

Objektno orijentirano programiranje

15.0: Grafičko sučelje – Swing *Osnovne komponente*

Priprema za 15. temu predavanja

Creative Commons

You are free to

- **Share** — copy and redistribute the material in any medium or format
- **Adapt** — remix, transform, and build upon the material

under the following terms

- **Attribution** — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.
- **NonCommercial** — You may not use the material for commercial purposes.
- **ShareAlike** — If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.
- <https://creativecommons.org/licenses/by-nc-sa/4.0/>



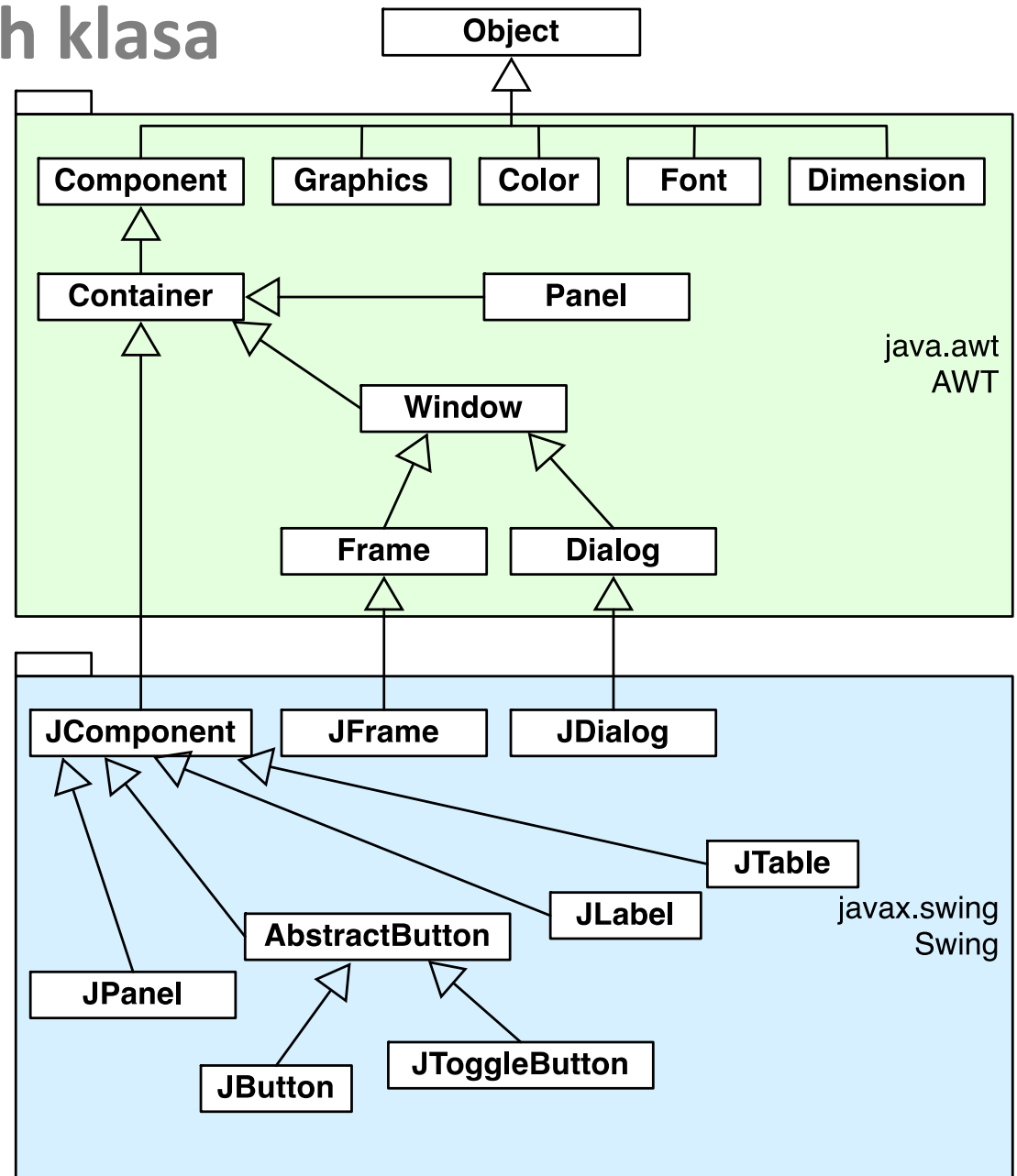
Sadržaj

- Izrada grafičkog sučelja značajno ovisi o tehnologiji koja se koristi, pa ćemo se zadržati na nekim osnovnim univerzalnim konceptima
 - pregled osnovnih komponente GUI-ja na primjeru Java Swinga
 - (tema ove pripreme)
 - upravljanje razmještajem grafičkih komponenti
 - model i obrada događaja
 - (tema predavanja 15.1)
 - obrada događaja koji imaju dulje trajanje i interakcija pozadinskih poslova s grafičkim sučeljem
 - (tema predavanja 15.2)
 - primjer korištenja izbornika i definiranje akcija umjesto pojedinačnih obrada događaja
 - (tema predavanja 15.3 za samostalno čitanje)

Dijelovi grafičkog korisničkog sučelja (GUI) u Javi

- grafičko korisničko sučelje (GUI) u Javi se sastoji od:
 - AWT (Abstract Window Toolkit) - paket `java.awt`
