

7. Sedma laboratorijska vježba

7.1. Tema vježbe

Svrha laboratorijske vježbe uvođenje grafičkog sučelja temeljenog na JavaFX tehnologiji koja će korisnika omogućavati upravljanje aplikacijom korištenjem unosnih polja, gumbiju i izbornika, umjesto konzole.

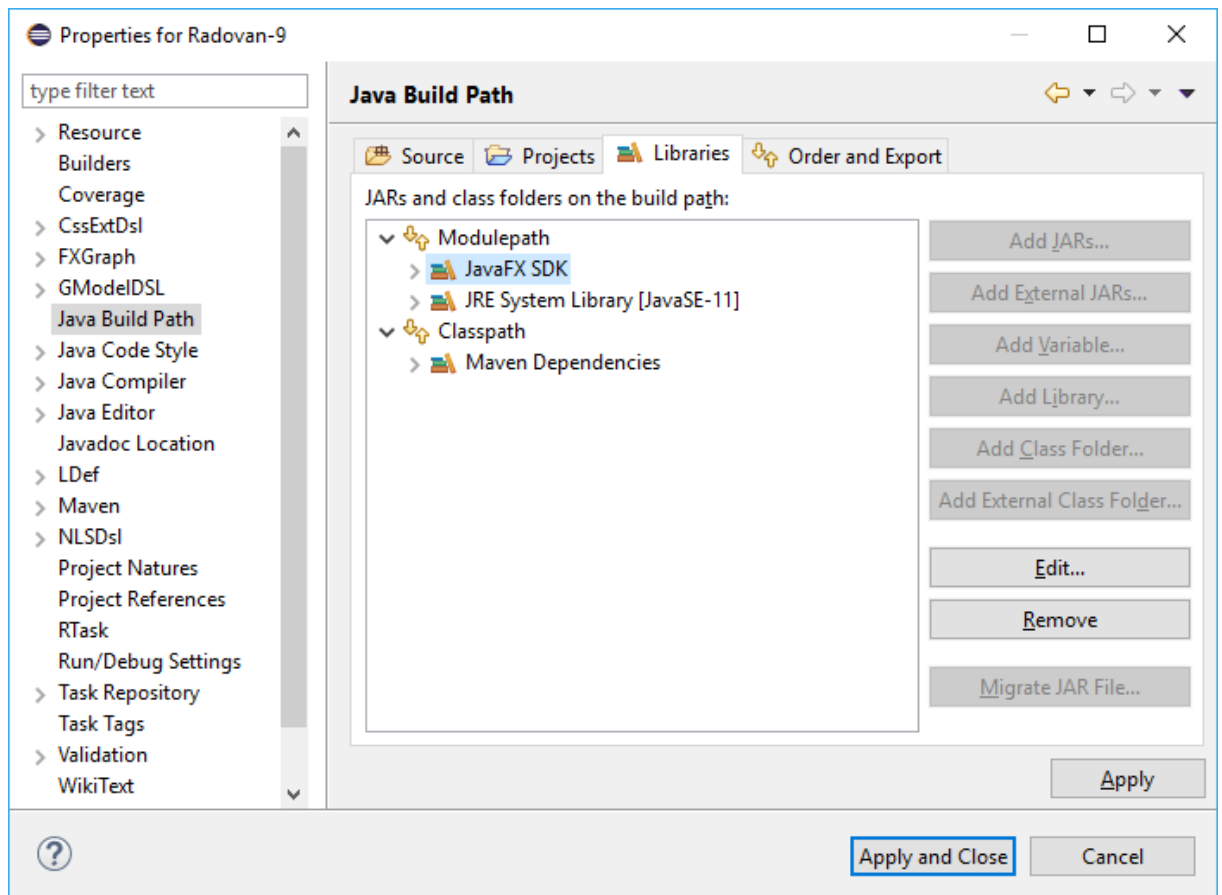
7.2. Zadatak za pripremu

Proširiti rješenje šeste laboratorijske vježbe na način kako je opisano u sljedećim koracima:

