

7. Sedma laboratorijska vježba

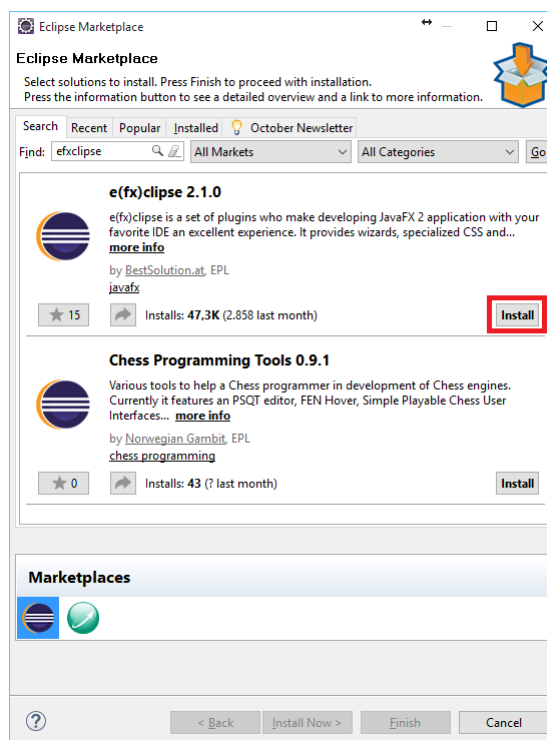
7.1. JavaFX

Svrha laboratorijske vježbe je dizajniranje ekrana korisničkog sučelja pomoću JavaFX tehnologije. Aplikacija mora podržavati funkcionalnosti koje su navedene u ovom videu: https://www.youtube.com/watch?v=8LTecUy_Jxg.

7.2. Zadatak za pripremu

Nastaviti razvoj aplikacije iz šeste laboratorijske vježbe i oblikovati ekrane grafičkog sučelja aplikacije prema koracima u nastavku.

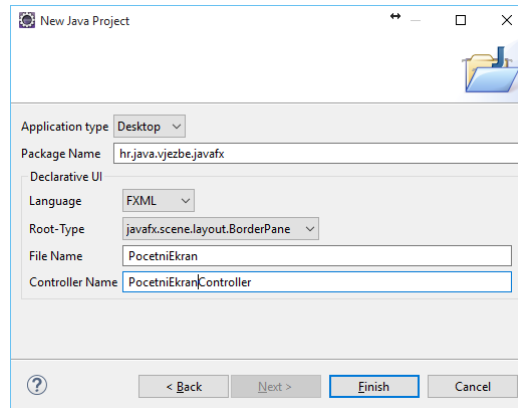
1. Sa stranice <http://www.oracle.com/technetwork/java/javase/downloads/javafxscenebuilder-info-2157684.html> preuzeti „Scene Builder 2.0“ i instalirati ga (na laboratorijskim vježbama je to već obavljeno).
2. Instalirati dodatak „e(fx)clipse“ u razvojno okruženje Eclipse korištenjem opcije „Help->Eclipse Marketplace...“, upisom „efxclipse“ u „Find:“ polje i pritiskom na gumb „Install“. Nakon dijaloga na slici 1. pojavljuje se ekran za potvrdu instalacije, odabrati prihvaćanje licence te pričekati da proces instalacije završi i potvrditi ponovno pokretanje Eclipsea.



Slika 1. Dijalog za instalaciju dodatka „efxclipse“

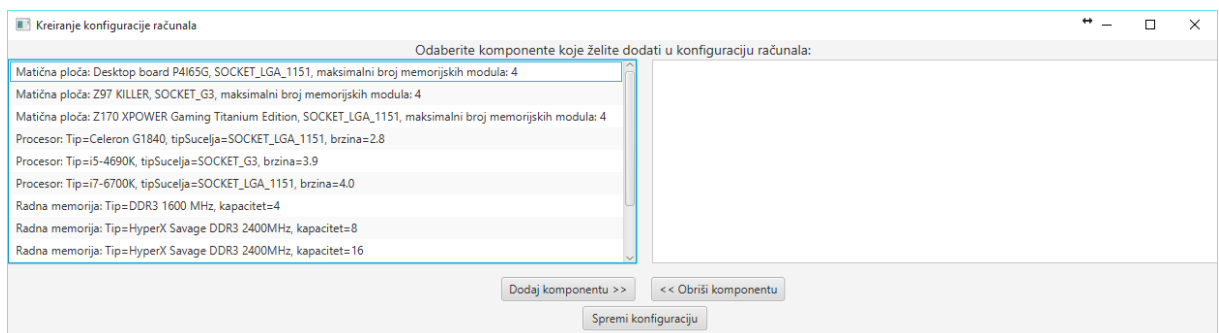
3. Kreirati novi „JavaFX“ projekt korištenjem opcije „File->New->JavaFX“ i nazvati ga prema vlastitom prezimenu sa sufiksom „-7“. Pritiskom na tipku „Next“ (dva puta) prikazuje se dijalog na kojem je potrebno definirati naziv paketa i vrstu aplikacije

prema slici 2:



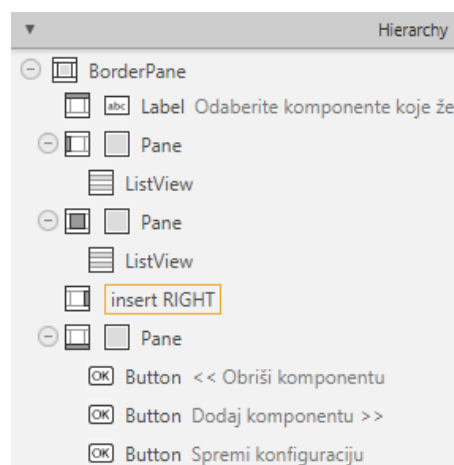
Slika 2. Dijalog za definiranje parametara JavaFX projekta

4. Pretvoriti JavaFX projekt u „Maven Project“, prebaciti „pom.xml“ datoteku i sve ostale resurse kako bi se mogli učitavati podaci iz datoteka i prikazivati na ekranu.
5. Korištenjem Scene Buildera otvoriti „FXML“ datoteku (korištenjem desne tipke miša i odabirom opcije „Open with Scene Builder“) i dizajnirati ekran aplikacije na način da izgleda ovako:



Slika 3. Izgled ekrana aplikacije

Prijedlog rasporeda i hijerarhije izgleda ovako:



Slika 4. Hijerarhija komponenti

6. U klasi „PocetniEkranController“ implementirati dohvat podataka o komponentama iz datoteka i popunjavanje u komponentu „ListView“ koja se nalazi na lijevoj strani. Osim toga implementirati funkcionalnost prebacivanja komponenti s lijeve strane na desnu stranu pritiskom na gumb „Dodaj komponentu >>“ te brisanje komponenti s desne strane pritiskom na gumb „<< Obriši komponentu“.
7. Prilikom svakog dodavanja komponenti potrebno je obaviti sljedeće provjere i ispisati eventualnu poruku o pogrešci:
 - 1) Konfiguracija može imati samo jednu matičnu ploču i procesor.
 - 2) Konfiguracija može imati samo matičnu ploču i procesor koji su kompatibilni po pitanju sučelja.
 - 3) Konfiguracija može maksimalno imati onoliko modula radne memorije koliko je definirano matičnom pločom.
 - 4) Konfiguracija mora imati barem po jednu komponentu od svake vrste: matične ploče, procesora, radne memorije i tvrdog diska.Validacije se moraju izvršavati bez obzira na poredak dodavanja komponenti u konfiguraciju (npr. radna memorija i matična ploča, procesor i matična ploča itd.).
8. Implementirati spremanje podataka o odabranoj konfiguraciji u tekstualnu datoteku „konfiguracija.txt“ na pritisak gumba „Spremi konfiguraciju“.

MOGUĆNOSTI UNAPREĐENJA ZADATAKA:

1. Proširiti rješenje na način da se na grafičkom sučelju također ispisuju i informacije koliki je minimalan broj svake komponente koji se mora dodati na grafičko sučelje kako bi se kreirala funkcionalna konfiguracija računala.
2. Proširiti rješenje na način da se na lijevoj strani aplikacije komponente koje su odabrane označe drugom bojom.

NAPOMENE:

1. Nakon svake promjene u „Scene Builder“ okruženju napraviti „Refresh“ projekta unutar Eclipse razvojnog okruženja.
2. Dijaloge koji prikazuju pogreške i poruke korisniku moguće je implementirati na sljedeći način:

```
Alert alert = new Alert(AlertType.ERROR);
alert.setTitle("Pogreška");
alert.setHeaderText("Matična ploča već postoji");
alert.setContentText("Već ste odabrali matičnu ploču!");
alert.showAndWait();
```

3. Podatke o konfiguraciji računala moguće je u datoteku spremiti korištenjem jedne linije programskog koda, ako je implementirana „toString” metoda koja sprema podatke o svim komponentama računala:

```
Files.write(Paths.get("./konfiguracija.txt"),  
racunalo.toString().getBytes());
```