

Kreiranje prve Java FX aplikacije

U prethodnoj lekciji objašnjene su osnove funkcionisanja Java FX aplikacije. Ova lekcija biće posvećena kreiranju prve jednostavne aplikacije sa Java FX grafičkim korisničkim okruženjem.

Kreiranje uslova za rad

S obzirom na to da je od verzije 8 Java programskog jezika Java FX sastavni deo JRE i JDK-a, praktično nije potrebno nikakvo dodatno preuzimanje ili podešavanje. Jednostavno, dovoljno je preuzeti JDK verzije 8 ili više i automatski sve funkcionalnosti potrebne za rad sa Java FX-om biće dostupne. Ove funkcionalnosti nalaze se zapakovane u JAR arhivu pod nazivom **jfxrt.jar**, koja se nalazi na putanji `jre\lib\ext` JDK foldera.

Kreiranje osnovne klase

Osnovana klasa Java FX aplikacije mora da nasledi klasu `Application` iz paketa `javafx.application`. `Application` klasa definiše apstraktnu metodu `start`, koja stoga mora biti implementirana u svakoj nasleđujućoj klasi koja i sama nije apstraktna.

Metoda **start** je ulazna tačka svake Java FX aplikacije.

Uzimajući sve ovo u obzir, osnovna klasa Java FX aplikacije koja bi se nalazila u paketu `main` mogla bi da izgleda ovako:

```
package main;

import javafx.application.Application;
import javafx.stage.Stage;

public class MainJavaFX extends Application{

    @Override
    public void start(Stage primaryStage) throws Exception {
        //logic goes here
    }

}
```

Kao što je već rečeno, ulazna tačka Java FX aplikacije je metoda `start`. Ona biva pozvana od Java FX application lauchera. Takođe, može se primetiti da metoda `start` prihvata jedan parametar tipa `Stage`, što je inače primarni stage aplikacije. Tokom životnog toka aplikacije mogu se kreirati i dodatni stagevi, ali primarni se uvek automatski kreira.

Pozornica

Ukoliko znate da „stage“ u stvari znači pozornica, lako se može zaključiti da se pozornica koristi za prikaz neke scene. Scena se sastoji iz vizuelnih elemenata kao što su tekst, oblici, slike, kontrole, animacije i efekti, sa kojima korisnik može vršiti interakciju. Takva interakcija je inače i osnovna svrha grafičkog korisničkog okruženja bilo koje aplikacije.

Pozornica izgleda drugačije u zavisnosti od okruženja u kome se aplikacija izvršava. Nije potrebno preduzeti bilo kakvu akciju specifičnu za određenu platformu, zato što će to automatski uraditi Java FX. Primarna pozornica koju kreira aplikacija ne poseduje nijednu scenu. Pre nego što se upustimo u kreiranje scene, hajde da prikazemo pozornicu, bez scene, kako biste mogli steći uvid u to koji su njeni elementi.

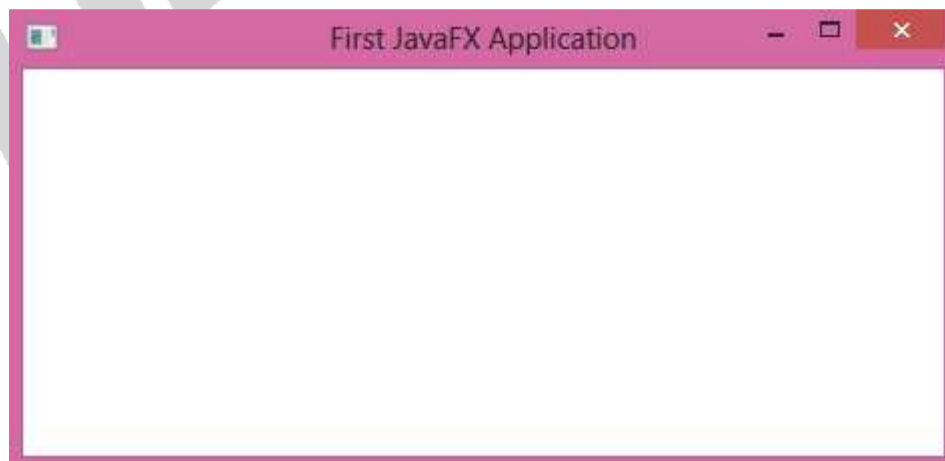
Pozornica se prikazuje pozivom metode show nad objektom tipa Stage. Stageu se može dodeliti i naziv i to metodom set Title. Sa navedenim izmenama metoda start bi izgledala ovako:

```
@Override
public void start(Stage primaryStage) throws Exception {
    primaryStage.setTitle("First JavaFX Application");
    primaryStage.show();
}
```

