

## 1 JavaFX UI kontrole

JavaFX kontrole predstavljene su uobičajeno, čvorovima u grafu scene. Klase koje opisuju kontrole nalaze se u paketu **javafx.scene.control**.

Za kontrole se mogu postaviti i stilovi primenom CSS-a, kao i odgovarajući efekti.

### 2 Hijerarhija apstraktne klase *javafx.scene.control.Labeled*

```
java.lang.Object →  
    javafx.scene.Node →  
        javafx.scene.Parent →  
            javafx.scene.control.Control →  
                javafx.scene.control.Labeled
```

#### Potklase:

**ButtonBase, Cell, Label, TitlePane**

Apstraktna klasa *Labeled* opisuje kontrolu koja kao deo svog korisničkog interfejsa ima tekstualni sadržaj.

Na primer, dugme (*Button*) prikazuje tekst, kao što to čini i labela (*Label*), *Tooltip* (kratak opis namene objekta koji se prikazuje kada se miš pozicionira iznad objekta) i mnoge druge kontrole.

<http://docs.oracle.com/javafx/2/api/javafx/scene/control/Labeled.html>

### 3 Apstraktna klasa *ButtonBase*

#### Potklase:

**Button, CheckBox, Hyperlink, MenuButton, ToggleButton**

U vezi sa klasom *ButtonBase* je i koncept označen kao „arming“. Na primer, dugme ima status „arming“ ukoliko je dugme omogućeno i nad dugmetom je pritisnuto dugme miša. Kada se dugme otpusti, dešava se odgovarajuća akcija (dugme je „fired“).

<http://docs.oracle.com/javafx/2/api/javafx/scene/control/ButtonBase.html>

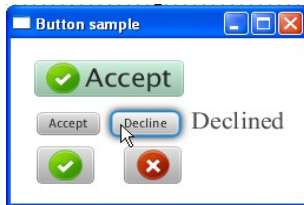
## 4 Button

Jednostavna kontrola – dugme. Može da sadrži tekst i/ili grafiku.

Klasa *Button* implementira interfejs *Skinnable* (sve kontrole implementiraju ovaj interfejs) i *EventTarget* (što znači da može biti cilj događaja).

Kontrola dugme ima 3 različita moda:

- **Normal:** normalno dugme koje reaguje na pritisak
- **Default:** podrazumevano dugme koje reaguje na pritisak *VK\_ENTER* sa tastature, ukoliko ga nijedan drugi čvor sa scene nije preuzeo
- **Cancel:** dugme za otkazivanje koje reaguje na pritisak *VK\_ESC* sa tastature, ukoliko ga nijedan drugi čvor sa scene nije preuzeo



### Konstruktori

**Button()** - kreira dugme sa praznim stringom kao labelom

**Button(java.lang.String text)** - kreira dugme sa naznačenim tekstom kao labelom

**Button(java.lang.String text, Node graphic)** - kreira dugme sa naznačenim tekstom i ikonicom kao labelom

```
Button button = new Button("Click Me");
```

### Nasledene osobine

Iz klase **javafx.scene.control.Labeled**:

alignment, contentDisplay, ellipsisString, font, graphic, graphicTextGap, labelPadding, mnemonicParsing, textAlignment, textFill, textOverrun, text, underline, wrapText

Iz klase **javafx.scene.control.Control**:

contextMenu, height, maxHeight, maxWidth, minHeight, minWidth, prefHeight, prefWidth, skinClassName, skin, tooltip, width  
...

### Metodi

- **boolean isCancelButton()**
- **boolean isDefaultButton()**
- **void setCancelButton(boolean value)**
- **void setDefaultButton(boolean value)**

### Obrada događaja dugmeta

Kada se pritisne i otpusti dugme šalje se objekat **ActionEvent**. Aplikacija može da izvrši neku akciju na osnovu tog događaja. Akcija se definiše implementacijom **EventHandler**-a koji obrađuje **ActionEvent**.

Dugme takođe može da reaguje i na događaje mišem, tako što se **EventHandler**-u prosleđuju na obradu **MouseEvent** objekti.