 - komponente ovisne o operacijskom sustavu
 - osnovne grafičke klase
 - danas se koriste samo dijelovi u sklopu Swinga
 - Swing
 - komponente neovisne o operacijskom sustavu
 - nadogradnja AWT-ovih klasa
 - danas uobičajeni način izrade grafičkog korisničkog sučelja u Javi
 - paket `javax.swing`
 - 2D API
 - klase za crtanje u 2D-u u Swing-komponentama – nećemo koristiti
 - Accessibility API – nećemo razmatrati
 - klase koje omogućuju komponentama Swinga pristup tehnologiji koje pomaže osobama s posebnim potrebama u radu s računalom
 - Java FX – nećemo razmatrati

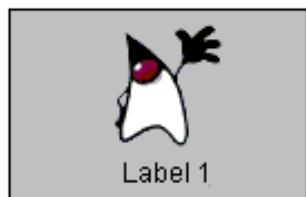
Hijerarhija bitnih klasa



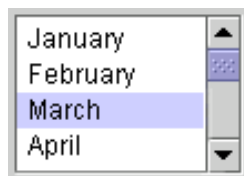
Osnove komponente GUI-a

- klasa *Component*
 - osnovna klasa iz AWT-a za sve vizualne komponente (meni, gumb, liste, labela, ...)
- klasa *Container*
 - osnovna klasa za vizualne komponente koje mogu unutar sebe sadržavati druge komponente
 - također iz AWT-a, nasljeđuje *Component*
- klasa *JComponent*
 - osnovna klasa za sve vizualne komponente u Swingu (nasljeđuje *Container*)
- sučelja *LayoutManager* i *LayoutManager2*
 - sučelja koje definiraju metode za pozicioniranje i postavljanje veličine komponenti unutar *Container*a
- klasa *Graphics*
 - apstraktna klasa koja definira metode za grafičke operacije unutar *Component*a. Svaki objekt *Component* ima dodijeljen (ili ga primaju u metodama za crtanje) objekt tipa *Graphics*

