

## Elektronska Prijava Ispita

### Opis:

Korišćenjem Java RMI tehnologije implementirati sistem koji omogućava studentima da prijavljuju ispite putem korisničkog servisa elektronske studentske službe.

### Sistem treba da sadrži sledeće klase sa odgovarajućim metodama:

- **Server** - klasa koja sadrži logiku izvršavanja serverskog dela sistema.
- **Klijent** - klasa koja sadrži logiku izvršavanja klijentskog dela sistema.
- **Student** - interfejs klasa koja sadrži podatke o jednom studentu.
  - **Prijava vratiPrijavu ()** - metoda koja vraća ispitnu prijavu datog studenta.
  - **void prijaviIspit(String ispit)** - metoda koja dodaje ispit u ispitnu prijavu trenutnog studenta.
- **StudentImpl** - klasa koja implementira interfejs klase Student
  - **StudentImpl(String brIndeksa)** - konstruktor koji uzima jedinstveni broj indeksa.
- **Prijava** - interfejs klase koja treba da sadrži podatke o prijavljenim ispitima.
  - **String vratiIspite ()** - metoda koja vraća nazive ispita u obliku formatiranog teksta.
- **PrijavaImpl** - klasa koja implementira interfejs klase Prijava
- **EStudSluzbaManager** - interfejs klasa koja sadrži (bazu) niz Studenata.
  - **Student vratiStudenta (String brIndeksa)** - metoda koja vraća referencu na objekat Student za zadati broj indeksa.
- **EStudSluzbaManagerImpl** - klasa koja implementira interfejs klase **EStudSluzba**
- **EStudSluzbaCallback** - interfejs klasa koja treba da sadrzi callback metod.

### Sistem treba da prati sledeći scenario izvršavanja:

1. Vršiti se startovanje servera na kome se kreira instanca klase **EStudSluzbaManagerImpl**. U okviru klase **EStudSluzbaManagerImpl** se inicijalizuje niz od deset objekata tipa Student sa odgovarajućim brojevima indeksa.
2. Vršiti se startovanje klijenta u okviru koga se vrši konktovanje na server i referenciranje objekta tipa **EStudSluzbaManagerImpl**. U okviru klijenta se takođe vrši i registrovanje odgovarajućeg callback objekta.
3. Na klijentskoj strani se u okviru komandne linije ispisuje meni sa opcijama:
  - a) Prijava ispita
  - b) Provera prijavljenih ispita
  - c) Kraj

4. Na klijentskoj strani se u okviru komandne linije unosi odgovarajuća opcija iz menija (npr. a) nakon čega se u okviru komandne linije zahteva unos broja indeksa studenta praćeno porukama "Unesite broj indeksa". Korišćenjem metode << *EStudSluzba* >>::*vratiStudenta ()* se prvo preuzima referenca na odgovarajućeg Studenta a nakon toga se od korisnika traži unos naziva ispita koji se prijavljuje praćeno porukom "Unesite naziv ispita " i prosleđivanje unete vrednosti metodi << *Student* >>:: *prijaviIspit ()*.

5. Nakon prijave ispita svi registrovani klijenti preko callback metoda dobijaju informacije o nazivu ispita i broju studenata koji su prijavili taj ispit.

6. Ukoliko se iz menija odabere opcija c) u okviru komandne linije se zahteva unos broja korisnika koji se prosleđuje pozivu metode << *Student* >>:: *vratiPrijavu()* nakon čega se u komandnoj liniji štampaju svi prijavljeni ispiti za tog studenta.

7. Postupak se ponavlja sve dok korisnik ne izabere opciju Kraj iz menija.

#### Primer izlaza na klijentskoj strani:

Dobrodošli u korisnički servis studentske službe. Za nastavak izaberite opciju:

- a) Prijava ispita
- b) Provera prijavljenih ispita
- c) Kraj

>a

Izbrali ste opciju za prijavu ispita:

Unesite broj indeksa:

>9841

Unesite naziv ispita:

>DS

Dobrodošli u korisnički servis studentske službe. Za nastavak izaberite opciju:

- a) Prijava ispita
- b) Provera prijavljenih ispita
- c) Kraj

>b

Izbrali ste opciju za proveru prijavljenih ispita:

Unesite broj indeksa:

>9841

Prijavljeni ispiti su:

- 1. DS
- 2. AIP