

## Zadatci za vježbu uz 10. tjedan predavanja / 2. dio labosa 4

### Zadatak 1.

Ovaj zadatak se nastavlja na zadatak 4 iz 1. dijela laboratorija 4. Potrebno je proširiti bazu podataka s dvije dodatne tablice: *predmeti* i *upisani predmeti*. Svaka tablica je spremljena u svojoj datoteci (npr. *studenti.txt*, *predmeti.txt* i *upisaniPredmeti.txt*). Implementacije smjestite u paket *hr.fer.oop.topic10.db*. U repozitoriju Ferka pronaći ćete primjerke tih datoteka. Datoteka *studenti.txt* je ista ona datoteka iz zadatka 4. Datoteka *predmeti.txt* predstavlja predmete i ima sljedeće atribute: *courseID*, *courseName*. Datoteka *upisaniPredmeti.txt* predstavlja relaciju između studenta i predmeta (koji je upisao) i ocjene te ima sljedeće atribute: *courseID*, *studentJMBAG*, *grade*. Napravite klase ekvivalentne klasama iz zadatka 4: *CourseDatabase*, *CourseRecord*, *EnrolmentDatabase* i *EnrolmentRecord*. Objekti iz klase *CourseRecord* su jednaki ako im je *courseID* jednak, a iz klase *EnrolmentRecord* ako su im *courseID* i *studentJMBAG* jednaki. Indekse u klasama *\*Database* napravite prema tome što je potrebno za daljnje korištenje u programu. Klase *\*Database* imaju sljedeće metode za pretraživanje i promjenu podataka:

```
public class CourseDatabase {
    ...
    public CourseRecord findById(String courseID) {...}

    // case insensitive searching with * e.g. "objektno*"
    public CourseRecord findFirstByName(String partialCourseName) {...}
}

public class EnrolmentDatabase {
    ...
    public Collection<EnrolmentRecord> findByStudent(String studentJMBAG) {...}
    public Collection<EnrolmentRecord> findByCourse(String courseID) {...}
    public EnrolmentRecord findByStudentAndCourse(String studentJMBAG,
        String courseID) {...}

    public EnrolmentRecord newCourse(String studentJMBAG,
        String courseID) {...} // default grade is 0 - not passed
    public void updateEnrolment(EnrolmentRecord record) {...}
    public void deleteRecord(String studentJMBAG, String courseID) {...}
}
```

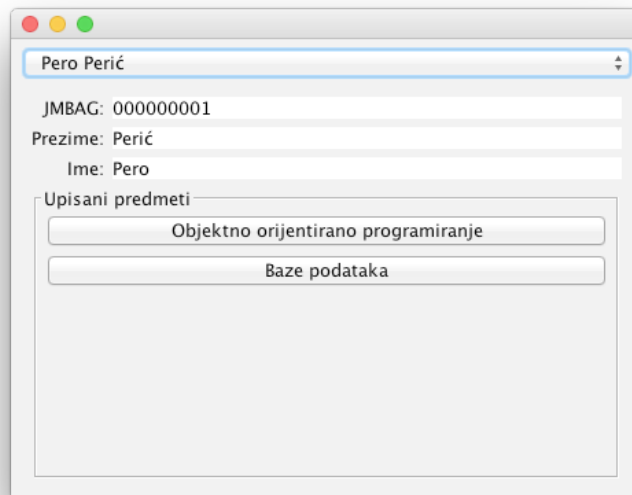
Napravite klasu *Database* kojoj u konstruktoru predajete nazive svih triju datoteka koje se učitaju u memoriju. Klasa *Database* mora imati sljedeće metode:

```
public class Database {
    public StudentDatabase getStudentTable() {...}
    public CourseDatabase getCourseTable() {...}
    public EnrolmentDatabase getEnrolmentTable() {...}
    ...
}
```

Napravite klasu koja ima metodu *main* koja demonstrira dohvaćanje popisa studenata i ispisa, dohvaćanje popisa predmeta jednog studenta, stvaranje novog predmeta, upisivanje predmeta nekom studentu, postavljanje ocjene, brisanje upisa predmeta.