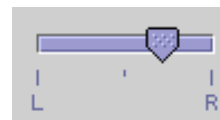
Pregled Swingovih komponenti (1)



poruke



lista



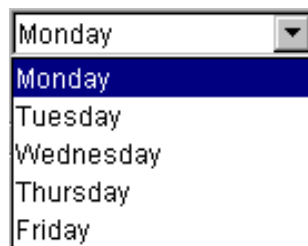
klizač



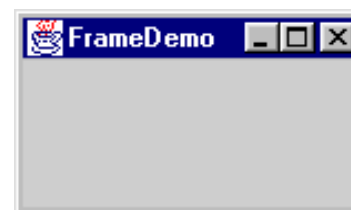
pokazivač
napredovanja



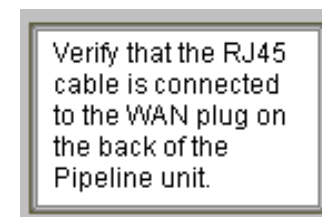
gumbi



padajuća lista



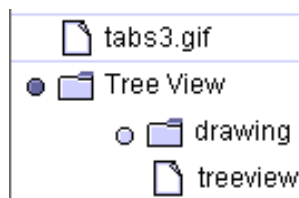
prozor



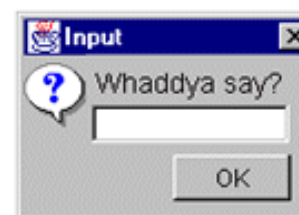
tekst



polja za tekst



stablo



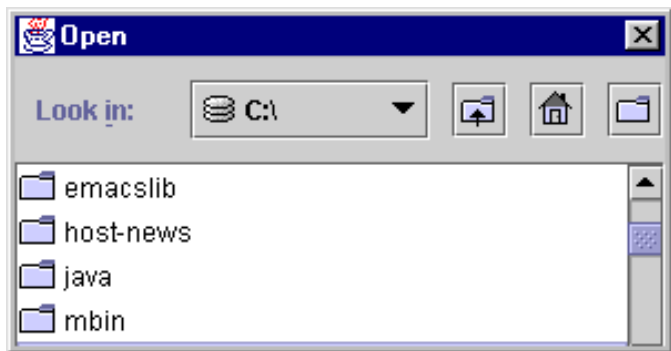
dijalog

First Na...	Last Name
Mark	Andrews
Tom	Ball
Alan	Chung
Jeff	Dinkins

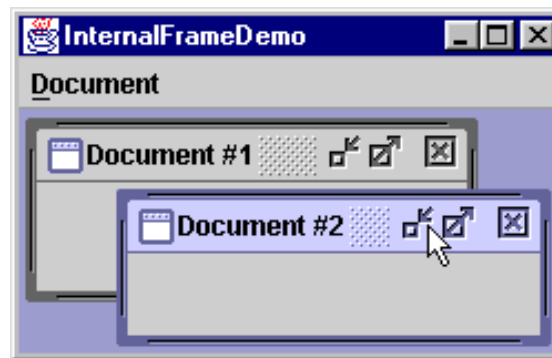
tablica

<http://docs.oracle.com/javase/tutorial/uiswing/components/componentlist.html>

Pregled Swingovih komponenti (2)



dijalog datoteka



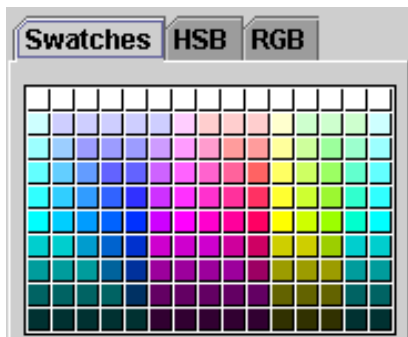
interni prozori



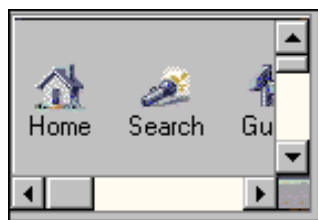
izbornik



izbornik alata



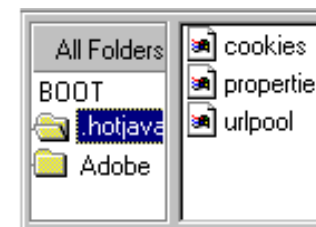
dijalog boja



pomični panel



kartični panel



podijeljeni
panel

<http://docs.oracle.com/javase/tutorial/uiswing/components/componentlist.html>

Vršne grafičke komponente

- u Swingu postoje 3 vršne komponente (engl. top level containers):
 - JFrame
 - standardni prozor
 - JDialog
 - dijalog koji je vezan za neku drugu komponentu (dijaloška kutija)
 - JApplet
 - služi za prikaz unutar web-stranice u pregledniku (engl. browser)
 - nećemo ga obrađivati na kolegiju
- svaka komponenta sadrži *contentPane* u koji se stavljaju druge komponente (vidjeti ćemo kasnije)

