

Evidencija prijava i odjava ispita

Popis predmeta
 Programsko inženjerstvo
 Analiza i razvoj programa

Rokovi za odabrani predmet
 1.2.2015.
 24.6.2015.
 2.7.2015.
 9.9.2015.

Datum prijave
 3.9.2015.

Datum odjave

Novi predmet

Novi rok
 9. rujna 2015.

Dodaj predmet

Dodaj rok

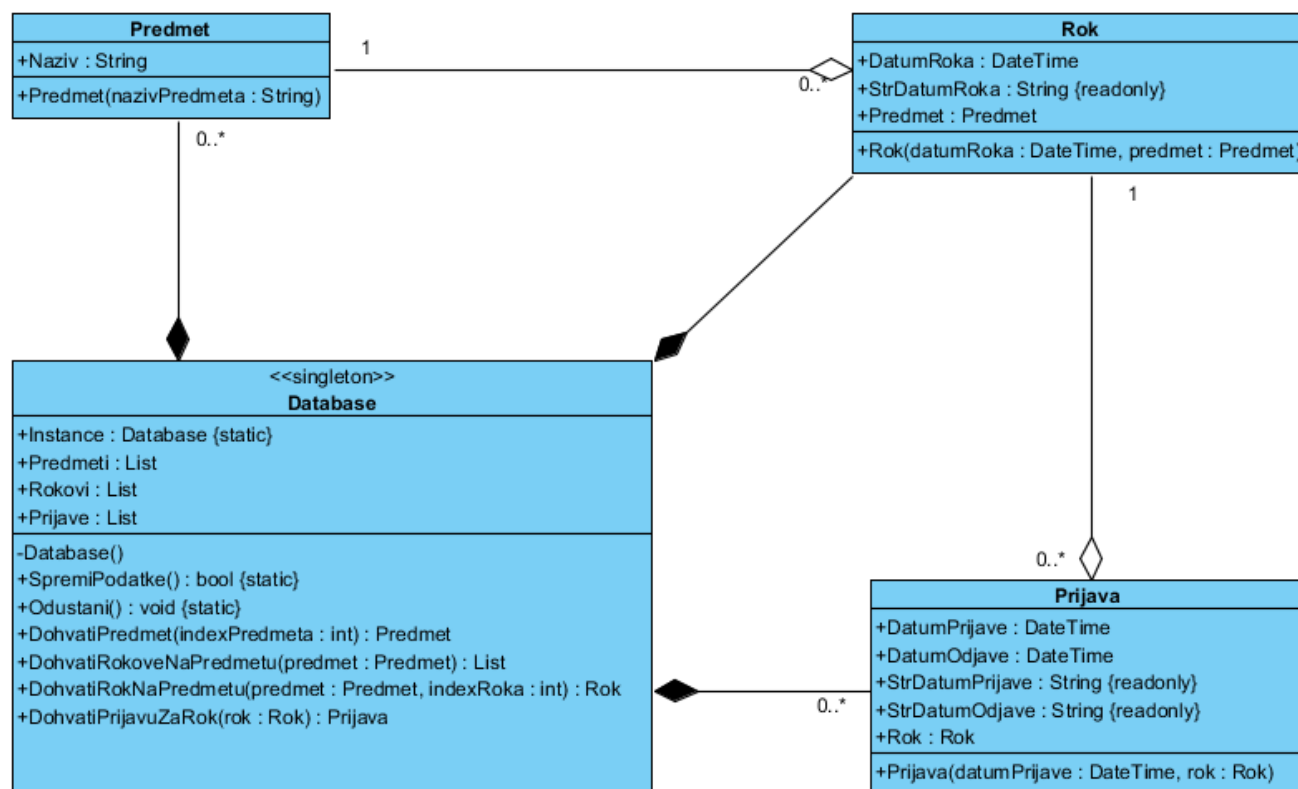
Prijavi ispit

Odjavi ispit

Spremi promjene

Odustani

Napraviti aplikaciju za evidenciju prijava i odjava fakultetskih ispita. Aplikacija se u potpunosti mora oslanjati na biblioteku *LocalDatabase.dll* koja omogućuje čitanje, unos i pohranu podataka. Za rješenje zadatka dovoljno je implementirati samo korisničko sučelje aplikacije (jednu klasu), budući da *LocalDatabase.dll* sadrži sve ostale klase i metode iz logičkog sloja i podatkovnog sloja. Dijagram klasa *LocalDatabase.dll* biblioteke prikazan je na slici ispod.



