovom ili kombinacijom, podvučeno je to slovo iz teksta u okviru menija
- Accelerators – aktiviraju se čak i kad meni nije vidljiv, pozivanje nekom kombinacijom