Da bi se aplikacija pokrenula, neophodno je dodati main metodu u već kreiranu klasu i u okviru main metode pozvati statičku metodu launch Application klase. Ova metoda prihvata niz stringova koji se odnosi na argumente koji se prosleđuju Java FX aplikaciji. Stoga, main metoda bi izgledala ovako:

```
public static void main(String[] args) {
    Application.launch(args);
}
```

Nakon pisanja do sada prikazanog koda, aplikacija se može pokrenuti. Pokretanje će dozvati sledeći prozor (na Windows 8 operativnom sistemu):



2.1 – Stage

Ono što se odmah može zapaziti jeste da je centralni deo prozora prazan. Centralni deo prozora je deo u kome je potrebno da se pojavi sadržaj i to korišćenjem scene stagea. S obzirom na to da do sada nismo kreirali nijednu scenu, centralni deo prozora je prazan. Na slici se može primetiti i naslov u naslovnoj liniji koji je podešen korišćenjem metode set Title.

Scena

Do sada se ništa preterano zanimljivo nije dogodilo u razvoju naše prve Java FX aplikacije. Zbog toga sada prelazimo na dodavanje scene kreiranoj pozornici, što će nam omogućiti dodavanje vizuelnih elemenata.

Scena se kreira korišćenjem klase Scene, koja se nalazi u javafx.scene paketu. Stage može sadržati jednu scenu, a scena veći broj vizuelnih elemenata. Sadržaj scene je organizovan u strukturu stabla. Na vrhu takve strukture je koreni element ili čvor. Ovaj koreni čvor može sadržati svoje podelemente. Nadalje, svaki od ovih potomaka može imati i svoje potomke.

Za dodavanje konkretnih elemenata sceni, te kreiranje scene, prvo će biti neophodno kreirati koreni element. Za to će sada poslužiti element tip V Box.

Napomena

Bilo koji čvor koji nasleđuje klasu javafx.scene.Parent može biti korišćen kao koreni. Određeni elementi poznati su i kao kontejneri i to su elementi koji mogu da sadrže više drugih elemenata, te se mogu upotrebiti kao koreni. Jedan od ovih elemenata je i V box, koji će biti iskorišćen u ovoj lekciji. Kontejneri će biti detaljno obrađeni u nastavku kursa.

Instanca klase V Box dobija se pozivanjem podrazumevanog konstruktora klase:

```
VBox root = new VBox();
```

Čvor koji može da sadrži potomke izlaže metodu getChildren koja vraća objekat tipa ObservableList. To je zapravo kolekcija u kojoj su sadržani svi potomci čvora. Mi ćemo ovu kolekciju iskoristiti za dodavanje jednog potomka, kojeg pre dodavanja moramo i kreirati. To će biti običan tekstualni element (kontrola):

```
Text msg = new Text("Hello from JavaFX");
```

Napomena

Java FX sadrži veliki broj kontrola, koje će biti predstavljene u nastavku kursa.

Sada ovaj element možemo dodati korenom čvoru:

```
root.getChildren().add(msg);
```

Kao što je već rečeno, kolekcija potomaka se preuzima metodom `getChildren`, a zatim se metodom `add` njoj dodaje instance novog potomka.