Stvaranje i otvaranje prozora

```
public class FirstWindow extends JFrame {
    public FirstWindow() {
        JLabel label = new JLabel("Hello from Swing GUI.");
        add(label);
        setSize(200, 100);
    }
    public static void main(String[] args) {
        try {
            SwingUtilities.invokeAndWait(new Runnable() {
                @Override
                public void run() {
                    FirstWindow window = new FirstWindow();
                    window.setLocation(20, 20); window.setVisible(true);
                }
            });
        } catch (InvocationTargetException | InterruptedException e){
            e.printStackTrace();
        }
        System.out.println("Main program continues...");
    }
}
```

Komentari na FirstWindow (1)

- Klasa *FirstWindow* nasljeđuje *JFrame*
 - U ovom jednostavnom primjeru glavni program napisan unutar iste klase, ali može biti i negdje drugdje
- Zadaća glavnog programa je pripremiti (instancirati) glavni prozor, odrediti mu početnu lokaciju prikaza i prikazati ga na ekranu
- Dijelovi i postavke glavnog prozora se postavljaju u konstruktoru,
 - objekt tip *JLabel* s nekim tekstom (dodan u prozor metodom *add*)
 - postavljena inicijalna veličina glavnog prozora (metoda *setSize*)
- Nakon toga glavni program nastavlja dalje sa svojim radom, a grafički prozor i ostale akcije vezane uz njega nastavljaju s radom neovisno o glavnom programu (rade paralelno)
 - provjerite pokretanjem iz naredbenog retka
 - **Napomena: kada se klikne na x prozor se samo skriva, ali je glavni prozor i dalje aktivan - bit će „popravljeno” u sljedećem primjeru**

Komentari na FirstWindow (2)

- Napišemo li dvije naredbe u glavnom programu jednu za drugom, one će se izvršiti sekvencijalno
 - druga počinje s izvršavanjem tek nakon što prva završi – **jedna nit izvršavanja**
 - nit ili dretva – engl. *thread*
- Akcije vezane za komponente grafičkog sučelja imaju svoju posebnu dretvu za izvršavanje
 - naziva se grafičkom dretvom – engl. event dispatch(ing) thread (EDT)
 - omogućava paralelno izvršavanje akcija vezanih za grafičko sučelje, neovisno o tome što se zbiva u kodu koji ga je pokrenuo
- Pripremu prozora glavni program treba obaviti na EDT-u koristeći metode *invokeAndWait* ili *invokeLater* iz klase *SwingUtilities*
 - argument je objekt tipa *Runnable* (funkcijsko sučelje s metodu `void run()`)
 - razlika između gore navedenih metoda je hoćemo li pričekati da se programski kod naveden u metodi *run* izvrši ili ne
 - trenutak izvršavanja može ovisiti o nekim drugim faktorima, ali o tome naknadno

„Popravljeni prozor“

- Napomena: umjesto try-catch bloka za provjeravane iznimke, postavljeno da main baca iznimku, a umjesto anonimne klase za Runnable korišten lambda izraz.

```
public class CloseableWindow extends JFrame {  
    public CloseableWindow() {  
        JLabel label = new JLabel("Hello from Swing GUI.");  
        add(label);  
        setSize(200, 100);  
        setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);  
        setTitle("Closable Window");  
    }  
    public static void main(String[] args) throws Exception {  
        SwingUtilities.invokeLater(() -> {  
            CloseableWindow window = new CloseableWindow();  
            window.setLocation(20, 20);  
            window.setVisible(true);  
        });  
        System.out.println("Main program continues...");  
    }  
}
```

...