1. Korištenjem stranice <https://gluonhq.com/products/scene-builder/> preuzeti Scene Builder aplikaciju (koristiti posljednju verziju i opciju „Windows Installer“, može i za Javu 10). Na računalima na kojima se izvode laboratorijske vježbe je instalacija već obavljena.
2. Korištenjem opcije „Help->Eclipse Marketplace“ dohvatiti i instalirati posljednju verziju dodatka „e(fx)clipse“ koji omogućava kreiranje JavaFX projekata unutar razvojnog okruženja Eclipse.
3. Kreirati novi JavaFX projekt te ga nazvati po uzoru na ostale projekte iz prošlih laboratorijskih vježbi, npr. „Kovacevic-7“. Koristiti upute s Youtube videa <https://www.youtube.com/watch?v=BB2yig1bwAg>. Kod odabira verzije Jave koristiti Java 9 inačicu radi izbjegavanja mogućih problema s Javom 11.
4. Pretvoriti projekt u „Maven projekt“ i prebaciti ovisnosti o „logback“ biblioteci iz prošlih laboratorijskih vježbi. Po potrebi nad projektom izvršiti akciju „Maven -> Update Project“. Unutar „pom.xml“ datoteke potrebno je sljedeće „dependency“ zapise:

```
<dependency>
  <groupId>org.openjfx</groupId>
  <artifactId>javafx-controls</artifactId>
  <version>11.0.1</version>
</dependency>
<dependency>
  <groupId>org.openjfx</groupId>
  <artifactId>javafx-fxml</artifactId>
  <version>11.0.1</version>
</dependency>
```

5. Korištenjem opcije „Java Build Path“ potrebno je maknuti „JavaFX SDK“ pritiskom na gumb „Remove“ kao što je prikazano na slici 1.



Slika 1. Prikaz „Java Build Path“ konfiguracije projekta

6. Korištenjem stranice <https://gluonhq.com/products/javafx/> preuzeti arhivu „JavaFX Windows SDK“ (kao što je prikazano na slici 2) te je raspakirati. Inačica JavaFX SDK okruženja mora odgovarati inačici JDK-a pa je po potrebi moguće preuzeti i verziju 11.0.2.

JavaFX

The JavaFX 11 runtime is available as a platform-specific SDK, as a number of jmods, and as a set of artifacts in maven central.

The OpenJFX page at openjfx.io is a great starting place to learn more about JavaFX 11.

The Release Notes for JavaFX 11 are available in the OpenJFX GitHub repository: [Release Notes](#).