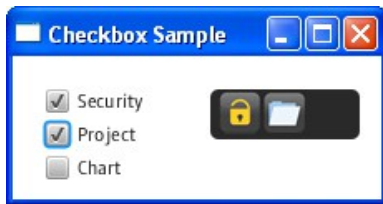
<http://docs.oracle.com/javafx/2/api/javafx/scene/control/Button.html>




## 5 CheckBox

**CheckBox** kontrola može da bude u tri stanja, mada se najčešće koristi kao kontrola u dva stanja (prva dva navedena u listi):

- **checked**: indeterminate == false, checked == true

- **unchecked:** indeterminate == false, checked == false
- **undefined:** indeterminate == true



Property Values	Checkbox Appearance
INDETERMINATE = false SELECTED = false	 I agree
INDETERMINATE = false SELECTED = true	 I agree
INDETERMINATE = true SELECTED = true/false	 I agree

### Konstruktori

**CheckBox()** - kreira check box sa praznim stringom kao labelom

**CheckBox(java.lang.String text)** - kreira check box sa naznačenim tekstom kao labelom

```
CheckBox cb = new CheckBox("a checkbox");  
cb.setIndeterminate(false);
```

### Osobine

**boolean allowIndeterminate** - određuje da li mogu da se koriste sva tri stanja

**booleanProperty indeterminate** - određuje da li je CheckBox u stanju indeterminate

**booleanProperty selected** - pokazuje da li je CheckBox označen

### Nasledene osobine

Iz klase **javafx.scene.control.Labeled**:

alignment, contentDisplay, ellipsisString, font, graphic, graphicTextGap, labelPadding, mnemonicParsing, textAlignment, textFill, textOverrun, text, underline, wrapText

Iz klase **javafx.scene.control.Control**:

contextMenu, height, maxHeight, maxWidth, minHeight, minWidth, prefHeight, prefWidth, skinClassName, skin, tooltip, width

### Metodi

- **booleanProperty allowIndeterminateProperty()**
- **boolean isIndeterminate()**

- **boolean isSelected()**
- **void setIndeterminate(boolean value)**
- **void setSelected(boolean value)**
- ...

<http://docs.oracle.com/javafx/2/api/javafx/scene/control/CheckBox.html>

## 6 ToggleButton

Kontrola koja pokazuje da li je opcija izabrana ili ne.

**Potklasa:**

**RadioButton**

Za razliku od `RadioButton`, ako je `ToggleButton` dugme izabrano i ponovo se klikne na njega, njegov izbor će biti poništen, dok kod `RadioButton` dugmeta, u tom slučaju neće biti promena. U grupi `RadioButton` tačno jedna opcija može da bude izabrana, dok to nije slučaj kod `ToggleButton` grupe.

**Konstruktori**

**ToggleButton()** - kreira dugme sa praznim stringom kao labelom

**ToggleButton(java.lang.String text)** - kreira dugme sa naznačenim tekstom kao labelom

**ToggleButton(java.lang.String text, Node graphic)** - kreira dugme sa naznačenim tekstom i ikonicom kao labelom

`ToggleButton` dugmad se najčešće smeštaju u grupu.

**Primer**

```
ToggleButton tb1 = new ToggleButton("toggle button 1");
ToggleButton tb2 = new ToggleButton("toggle button 2");
ToggleButton tb3 = new ToggleButton("toggle button 3");
ToggleGroup group = new ToggleGroup();
tb1.setToggleGroup(group);
tb2.setToggleGroup(group);
tb3.setToggleGroup(group);
```

**Osobine**

**BooleanProperty selected** - označava da li je dugme izabrano

**ObjectProperty<ToggleGroup> toggleGroup** - grupa kojoj `ToggleButton` pripada

**Nasledene osobine**

Iz klase `javafx.scene.control.Labeled`:

`alignment`, `contentDisplay`, `ellipsisString`, `font`, `graphic`, `graphicTextGap`, `labelPadding`, `mnemonicParsing`, `textAlignment`, `textFill`, `textOverrun`, `text`, `underline`, `wrapText`

