

---

# GUI - Komponente

Bojan Tomić

# GUI Komponente

---


- Svaki deo grafičkog interfejsa je implementiran kao klasa paketa “javax.swing”
- Svaka od ovih klasa ima svoju grafičku reprezentaciju i naziva se “grafička komponenta”
- Ove klase imaju mnogo zajedničkih atributa ali svaka ima i neke posebne
- Kada se dodaju na prozor (JFrame), komponente se najčešće implementiraju kao atributi klase koja predstavlja prozor

# JPanel

---

- Osnovna klasa koja predstavlja praznu površinu (panel) na koju se mogu staviti komponente
- ContentPane je instanca JPanel klase
- Svaki panel može imati svoj raspored komponenti (Layout manager)

# JButton

- Klasa koja predstavlja dugme 
- Neki od atributa (property) klase JButton su
- bounds (x,y,width,height)
  - x i y su koordinate gornjeg levog ćoška dugmeta ali relativne u odnosu na Content Pane
  - width i height su dužina i visina komponente izražene u pikselima
- enabled
  - true - dugme je aktivno i reaguje na događaje
  - false - dugme nije aktivno-ne reaguje na događaje

# JButton

---

- font
  - predstavlja karakteristike fonta kojim će se ispisati tekst na dugmetu (npr. Arial, Bold, 12)
- icon
  - služi za postavljanje određene ikone na dugme
- horizontalAlignment
  - CENTER - tekst i (eventualno) ikona se pojavljuju na sredini dugmeta - po horizontalnoj osi
  - LEFT ili LEADING - tekst i (eventualno) ikona se pojavljuju uz levu ivicu dugmeta
  - RIGHT ili TRAILING - pojavljuju se uz desnu ivicu

# JButton

---

- `horizontalTextPosition`
  - LEFT ili LEADING - ako dugme ima ikonu, tekst će se ispisati sa leve strane ikone
  - RIGHT ili TRAILING - tekst će se ispisati sa desne strane ikone
  - CENTER - tekst će se ispisati na sredini i to preko ikone
- `location (x,y)`
  - x i y su koordinate gornjeg levog ćoška dugmeta ali relativne u odnosu na Content Pane

# JButton

---

- size (width, height)
  - width - širina dugmeta u pikselima (po X osi)
  - height - visina dugmeta u pikselima (po Y osi)
- text
  - naslov koji je ispisan na dugmetu
- toolTipText
  - tekst koji se ispiše korisniku svaki put kad pokazivač miša stavi preko dugmeta a ne klikne na njega

# JButton

---

- `verticalAlignment`
  - `CENTER` - tekst i (eventualno) ikona se pojavljuju na sredini dugmeta - po vertikalnoj osi
  - `TOP` - tekst i (eventualno) ikona se pojavljuju uz gornju ivicu dugmeta
  - `BOTTOM` - pojavljuju se uz donju ivicu
- `verticalTextPosition`
  - `TOP` - ako dugme ima ikonu, tekst će se ispisati iznad ikone
  - `BOTTOM` - tekst će se ispisati ispod ikone
  - `CENTER` - tekst će se ispisati preko ikone



# JButton

---

- visible
  - true - kada se inicijalizuje, dugme je vidljivo
  - false - kada se inicijalizuje, dugme nije vidljivo
- Mnoge komponente imaju neke od ovih atributa pa se neće ponovo navoditi i objašnjavati već će se samo navoditi i objašnjavati posebni atributi (ukoliko ih komponenta ima)

# JLabel

- JLabel klasa predstavlja labelu koja može imati i neku sliku na sebu
- Pri tome ova klasa nema nikakvu posebnu funkciju i služi samo radi estetike
- JLabel nema neke posebne attribute i uglavnom su isti kao kod JButton



# JCheckBox

- JCheckBox klasa predstavlja kućicu koja može da se označi (check).
- Pored teksta, može da ima i ikonu
- Posebni atributi
- selected
  - true - kućica je označena
  - false - kućica nije označena
- Ako se želi proveriti da li je kućica označena poziva se isSelected metoda



# JCheckBox

---

- Ne postoji ograničenje u smislu broja označenih komponenti
- JCheckBox se koristi kada je potrebno omogućiti višestruki izbor opcija (može se izabrati više alternativa)

# JRadioButton

- JRadioButton klasa takođe predstavlja dugme koje može da se označi (check).
- Pored teksta, može da ima i ikonu
- Posebni atributi
- selected
  - true - kućica je označena
  - false - kućica nije označena
- Ako se želi proveriti da li je dugme označeno poziva se isSelected metoda



# JRadioButton

---

- Za razliku od JCheckBox, ovde postoji ograničenje u smislu broja označenih komponenti
- JRadioButton se koristi kada je potrebno omogućiti izbor samo jedne opcije
- U tu svrhu se koristi klasa “`javax.swing.ButtonGroup`”
- Ako grupa sadrži više JRadioButton objekata, može biti označen samo jedan