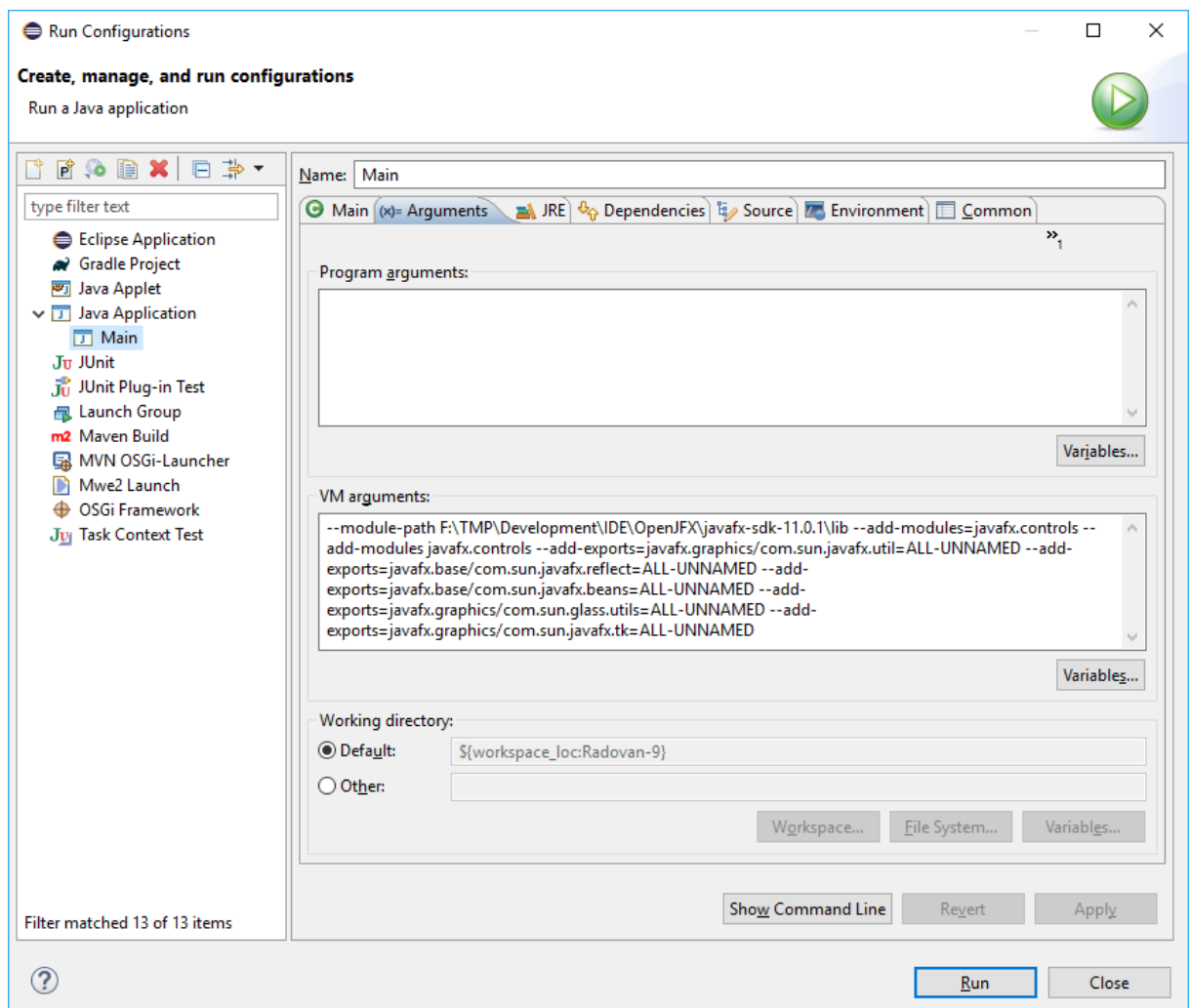
This software is licensed under GPL v2 + Classpath (see <http://openjdk.java.net/legal/gplv2+ce.html>).

Product	Version	Platform	Download
JavaFX Windows SDK	11.0.1	Windows	Download [SHA256]
JavaFX Windows jmods	11.0.1	Windows	Download [SHA256]
JavaFX Mac OS X SDK	11.0.1	Mac	Download [SHA256]
JavaFX Mac OS X jmods	11.0.1	Mac	Download [SHA256]
JavaFX Linux SDK	11.0.1	Linux	Download [SHA256]
JavaFX Linux jmods	11.0.1	Linux	Download [SHA256]

Slika 2. Stranica za preuzimanje JavaFX Windows SDK okruženja

7. Pokrenuti aplikaciju korištenjem opcije „Run As -> Java Application“ kako bi se kreirala „Run Configuration“ konfiguracija (bez obzira što kod pokretanja Eclipse može dojaviti pogrešku). Nakon toga korištenjem opcije „Run As -> Run Configurations...“ otvoriti dijalog za kreiranje konfiguracije za pokretanje aplikacije, otvoriti „Arguments“ tab i u polje „VM arguments“ (slika 3) postaviti sljedeće parametre, pri čemu putanja do „lib“ mape unutar raspakirane mape JavaFX SDK okruženja (iz šestog koraka) mora biti postavljena kako bi odgovarala lokaciji na lokalnom disku:

```
--module-path F:\TMP\Development\IDE\OpenJFX\javafx-sdk-11.0.1\lib  
--add-modules=javafx.controls  
--add-modules javafx.controls  
--add-exports=javafx.graphics/com.sun.javafx.util=ALL-UNNAMED  
--add-exports=javafx.base/com.sun.javafx.reflect=ALL-UNNAMED  
--add-exports=javafx.base/com.sun.javafx.beans=ALL-UNNAMED  
--add-exports=javafx.graphics/com.sun.glass.utils=ALL-UNNAMED  
--add-exports=javafx.graphics/com.sun.javafx.tk=ALL-UNNAMED
```



