

Uputstvo za kreiranje JavaFX aplikacije korišćenjem Scene Builder-a

FXML i SceneBuilder

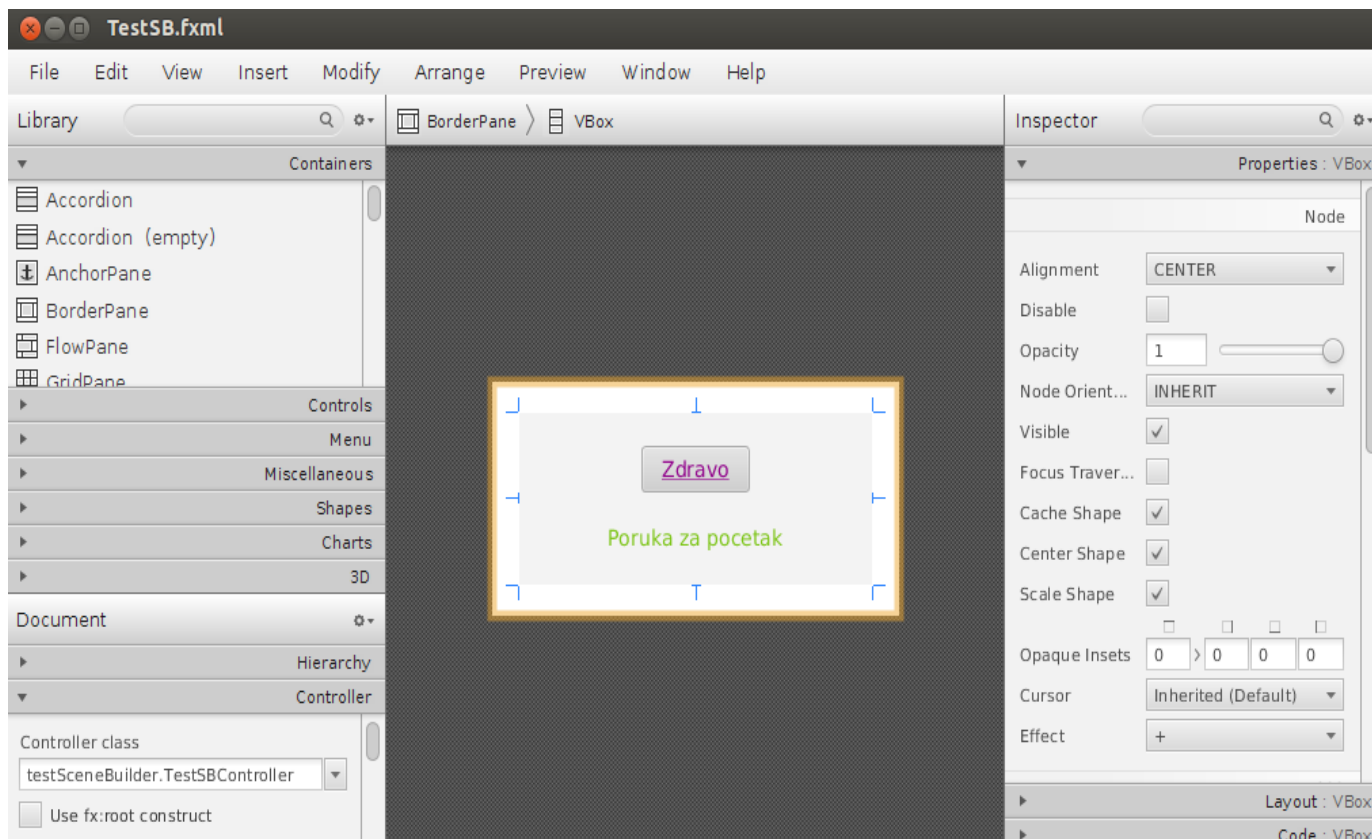
Scene Builder generiše izvorni kod FXML dokumenta kako se definiše grafički korisnički interfejs za JavaFX aplikaciju.

Scene Builder omogućava da se brzo i jednostavno napravi prototip interaktivne aplikacije koja povezuje vizuelne komponente sa logikom aplikacije.

JavaFX Scene Builder je alat za dizajniranje JavaFX aplikacija.

Omogućava jednostavno prevlačenje (drag-and-drop) i pozicioniranje komponenata grafičkog korisničkog interfejsa na JavaFX scenu. Kako se gradi scena, tako se automatski generiše odgovarajući FXML dokument.

Osnovni prozor JavaFX Scene Builder-a



Osnovni prozor sastoji se iz nekoliko panela:

- Library Panel - omogućava izbor, prevlačenje i pozicioniranje komponente u centralnu radnu površinu (Content Panel).
- Document Panel - sadrži hijerarhijski prikaz čvorova grafa scene. Klikom na strelicu sa desne strane može se izabrati način prikaza hijerarhije. Takođe, sadrži i informacije o Controller klasi.
- Inspector Panel - podeljen je na tri sekcije: Properties, Layout i Code

Primer: testSceneBuilder

Pravljenje FXML dokumenta

Napraviti JavaFX projekat:

New -> Other... -> JavaFX Project

U polje Name uneti ime projekta (test). Nakon toga napraviti odgovarajući paket (testSceneBuilder) i u njemu FXML dokument:

New -> Other... -> JavaFX -> New FXML Document

U polje Name uneti ime dokumenta (TestSB) i postaviti Root Element (koreni čvor grafa scene), za njega izabrati BorderPane.