CloseableWindow

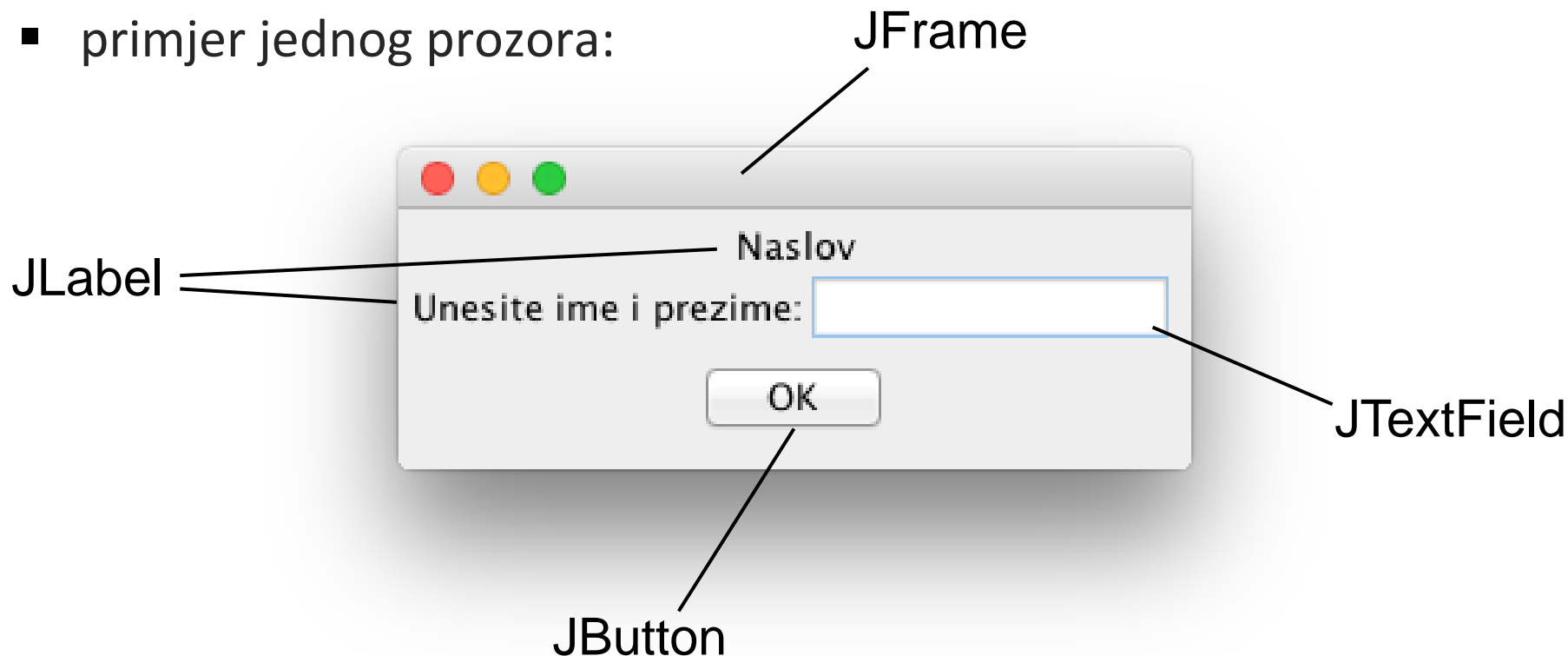
- metoda *setDefaultCloseOperation*
 - podešava što će se dogoditi kada se prozor zatvori (konstante u *WindowConstants*):
 - DO_NOTHING_ON_CLOSE – ne radi ništa
 - HIDE_ON_CLOSE – skriva prozor (podrazumijevano)
 - DISPOSE_ON_CLOSE – skriva i oslobađa resurse koje je prozor zauzeo
 - ovo koristiti kada ne želimo izaći iz aplikacije
 - EXIT_ON_CLOSE – gasi aplikaciju i proces u kojem je sve pokrenuto
 - ovo koristimo kada želimo izaći iz aplikacije
- metoda *setSize*
 - podešava veličinu prozora (širina i visina u pikselima)
- metoda *setTitle* – podešava naslov