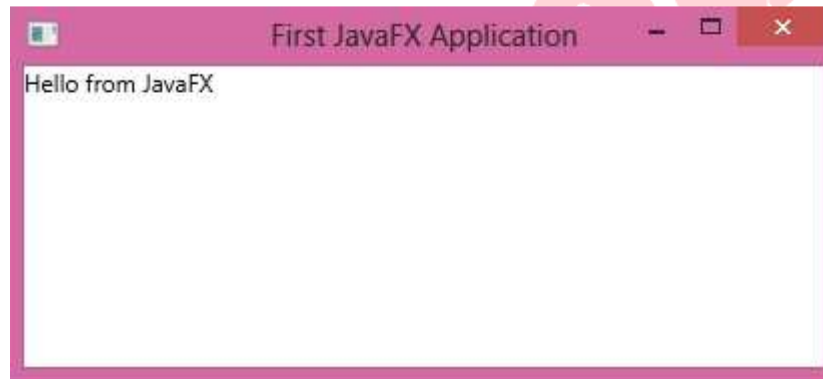
Tek sada, nakon kreiranja korenog elementa, može se pristupiti kreiranju scene. Objekat tipa `Scene` se može konstruisati korišćenjem nekoliko konstruktora u zavisnosti od potreba. Ovom prilikom će biti iskorišćen konstruktor koji prihvata tri parametra, i to koreni element, visinu i širinu scene.

```
Scene scene = new Scene(root, 300, 50);
```

Na kraju ostaje još da se scena postavi na pozornicu. Za to se koristi metoda `setScene` kojoj se prosleđuje instance `Scene` objekta:

```
primaryStage.setScene(scene);
```

Kada se ovako napisana aplikacija pokrene, rezultat će biti sledeći:



2.2 – Java FX aplikacija

Dodavanje interaktivnosti

Do sada smo kreirali prilično bazičnu Java FX aplikaciju bez bilo kakve interaktivnosti. U ovoj lekciji se nećemo ovde zaustaviti, već ćemo otići i korak dalje. Pokušajmo da kreiranu aplikaciju obogatimo tako što ćemo je oplemeniti dodatnim kontrolama koje će omogućiti ostvarivanje određene dinamičnosti.

Aplikaciji ćemo dodati dve kontrole koje će omogućiti korisniku unos imena i aktiviranje logike za prikaz odgovarajuće poruke.

Prvo će biti instancirani objekti koji će predstavljati jedan label i polje za unos teksta:

```
Label nameLbl = new Label("Enter your name:");  
TextField nameFld = new TextField();
```

Napomena

Ne zaboravite da je za korišćenje Java FX kontrola neophodno dodati odgovarajuće pakete sa prefiksom javafx.

Ovoj aplikaciji dodaćemo još i jednu Button kontrolu, koja će aktivirati ispis poruke.

```
Button sayHelloBtn = new Button("Say Hello");
```

Za kreiranu taster kontrolu potrebno je definisati logiku koja će se izvršiti nakon izvršenog pritiska tastera. To se postiže metodom `setOnAction` koja prihvata parametar tipa `EventHandler<ActionEvent>`. Instancu ovog tipa je moguće proslediti ovoj metodi anonimno, ili pak iskoristiti znatno elegantnije rešenje koje se ogleda u upotrebi lambda izraza.

```
sayHelloBtn.setOnAction((ActionEvent event) -> {  
    //logic goes here  
});
```

Prilikom klika na Button kontrolu, potrebno je da se prikaže odgovarajuća pozdravna poruka u zavisnosti od unetog imena. Ovo nameće zaključak da će biti neophodno prvo pročitati zapis koji je korisnik uneo u kontrolu za unos teksta i pozdravnu poruku modelovati u zavisnosti od tog podatka. Evo kako bi to moglo izgledati:

```
String msg = "";  
  
String name = nameFld.getText();  
if (name.trim().length() > 0) {  
    msg = "Hello " + name;  
} else {  
    msg = "Hello there";  
}
```

Poruka će se prikazati u okviru dijaloga za prikaz informacija.

```
Alert alert = new Alert(AlertType.INFORMATION);  
alert.setTitle("Information Dialog");  
alert.setHeaderText(null);  
alert.setContentText(msg);  
alert.showAndWait();
```

Na kraju je potrebno sve vizuelne elemente dodati korenom elementu. Za dodavanje većeg broja elemenata kolekciji elemenata može se iskoristiti metoda `addAll`.