Otvaranje FXML dokumenta pomoću Scene Builder-a - iz kontekstnog menija za TestSB.fxml fajl (nakon desnog klika) izabrati opciju:

Open with SceneBuilder

U Hierarchy panelu selektovati BorderPane, a u Inspector panelu izabrati sekciju Layout i postaviti:

- Pref Width: 300, Pref Height: 150
- Padding: 10 (sve vrednosti)

Izabrati VBox i prevući ga u centralno područje. Selektovati VBox u Hierarchy panelu. U sekciji Layout Inspector panela postaviti:

- Margin: 10 (sve vrednosti)
- Spacing: 20

U sekciji Properties postaviti:

- Alignment: CENTER (komponente koje će biti dodate u VBox biće centrirane)

Izabrati Button i prevući ga u VBox. Selektovati Button u Hierarchy panelu.

U sekciji Properties Inspector panela postaviti:

- Text: Zdravo
- Font: klik na strelicu
 - family: Bookman Old Style
 - style: Bold Italic
 - size: 14
- Text Fill: web: #990093 (ili bilo koju drugu boju)
- čekirati Underline

Izabrati Label i prevući je u VBox. Selektovati Label u Hierachy panelu. U sekciji Properties Inspector panela postaviti:

- Text: Poruka za pocetak
- Font: klik na strelicu
 - family: Antique Olive
 - style: Italic
 - size: 14
- Text Fill: web: #83d017 (ili bilo koju drugu boju)

U sekciji Layout Inspector panela postaviti za labelu:

- Pref Width: 80, Pref Height: 30

Da bi se videle generisane komponente:

Preview → Show Preview in Window (Ctrl+P)

Sačuvati izmene i potom pogledati sadržaj FXML dokumenta u Eclipse okruženju. Može se uočiti kod koji je automatski generisan, postepeno, tokom izgradnje samog korisničkog interfejsa u Scene Builder-u.

Pravljenje Controller klase

Pre pravljenja klase neopodno je projektu dodati jar fajl za JavaFX:

Properties → Java Build Path → Add External JARS → jfxrt.jar

Napomena: jfxrt.jar možete pronaći u folderu koji se dobija prilikom instaliranja JavaFX Scene Builder-a.

U odgovarajućem paketu (testSceneBuilder) generisati Java klasu (TestSBController) koja implementira interfejs javafx.fxml.Initializable (postaviti ovo svojstvo prilikom pravljenja klase ili dodati odgovarajući implements naknadno u klasu).

U klasi obezbediti instancne članice za one komponente koje su potrebne u kodu za obradu događaja (u primeru su to dugme i labela).

Obezbediti i metode za obradu događaja u skladu sa zahtevima. U primeru postoje tri metoda (nazivi su opcioni, ali bi trebalo da ukazuju na vrstu događaja):

- `handleButtonAction(ActionEvent)`
- `handleOnMouseEntered(MouseEvent)`
- `handleOnMouseExited(MouseEvent)`

Potrebno je u Scene Builder-u postaviti klasu za kontrolu korenom elementu. Na taj način se Scene Builder-u omogućuje da pristupi nazivima Event Handler-a i deklariranih instancnih promenljivih.

Za `BorderPane`:

- Controller → Controller class

postaviti odgovarajuće ime klase za kontrolu (`testSceneBuilder.TestSBController`).

Takođe, za dugme i labelu treba postaviti vrednost atributa `fx:id` na ime odgovarajuće instancne promenljive iz klase za kontrolu koja se na njega/nju odnosi.

- Inspector panel → Code → `fx:id`
 - `fx:id` : dugme (za dugme)
 - `fx:id` : labela (za labelu)

Potrebno je postaviti vrednosti za događaje koji se vezuju za dugme na imena odgovarajućih metoda:

- Code → On Action → `handleButtonAction`
- Code → On Mouse Entered → `handleOnMouseEntered`
- Code → On Mouse Exited → `handleOnMouseExited`

Najčešće je dovoljno da se rukovanje događajima definiše na predstavljeni način. Ako je potrebno da se postigne veća kontrola nad ponašanjem kontrolera i elemenata kojima on upravlja, može da se definiše metod:

```
public void initialize();
```

koji se poziva jednom, prilikom implementacije kontrolera, kada su sadržaji FXML dokumenata koji su mu pridruženi u potpunosti učitani, čime se omogućava implementiranoj klasi da izvrši post-procesiranje nad sadržajem, da pristupi resursima koji su korišćeni pri učitavanju dokumenata ili lokacijama koje su korišćene za rešavanje relativnih putanja u dokumentu.

Ukoliko se u klasi za kontrolu polja članovi i metodi za upravljanje događajima označe kao `private` ili `protected`, neophodno je označiti ih obeležjem `javafx.fxml.FXML (@FXML)`, kako bi FXML dokument mogao da im pristupi ili da ih menja.

Pravljenje main klase za testiranje

Klasa `FXMLLoader` je odgovorna za učitavanje FXML izvornog dokumenta i vraćanje dobijenog grafa scene:

```
Parent root = FXMLLoader.load(getClass().getResource("TestSB.fxml"));
```