Iz klase **javafx.scene.control.Control**:

contextMenu, height, maxHeight, maxWidth, minHeight, minWidth, prefHeight, prefWidth, skinClassName, skin, tooltip, width

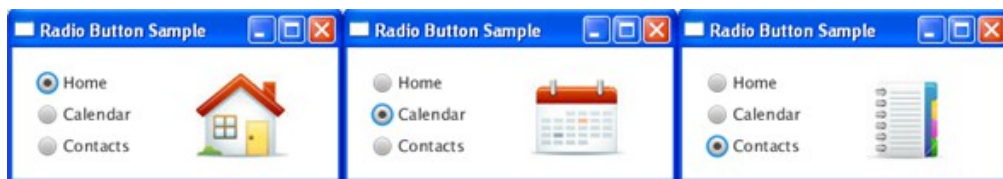
### Metodi

- **boolean isSelected()**
- **void setSelected(boolean value)**
- **void setToggleGroup(ToggleGroup value)**
- **ObjectProperty<ToggleGroup> toggleGroupProperty()**
- ...

<http://docs.oracle.com/javafx/2/api/javafx/scene/control/ToggleButton.html>

## 7 RadioButton

Klasa **RadioButton** izvedena je iz klase **ToggleButton**.



Podrazumevano **RadioButton** objekat ne pripada **ToggleGroup** grupi i neophodno ga je pridružiti.

```
ToggleGroup group = new ToggleGroup();
RadioButton button1 = new RadioButton("select first");
button1.setToggleGroup(group);
button1.setSelected(true);
RadioButton button2 = new RadioButton("select second");
button2.setToggleGroup(group);
```

### Obrada događaja **RadioButton** dugmeta

Kada se pritisne i otpusti **RadioButton** dugme šalje se objekat **ActionEvent**.

Aplikacija može da izvrši neku akciju na osnovu tog događaja. Akcija se definiše implementacijom **EventHandler**-a koji obrađuje **ActionEvent**.

Dugme takođe može da reaguje i na događaje mišem, tako što se **EventHandler**-u prosleđuju na obradu **MouseEvent** objekti.

<http://docs.oracle.com/javafx/2/api/javafx/scene/control/RadioButton.html>

## 8 Label



Labele se koriste kao tekst kontrole kojima ne može da se menja sadržaj od strane korisnika. Korisne su za prikazivanje teksta (natpisa, oznaka...). Labele su takođe korisne jer mogu da im budu pridruženi mnemonici, koji kada se koriste mogu da pošalju fokus kontroli koja je naznačena kao ciljna kontrola svojstva `labelFor`. Dakle, svojstvo `labelFor` označava da se labela zapravo odnosi na drugu kontrolu ili čvor.

### Konstruktori

**Label()** - kreira labelu sa praznim stringom

**Label(java.lang.String text)** - kreira labelu sa naznačenim tekstom

**Label(java.lang.String text, Node graphic)** - kreira labelu sa naznačenim tekstom i ikonicom

```
Label label = new Label("Ime i prezime: ");
```

### Nasledene osobine

Iz klase **javafx.scene.control.Labeled**:

`alignment`, `contentDisplay`, `ellipsisString`, `font`, `graphic`, `graphicTextGap`, `labelPadding`, `mnemonicParsing`, `textAlignment`, `textFill`, `textOverrun`, `text`, `underline`, `wrapText`

Iz klase **javafx.scene.control.Control**:

`contextMenu`, `height`, `maxHeight`, `maxWidth`, `minHeight`, `minWidth`, `prefHeight`, `prefWidth`, `skinClassName`, `skin`, `tooltip`, `width`

### Metodi

- **Node getLabelFor()** - vraća vrednost svojstva `labelFor`
- **void setLabelFor(Node value)** – postavlja vrednost svojstva `labelFor`
- ...