# JRadioButton

---

- Kreiranje grupe
  - `ButtonGroup bg = new ButtonGroup();`
- Dodavanje JRadioButton u grupu
  - `bg.add(JRadioButton);`

# JTextField

- JTextField klasa predstavlja polje za unos podatka u jednom redu
- Posebni atributi
- text
  - String - tekst koj se unese u polje za unos
- Za preuzimanje teksta se koristi metoda “getText” a za postavljanje teksta metoda “setText”



JTextField

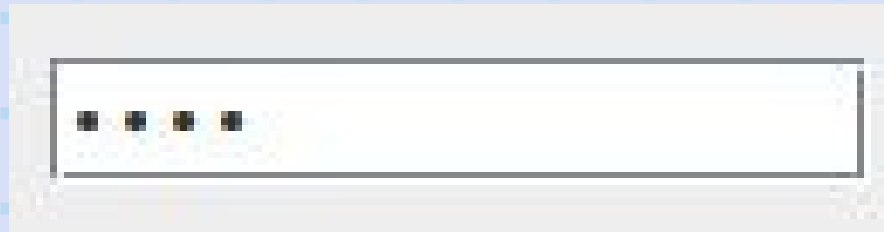


# JTextField

---

- Tekst je u String formatu, pa je često potrebno konvertovanje u druge formate
- Za double
  - `Double.parseDouble(String);`
- Za int
  - `Integer.parseInt(String);`

# JPasswordField



- Kao obično polje za unos osim što se uneti tekst ne vidi.
- getPassword metoda vraća niz znakova (char[ ]) sa unetom šifrom.

# JTextArea

- Polje za unos podatka u više redova
- Posebni atributi
- text
  - String - tekst koj se unese u polje za unos
- editable
  - true - tekst može da se menja
  - false - ne može da se menja



JTextArea

# JTextArea

---

- `lineWrap`
  - `true` - tekst koji se unese u polje biće automatski prelomljen u više redova
  - `false`
- `wrapStyleWord`
  - `true` - tekst koji se unese u polje biće automatski prelomljen u više redova ali tako da se prenose cele reči
  - `false`

# JTextArea

---

- Za preuzimanje teksta se koristi metoda “getText” a za postavljanje teksta metoda “setText”
- Za dodavanje teksta na kraj već postojećeg teksta, može da se koristi i metoda “append”
- JTextArea kao takav ne dolazi sa komponentama za skrolovanje teksta
- Ako tekst prevaziđe obim komponente, ne može se pomeriti tako da se vidi

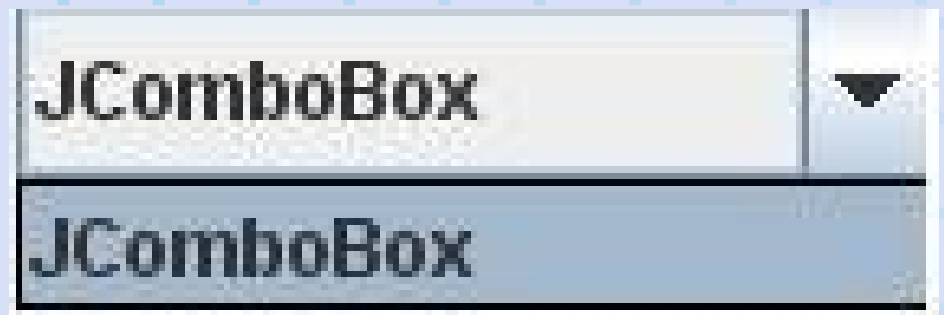
# JTextArea

---

- Da bi se ovo otklonilo, potrebno je uraditi sledeće:
  - Postaviti prvo komponentu JScrollPane na prozor
  - Na JScrollPane postaviti JTextArea
- Kada se ovako uradi, komponenta će automatski uključiti skrolovanje kada tekst bude prevazišao obime

# JComboBox

- Komponenta koja predstavlja padajuću listu
- Koristi se kada je potrebno ograničiti korisnika na određeni konačni skup opcija



# JComboBox

---

- Dodavanje stavki - metoda “addItem(Object)”
- Stavke liste mogu biti bilo koji objekti uključujući i String
- Kao tekst stavke će se ispisati ono što “toString” metoda unetog objekta vrati kao vrednost
- “getSelectedIndex()” vraća redni broj stavke koja je izabrana (-1 ako ništa nije izabrano)
- “getSelectedItem()” vraća izabranu stavku kao ceo objekat



# JList

- Predstavlja običnu listu stavki
- Koristi se situacijama kada je potrebno prikazati konačnu listu nekih odvojenih entiteta ili opcija