Slika 3. Konfiguracija za pokretanje JavaFX aplikacije

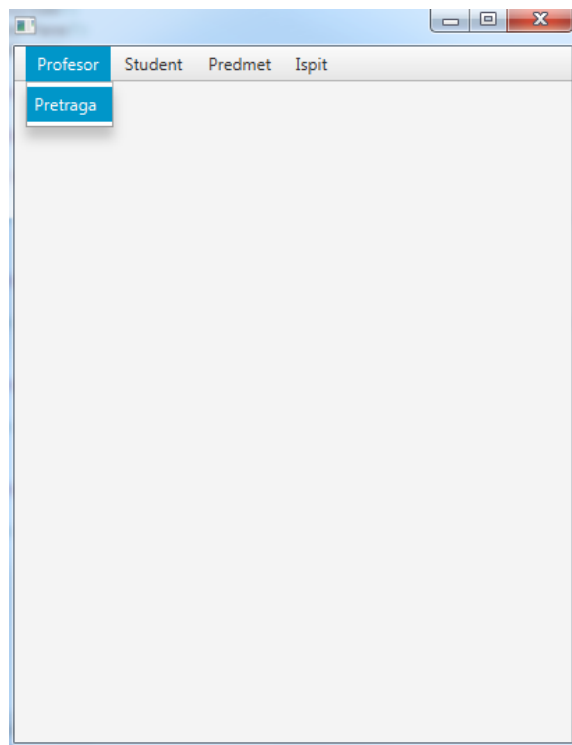
8. Iz prošle laboratorijske vježbe kopirati pakete „hr.java.vjezbe.entitet“, „hr.java.vjezbe.iznimke“ i „hr.java.vjezbe.sortiranje“.

9. Dizajnirati ekran koji će služiti kao početna stranica na kojoj se nalazi samo izbornik po uzoru na sliku 1. Svaki od izbornika mora uključivati glavne izbornike „Profesor“, „Student“, „Predmet“ i „Ispit“. Svaki od glavni izbornika mora uključivati podizbornik „Pretraga“. Za taj ekran potrebno je kreirati zasebnu „FXML“ datoteku i „Controller“ klasu. Otvaranje novih prozora unutar istog okvira aplikacije potrebno je implementirati na sljedeći način:

```
public void prikaziPretraguProfesora() {  
    BorderPane root;  
    try {  
        root = (BorderPane)FXMLLoader.Load(  
            getClass().getResource("profesori.fxml"));  
        Main.setMainPage(root);  
    } catch (IOException e) {  
        e.printStackTrace();  
    }  
}
```

Klasa „**Main**“ koja se generira uz pomoću Eclipsea kod kreiranja „JavaFX“ projekta mora otvarati ekran prikazivati na slici 1, ali i pomoću metode „setMainPage“ primati „BorderPane“ i postavljati sadržaj ekrana u istom okviru:

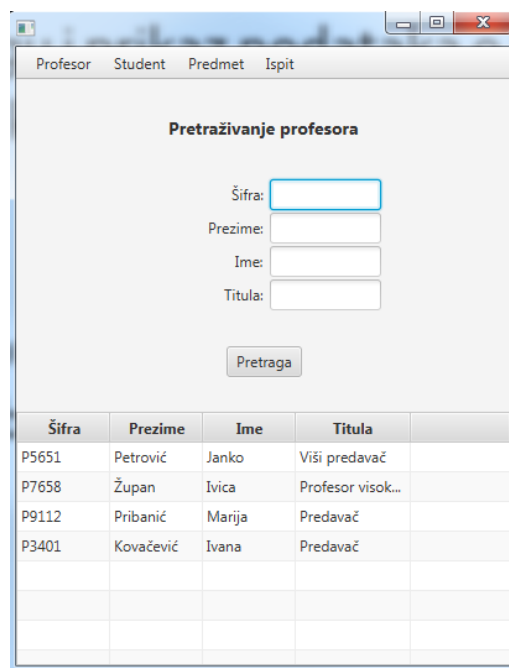
```
public static void setMainPage(BorderPane root) {  
    Scene scene = new Scene(root,400,500);  
    stage.setScene(scene);  
    stage.show();  
}
```