### Zadatak 2.

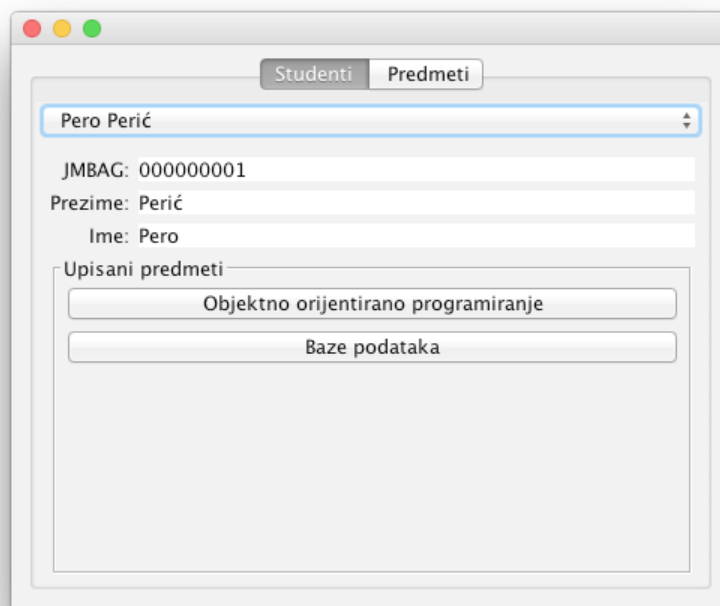
Napraviti grafičko sučelje koje izgleda kao na sljedećoj slici.

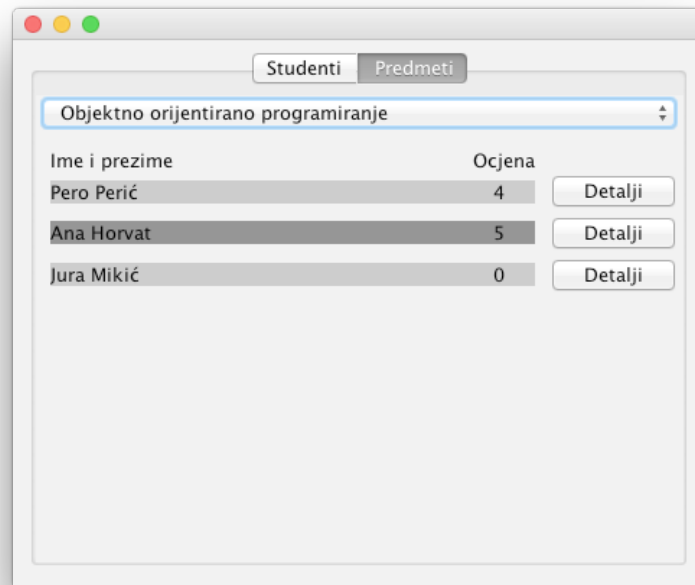


Klasa koja se pokreće je `StudentsFrame` koji prvo pročita podatke iz datoteka (baza podataka iz prošlog zadatka) i onda pokrene grafičko sučelje. Unutar prozora se nalazi klasa `StudentsPanel` (nasljeđuje `JPanel`) koji se sastoji od `JComboBox`-a na vrhu i ostatka u centralnom dijelu. U centralnom dijelu se nalaze informacije o studentu, a ispod toga koje je predmete upisao (gumbi). Kada se u `JComboBox`-u odabere drugi student, informacije u centralnom dijelu se trebaju ažurirati. Na klik na gumb predmeta otvara se novi prozor s porukom o tome kolika je ocjena studenta iz tog predmeta. Za to koristite klasu `JOptionPane` (više o tome na stranici: <http://docs.oracle.com/javase/tutorial/uiswing/components/dialog.html>).

### Zadatak 3.

Nadogradite grafičko sučelje iz prošlog zadatka tako da postoje dvije katicice (koristiti `JTabbedPane`) i koje izgleda kao na sljedećim slikama.





Kada se klikne na gumb *Detalji* nekog studenta potrebno se prebaciti u katicu *Studenti* i tamo prikazati podatke o studentu. Svi parni redovi trebaju biti u jednoj boji pozadine, a neparni drugoj boji pozadine.