# JList

---

- Dodavanje stavki se vrši ručno korišćenjem metode “setListData(Object[])” ili “setListData(Vector)”
- U prvom slučaju, metodi se prosleđuje niz bilo kakvih objekata - svaki objekat će postati stavka
- U drugom slučaju, metodi se prosleđuje objekat klase Vector koji u sebi sadrži niz objekata

# JList

---

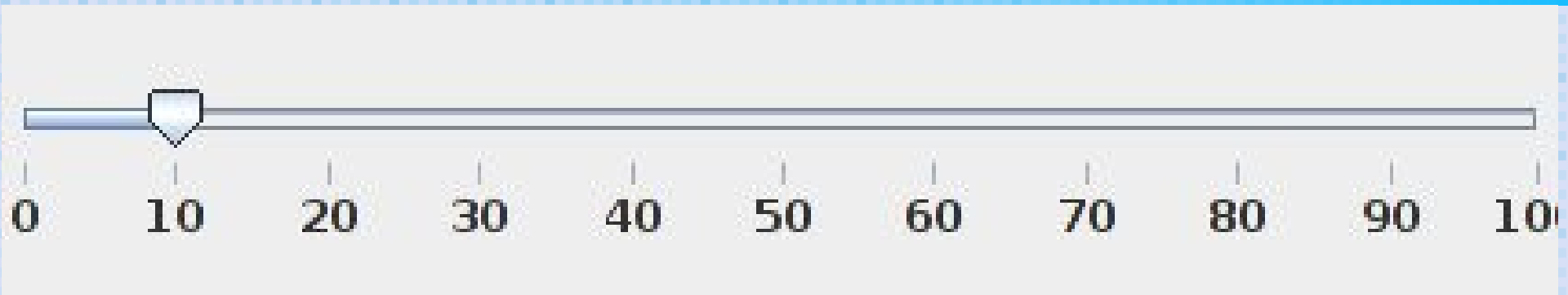
- “getSelectedIndex()” vraća redni broj stavke koja je izabrana (-1 ako ništa nije izabrano)
- “getSelectedValue()” vraća izabranu stavku kao ceo objekat

# JList

---

- JList nema opciju za skrolovanje teksta (slično kao JTextArea), pa je potrebno uraditi sledeće:
  - Postaviti prvo komponentu JScrollPane na prozor
  - Na JScrollPane postaviti JList
- Kada se ovako uradi, komponenta će automatski uključiti skrolovanje kada tekst bude prevazišao obime

# JSlider



- Klizna skala
- maximum – maksimalana vrednost na skali
- minimum – minimalna vrednost na skali
- value – početna tj. izabrana vrednost na skali
- majorTickSpacing – razmak između velikih podeoka na skali
- orientation – horizontalana ili vertikalna skala

# JSlider

---

- `paintLabels` – da li da se prikažu brojevi
- `paintTicks` – da li da se prikažu podeoci
- `paintTrack` – da li da se prikaže linija po kojoj ide klizač skale
- `snapToTicks` – da li da klizač može samo da se zaustavlja na podeocima ili i između
- Preuzimanje izabrane vrednosti preko `getValue` metode