## 9 Hijerarhija apstraktne klase *javafx.scene.control.TextInputControl*

```
java.lang.Object →
    javafx.scene.Node →
        javafx.scene.Parent →
            javafx.scene.control.Control →
                javafx.scene.control.TextInputControl
```

### Potklase:

**TextArea**, **TextField**

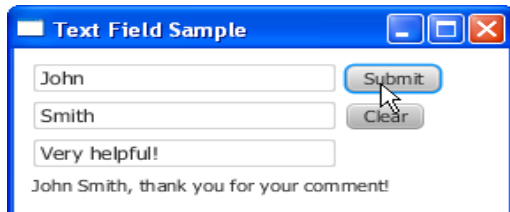
### Neke od osobina

**BooleanProperty editable** - označava da li `TextInputControl` objekat može da bude menjan od strane korisnika

**ReadOnlyIntegerProperty length** - broj karaktera u tekstualnom unosu

**StringProperty promptText** - tekst koji se inicijalno prikazuje u `TextInputControl` objektu ili `null` ukoliko nije određen

## 10 TextField



Komponenta koja omogućava korisniku da unese JEDNU neformatiranu liniju teksta.

`TextField` podržava prikazivanje inicijalnog teksta (*text prompt*) korisniku, ukoliko nije već prikazan tekst u objektu `TextField` (bilo da je uneo korisnik ili da je tekst unet iz programa). To je koristan način da se korisnik informiše o tome šta se od njega očekuje da unese u tekst polje.

### Konstruktori

**`TextField()`** - kreira tekst polje bez tekstuelnog sadržaja

**`TextField(java.lang.String text)`** - kreira tekst polje sa naznačenim tekstom

```
Label label = new Label("Name:");
TextField textField = new TextField();
textField.setPromptText("Unesite ime.");
HBox hb = new HBox();
hb.getChildren().addAll(label, textField);
hb.setSpacing(10);
```

### Neki nasleđeni metodi

Iz klase `javafx.scene.control.TextInputControl`:

`appendText`, `clear`, `copy`, `cut`, `deleteNextChar`, `deletePreviousChar`, `deleteText`, `end`, `endOfNextWord`, `getLength`, `getPromptText`, `getSelectedText`, **`getText`**, `isEditable`, `nextWord`, `previousWord`, `replaceText`, `selectAll`, `setEditable`, `setPromptText`, **`setText`**

Iz klase `javafx.scene.control.Control`:

`getHeight`, `getMaxHeight`, `getMaxWidth`, `getMinHeight`, `getMinWidth`, `getPrefHeight`, `getPrefWidth`, `getWidth`, `maxHeight`, `maxHeightProperty`, `maxWidth`...

### Metodi

- **`void setAlignment(Pos value)`**
- **`Pos getAlignment()`** ...

## **11 TextArea**

Komponenta koja omogućava korisniku da unese neformatirani tekst u više linija. Takođe je moguće postaviti inicijalni tekst.

### **Konstruktori**

**TextArea()** - kreira tekst oblast bez tekstuelnog sadržaja

**TextArea(java.lang.String text)** - kreira tekst oblast sa naznačenim tekstom

Najčešće se smešta u objekat **javafx.scene.control.ScrollPane**, radi boljeg pregleda:

```
public void start(final Stage primaryStage) {
    Group root = new Group();
    final TextArea textArea = TextAreaBuilder.create()
        .prefWidth(400)
        .wrapText(true)
        .build();
    ScrollPane scrollPane = new ScrollPane();
    scrollPane.setContent(textArea);
    scrollPane.setFitToWidth(true);
    scrollPane.setPrefWidth(400);
    scrollPane.setPrefHeight(180);
    VBox vBox = VBoxBuilder.create()
        .children(scrollPane)
        .build();
    root.getChildren().add(vBox);
    primaryStage.setScene(new Scene(root, 400, 180));
    primaryStage.show();
}
```

## **12 Korisni linkovi**

### **JavaFX UI Controls**

[http://docs.oracle.com/javafx/2/ui\\_controls/jfxpub-ui\\_controls.htm](http://docs.oracle.com/javafx/2/ui_controls/jfxpub-ui_controls.htm)