Jedno od mogućih rješenja specificirano je klasom prikazanom na slici ispod. Navedeno rješenje se može postići provođenjem sljedećih koraka:

1. Kreiranje novog projekta i glavne forme
2. Referenciranje na LocalDatabase.dll biblioteku
3. Uključivanje korištenja biblioteke u programski kod (**using**)
4. Popunjavanje popisa predmeta pri pokretanju aplikacije.
5. Evidencija trenutnog odabranog predmeta. **Prijedlog:** Koristiti metodu **DohvatiPredmet** iz klase **Database**.
Primjer:

```
private Predmet SelektiraniPredmet
{
    get
    {
        return Database.Instance.DohvatiPredmet(lstPredmeti.SelectedIndex);
    }
}
```

6. Popunjavanje popisa rokova za odabrani predmet pri promjeni selektiranog indeksa na popisu predmeta.
7. Evidencija trenutno odabranog roka. **Prijedlog:** Koristiti metodu **DohvatiRokNaPredmetu** iz klase **Database**, slično kao pod 5.
8. Implementacija dodavanja novog predmeta i novog roka. **Prijedlog:** Nakon dodavanja zapisa u bazu, iste metode mogu biti korištene za osvježavanje forme kao i za funkcionalnosti iz koraka 4. i 6.
9. Evidencija podataka o trenutnoj prijavi (ako postoji), te popunjavanje podataka o prijavi i odjavi roka pri promjeni selektiranog indeksa na popisu rokova za odabrani predmet.
Prijedlog: Za evidenciju vidi točke 5. i 7, te koristiti metodu **DohvatiPrijavuZaRok** iz klase **Database**. Za popunjavanje podatka o prijavi može se koristiti npr:

```
txtDatumPrijave.Text = SelektiranaPrijava != null ? SelektiranaPrijava.StrDatumPrijave : "";
```

10. Postaviti stanje gumba za prijavu i odjavu. **Prijedlog:** Za stanje se može koristiti sadržaj tekst polja koja prikazuju datume prijave i odjave, kao npr:

```
btnPrijavi.Enabled = txtDatumPrijave.Text == "";
```

11. Implementirati funkcionalnost prijave i odjave. **Prijedlog:** kod prijave kreirati novi objekt, a kod odjave koristiti postojeći objekt (**SelektiranaPrijava**) nastao pri koraku 9. Nakon dodavanja prijave i odjave, iste metode mogu biti korištene za popunjavanje podataka o prijavi i odjavi koje su pozivane pri promjeni selektiranog roka (vidi točku 9).
12. Implementirati funkcionalnosti spremanja promjena i odustajanja. **Prijedlog:** koristiti odgovarajuće statičke metode klase **Database**.

Frmevidencija
-SelektiranaPrijava : Prijava {readonly} -SelektiraniPredmet : Predmet {readonly} -SelektiraniRok : Rok {readonly}
+Frmevidencija() -OsvjeziPrikazPredmeta() : void -OsvjeziPrikazRokova() : void -OsvjeziPrikazPrijave() : void -btnDodajPredmet_Click() -btnDodajRok_Click() -btnPrijaviRok_Click() -btnOdjaviRok_Click() -btnSpremiPromjene_Click() -btnOdustani_Click() -lstPredmeti_SelectedIndexChanged() -lstRokovi_SelectedIndexChanged()