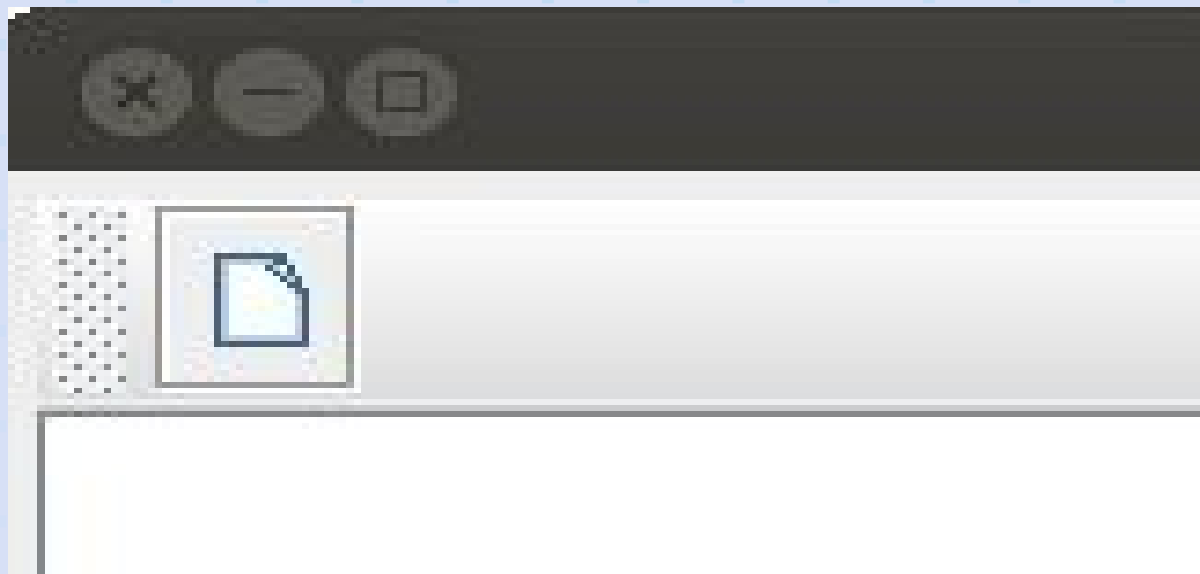
# JSpinner



- Polje za unos ali sa ograničenim skupom vrednosti
- Brojevi, ali i bilo koje liste vrednosti
- Skup vrednosti se definiše preko modela
  - Spinner number model (max, min, početna vrednost, korak)
  - Spinner list model (bilo koja lista tj. njeni elementi)
  - Spinner date model

# JToolBar

- Paleta sa komponentama (obično JButton komponente)
- Može da se pomera u okviru ekrana ili da se izvuče van ekrana
- JSeparator – separator za podesivi razmak između komponenti, gde je to potrebno





# JTable

- Komponenta za predstavljanje tabele
- Čelije ne moraju biti polja za unos već i druge komponente
- Tabela – Model – Podaci
- Stavlja se na JScrollPane da bi imala skrolere i da bi se automatski prikazalo zaglavlje

Kolona 1	Kolona 2	Kolona 3
		<input type="checkbox"/>
		<input type="checkbox"/>
		<input type="checkbox"/>

# JTable

---

- `fillsViewportHeight` – da li da tabela ispuni i ostatak prostora na panelu u kojem se nalazi (npr. na `JScrollPane`)
- `showGrid` – da li da se iscrtavaju okviri oko ćelija (pa se tek onda gledaju sledeća dva)
- `showHorizontalLines` – da se prikažu horizontalne linije oko ćelija
- `showVerticalLines` – da se prikažu vertikalne linije oko ćelija

# JTable

---

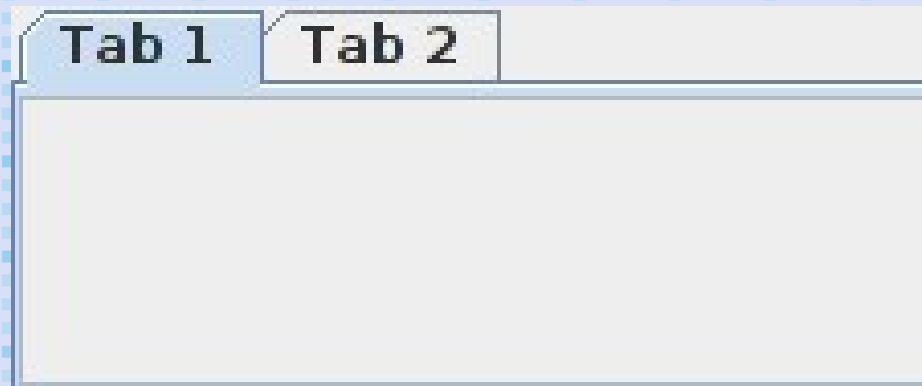
- `rowSelectionAllowed` – da li je dozvoljeno označavanje celih redova
- `columnSelectionAllowed` – da li je dozvoljeno označavanje celih kolona
- `cellSelectionAllowed` – da li je dozvoljeno označavanje pojedinačnih ćelija
- `getValueAt (int row, int column)` – vraća vrednost određene ćelije
- `setValueAt (int row, int column)` – vraća vrednost određene ćelije

# JTree

- Prikaz neke hijerarhije u vidu stabla
- Koren, čvor, list
- JTree – Tree model
- rootVisible – da li se vidi koren stabla
- ShowsRootHandles – da li se vidi “drška” na korenu



# JTabbedPane



- Klasa koja predstavlja više panela sa jezičcima
- Svaki jezičak ima svoj naslov (opciono i ikonicu) i vezan je tačno za jedan panel (JPanel)
- Svaki panel može imati različite komponente
- U svakom trenutku je vidljiv samo jedan panel

# JTabbedPane

---

- Metoda koja vraća redni broj trenutno izabranog jezička (-1 ako nijedan nije izabran)  
`getSelectedIndex()`
- Ova komponenta se koristi ako je na istom prozoru potrebno prikazati veliki broj komponenti ali koje se mogu podeliti u više grupa.

# JSplitPane

- Omogućava podelu površine na dva dela čija veličina se može menjati mišem
- Ta dva dela se mogu dalje deliti na manje celine novim JsplittedPane komponentama itd.
- continuousLayout – konstantno iscrtavanje kad se separator pomera