Slika 1. Početni ekran

10. Implementirati ekran za pretragu i prikaz podataka o profesorima kao što je prikazano na slici 2. Za ekran je potrebno koristiti „FXML“ datoteku i „Controller“ klasu. Taj ekran

mora se otvarati na odabir opcije „Profesor->Pretraga“. Aplikacija mora funkcionirati na način kao što je prikazano u video isječku <https://www.youtube.com/watch?v=mMi4BjVM2n4>. Prilikom prikaza ekrana je odmah potrebno prikazati sve podatke o profesorima pročitane iz datoteke. Logiku čitanja sadržaja datoteka potrebno je preuzeti iz prethodne laboratorijske vježbe i prebaciti u statičke metode unutar klasi „Datoteke“ koja će se nalaziti unutar novog paketa „hr.java.vjezbe.util“. Svaka metoda koja se nalazi u klasi mora vraćati listu objekata koje kreira, npr. „List<Profesor> ucitajDatotekuProfesora() {...}“. Metoda za dohvrat podataka mora se pozivati iz metode „initialize“ unutar „Controller“ klase.



Šifra	Prezime	Ime	Titula
P5651	Petrović	Janko	Viši predavač
P7658	Župan	Ivica	Profesor visok...
P9112	Pribanić	Marija	Predavač
P3401	Kovačević	Ivana	Predavač

Slika 2. Ekran za prikaz i pretragu profesora

11. Implementirati ekran za pretragu i prikaz podataka o studentima kao što je prikazano na slici 3. Za ekran je potrebno koristiti „FXML“ datoteku i „Controller“ klasu. Taj ekran mora se otvarati na odabir opcije „Student->Pretraga“ i ponašati se slično kao i ekran za pretragu profesora kako je i prikazano u video isječku <https://www.youtube.com/watch?v=mMi4BjVM2n4>.

JMBAG	Prezime	Ime	Datum rođenja
239847293847	Jurinić	Branko	29.12.1999.
309848034345	Dupan	Mirko	13.11.2000.
305939089081	Jelavić	Martina	10.09.1999.
405968405689	Marković	Petra	05.07.1999.

Slika 3. Ekran za prikaz i pretragu studenata

Za unos datuma je potrebno koristiti komponentu „DatePicker“, a „TableColumn“ komponentu je moguće konfigurirati na sljedeći način:

```
@FXML
private TableColumn<Student, String> datumRodjenjaTableColumn;

...

datumRodjenjaTableColumn.setCellValueFactory(
    new Callback<TableColumn.CellDataFeatures<Student, String>,
        ObservableValue<String>>() {
        @Override
        public ObservableValue<String> call(
            TableColumn.CellDataFeatures<Student, String> student) {
            SimpleStringProperty property = new SimpleStringProperty();
            DateTimeFormatter formatter =
                DateTimeFormatter.ofPattern("dd.MM.yyyy.");
            property.setValue(
                student.getValue().getDatumRodjenja().format(formatter));
            return property;
        }
    });
```

- Na isti način kao i prethodna dva ekrana potrebno je implementirati ekrane za pregled podataka o predmetima i ispitima. Za predmete je potrebno ispisivati podatke o šifri, nazivu, broju ECTS bodova i nositelju (ime i prezime profesora, pri čemu je potrebno provesti konverziju vrijednosti na sličan način kao i kod konverzije datuma). Za ispite je potrebno ispisivati naziv predmeta, ime i prezime studenta, ocjenu i datum s vremenom (konverzije složenih objekata kod ispisa u tablici je također potrebno implementirati slično kao u prošlom koraku).

NAPOMENA

Osim implementacija vježbe prema uputama, dozvoljeno je uvoditi i promjene ako su opravdane i ne narušavaju koncepte objektno-orijentiranog programiranja.