Nakon svega ovoga, aplikacija je spremna za pokretanje. Kompletan kod aplikacije dat je u nastavku:

```
package main;

import javafx.application.Application;
import javafx.event.ActionEvent;
import javafx.scene.Scene;
import javafx.scene.control.Alert;
import javafx.scene.control.Alert.AlertType;
import javafx.scene.control.Button;
import javafx.scene.control.Label;
import javafx.scene.control.TextField;
import javafx.scene.layout.VBox;
import javafx.stage.Stage;

public class Main extends Application {

    public static void main(String[] args) {
        Application.launch(args);
    }

    @Override
    public void start(Stage stage) {

        VBox vBox = new VBox();

        Label nameLbl = new Label("Enter your name:");
        TextField nameFld = new TextField();
        Button sayHelloBtn = new Button("Say Hello");
        sayHelloBtn.setOnAction((ActionEvent event) -> {

            String msg = "";

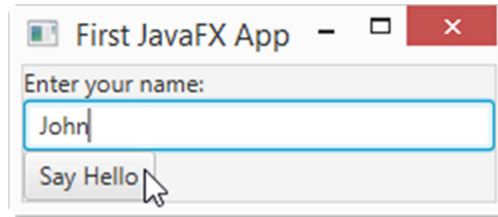
            String name = nameFld.getText();
            if (name.trim().length() > 0) {
                msg = "Hello " + name;
            } else {
                msg = "Hello there";
            }
            Alert alert = new Alert(AlertType.INFORMATION);
            alert.setTitle("Information Dialog");
            alert.setHeaderText(null);
            alert.setContentText(msg);
            alert.showAndWait();

        });

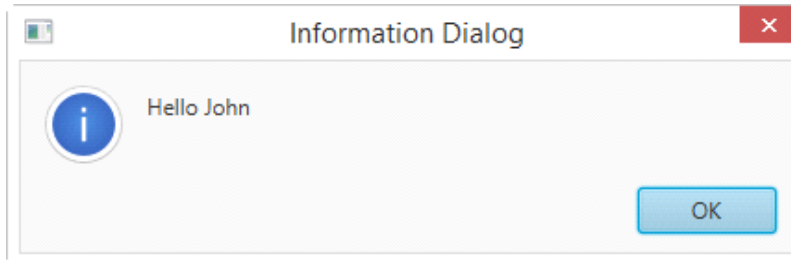
        vBox.getChildren().addAll(nameLbl, nameFld, sayHelloBtn);

        Scene scene = new Scene(vBox);
        stage.setScene(scene);
        stage.setTitle("First JavaFX App");
        stage.show();

    }
}
```



2.3 – Unos imena u tekst-kontrolu



2.4 – Alert dijalog sa prikazanom porukom

Pitanje

Kontrole Java FX-a se nalaze direktno na:

- a) **sceni**
- b) pozornici

Scena se kreira korišćenjem klase Scene, koja se nalazi u javafx.scene paketu. Stage može sadržati jednu scenu, a scena veći broj vizuelnih elemenata, odnosno kontrola.

Rezime

- Od verzije 8 Java programskog jezika Java FX je sastavni deo JRE i JDK-a, praktično nije potrebno nikakvo dodatno preuzimanje ili podešavanje.
- Java FX funkcionalnosti nalaze se zapakovane u JAR arhivu pod nazivom jfxrt.jar, koji se nalazi na putanji jre\lib\ext JDK foldera.
- Osnovana klasa Java FX aplikacije mora da nasledi klasu Application iz paketa javafx.application.
- Metoda **start** je ulazna tačka svake Java FX aplikacije.
- Metoda start prihvata jedan parametar tipa Stage, što je primarni stage aplikacije.
- Stage se koristi za prikaz scene.
- Scena se sastoji iz vizuelnih elemenata kao što su tekst, oblici, slike, kontrole, animacije i efekti, sa kojima korisnik može vršiti interakciju.
- Scena se kreira korišćenjem klase Scene, koja se nalazi u javafx.scene paketu.
- Sadržaj scene je organizovan u strukturu stabla. Na vrhu takve strukture je koreni element ili čvor.
- Kolekcija potomaka scene se preuzima metodom getChildren.

- Scena se dodeljuje pozornici metodom setScene kojoj se prosleđuje instanca Scene objekta.

LINKgroup