JComponent – osnovna komponenta

- apstraktna klasa koja sadrži
 - okvir (eng. border)
 - pozadinsku i crtaću boju (eng. background, foreground)
 - prozirnost (eng. opaque)
 - font
 - veličinu i poziciju:
 - *setPreferredSize()*, *getPreferredSize()* – preporučena veličina komponente
 - *setMinimumSize()*, *getMinimumSize()* – minimalna veličina komponente
 - *setMaximumSize()*, *getMaximumSize()* – maksimalna veličina komponente
 - *getHeight()*, *getWidth()*, *getSize()* – vraća veličinu/e komponente
 - *getBounds()* – vraća veličinu i koordinate komponente
- podrška za:
 - skočne izbornike (eng. popup menu), copy/cut/paste, tooltip, o{ne}mogućavanje, vidljivost, obrada osnovnih događaja (miš, tipkovnica, ...), iscrtavanje, ugniježdene komponente

Struktura programa s grafičkim korisničkim sučeljem

- prilikom izrade programa koji koristi vizualne komponente treba uzeti u obzir organizaciju hijerarhije klasa, organizaciju vizualnih komponenti u željeni izgled, model crtanja i obradu događaja
- primjer jednog prozora:



Osnovne komponente – *JLabel*

- prikaz teksta (jedna od najjednostavnijih komponenti)
- konstruktori:
 - `JLabel(String text)`
 - `JLabel(String text, int horizontalAlignment)`
 - `SwingConstants.LEFT, CENTER, RIGHT, LEADING` ili `TRAILING`
- najčešće korištene metode:
 - `getText()`
 - `setText(String text)`
 - `getVerticalAlignment()`,
 - `setVerticalAlignment(int alignment)`
 - `TOP, CENTER` ili `BOTTOM`
 - `setIcon(Icon icon)`
 - obično se koristi implementacija `ImageIcon` koja može učitati JPEG, GIF ili PNG

Primjer kontrole *JLabel* sa slikom

15_Swing/hr.fer.oop.swing0.WindowWithPictureLabel

```
public class WindowWithPictureLabel extends JFrame {  
    public WindowWithPictureLabel() {  
        setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);  
        setTitle("Picture");  
  
        ImageIcon image = new ImageIcon("fer.jpg");  
        JLabel label = new JLabel(image);  
  
        add(label);  
    }  
    ...  
}
```



Osnovne komponente – *JTextField*



- unos (i prikaz) teksta – tekst primarno unosi korisnik
- konstruktori:
 - `JTextField(int columns)`
 - `JTextField(String text)`
- najčešće korištene metode:
 - `getText()`
 - `getText(int offs, int len),`
 - `setText(String text)`
 - `getSelectedText()`
 - `select(int selectionStart, int selectionEnd)`
 - `selectAll()`
 - `setEditable(boolean b)`
 - `Document getDocument()`
 - o korištenju naknadno

Osnovne komponente – *JButton*

- gumb namijenjen interakciji s korisnikom (npr. klik na gumb)
- konstruktori:
 - `JButton(String text)`
 - `JButton(Action a)`
 - obradit ćemo kasnije
- najčešće korištene metode:
 - `getText()`
 - `String setText(String text)`
 - `setEnabled(boolean b)`
 - `setActionCommand(String actionCommand)` i `getActionCommand()` – omogućava slanje dodatne informacije metodi za obradu događaja (npr. ako više gumba ima istu obradu događaja, pa ih je potrebno razlikovati).

Osnovne komponente –

JPasswordField i *JFormattedTextField*

- `JPasswordField` 
 - unos lozinke od strane korisnika (nasljeđuje `JTextField`)
 - metoda `setEchoChar` postavlja zamjenski znak koji se ispisuje kada unosimo lozinku
 - metoda `getPassword` vraća upisanu lozinku
- `JFormattedTextField` 
 - služi za formatirani unos (unos koji se automatski prikazuje u zadanom formatu)
 - u konstruktoru prima objekt `java.text.Format`
 - članska varijabla `value` sadrži vrijednost koja se formatira
 - metoda `getValue` vraća zadnju dobro upisanu vrijednost, a ako ništa nije upisano vraća `null`

Osnovne komponente - *JCheckBox*, *JRadioButton*

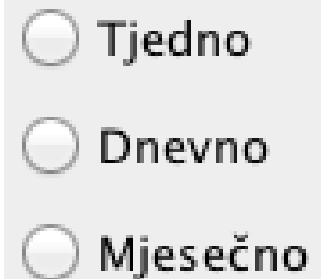
- *JCheckBox*



- mogućnost označavanja zadane vrijednosti
- konstruktor kao kod *JButton*a
- članska varijabla `selected` služi za oznaku kvačice (boolean)

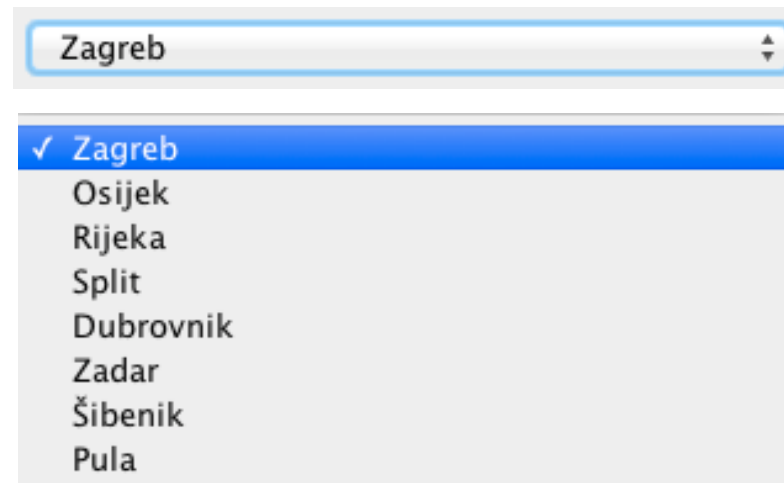
- *JRadioButton*

- odabir jedne od nekoliko ponuđenih opcija
- sličan *JCheckBoxu*
- grupiranje mogućih opcija klasom *ButtonGroup*
 - stvoriti novi objekt tipa *ButtonGroup* i korištenjem metode *add* u grupu dodati sve međusobno isključive opcije
 - nakon što se odabere nova opcija dolazi do automatskog poništavanja prethodno odabrane opcije unutar iste grupe
 - može postojati više grupa



Osnovne komponente – *JComboBox*

- padajuća lista
- tipično odabir jedne vrijednosti
 - može omogućavati upis (engl. *editable*)
- dodavanje elementata
 - u konstruktoru prima polje vrijednosti
 - ima metodu *addItem*
- odabir
 - dohvaćanje: `getSelectedIndex`, `getSelectedItem`
 - postavljanje odabira: `setSelectedIndex` ili `setSelectedItem`



Osnovne komponente – *JSlider* i *JProgressBar*

- *JSlider*

- može biti horizontalni ili vertikalni

- konstruktor:

- `JSlider(int orientation, int min, int max, int value)`

- orijentacija: `SwingConstants.VERTICAL` ili `HORIZONTAL`

- vrijednosti: min, max, inicijalna vrijednost

- postavljanje skale (broja crtica):

- `setMajorTickSpacing(int) i`

- `setMinorTickSpacing(int)`

- *JProgressBar*

- konstruktor:

`JProgressBar(int orient, int min, int max)`

- postavljanje vrijednosti: `setValue(int n)`

