

Mobilni Operater

Opis:

Korišćenjem Java RMI tehnologije implementirati sistem koji omogućava korisnicima kupovinu dodatnih minuta, poruka i interneta, kao i pregled stanja kod mobilnog operatera.

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.
- **Korisnik** - interfejs klasa koja sadrži informacije relevantne za jednog korisnika.
 - **void uplatiMinute(int minuti)** - metoda koja dodaje minute korisniku sa zadatim brojem i povećava racun u skladu sa tarifom.
 - **void uplatiPoruke(int poruke)** - metoda koja dodaje poruke korisniku sa zadatim brojem i povećava racun u skladu sa tarifom.
 - **void uplatiInternet(int internet)** - metoda koja dodaje internet korisniku sa yadatim brojem i povećava racun u skladu sa tarifom.
 - **Stanje vratiStanje()** - metoda koja vraća trenutno stanje korisnika.
- **KorisnikImpl** - klasa koja implementira interfejs klase Korisnik
 - **KorisnikImpl(String broj, int minuti, int poruke, int internet, int minutiTarifa, int porukeTarifa, int internetTarifa)** - konstruktor koji uzima broj telefona korisnika, pocetni broj minuta, pocetni broj poruka, pocetni broj megabajta interneta, tarifu dinara po minuti, tarifu dinara po poruci, tarifu dinara po megabajtu interneta
- **Stanje** - interfejs klasa koja sadrži jedno pitanje i ponuđene odgovore.
 - **int vratiMinute()** - metoda koja vraća broj preostalih minuta.
 - **int vratiPoruke()** - metoda koja vraća broj preostalih poruka.
 - **int vratiInternet()** - metoda koja vraća broj preostalih megabajta interneta.
 - **float vratiRačun()** - metoda koja vraća iznos treuntnog računa.
- **StanjeImpl** - klasa koja implementira interfejs klase Korisnik
 - **StanjeImpl(String broj, int minuti, int poruke, int internet, float račun)** - konstruktor koji uzima broj telefona korisnika, pocetni broj minuta, pocetni broj poruka i trenutni iznos računa,
- **OperaterManager** - interfejs klasa koja sadrži (bazu) niz korisnika.
 - **Korisnik vratiKorisnika(String broj)** - metoda koja vraća referencu na objekat tipa Korisnik za zadati broj telefona.
 - **void posaljiObavestenje(String poruka)** - metoda koja salje obavestenje svim registrovanim klijentima.
- **OperaterManagerImpl** - klasa koja implementira interfejs klase Operater
- **OperaterCallback** - interfejs klasa koja treba da sadrzi callback metode.

Sistem treba da prati sledeći scenario izvršavanja:

1. Vršiti se startovanje servera na kome se kreira instanca klase ***OperaterManagerImpl***. U okviru klase ***OperaterManagerImpl*** se inicijalizuje niz od deset objekata tipa Korisnik sa odgovarajućim brojevima telefona, minuta, poruka, interneta kao i tarifama.
2. Vršiti se startovanje klijenta u okviru koga se vrši konektovanje na server i referenciranje objekta tipa Operater kao i registrovanje Callback objekta.
3. Na klijentskoj strani se u okviru komandne linije ispisuje meni sa opcijama:
 - a) Administratorski mod
 - b) Korisnicki mod
4. Ukoliko je selektovana opcija "Administratorski mod" na klijentskoj strani se u okviru komandne linije ispisuje meni sa opcijama:
 - a) Slanje obavestenja
 - b) Izlaz
5. Na klijentskoj strani se u okviru komandne linije unosi odgovarajuća opcija iz menija (npr. a) nakon čega se u okviru komandne linije zahteva unos poruke i korišćenjem metode ***<<OperaterManagerImpl>>::posaljiObavestenje*** svim registrovanim klijentima puterm callback interfejsa salje obavještenje koje sadrži unetu poruku.
6. Ukoliko je selektovana opcija "Korisnicki mod" na klijentskoj strani se u okviru komandne linije ispisuje meni sa opcijama:
 - a) Uplata Minuta
 - b) Uplata Poruka
 - c) Uplata Interneta
 - d) Provera stanja
 - e) Kraj
7. 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 korisnika kao i broja minuta praćeno porukama "Unesite broj telefona: " i "Unesite broj minuta:". Korišćenjem metode ***<<Operater>>::vratiKorisnika()*** se prvo preuzima referenca na odgovarajućeg korisnika a zatim se pozivom metoda ***<<Korisnik>>::uplatiMinute()*** vrši dodavanje minuta.
8. Na klijentskoj strani se poziva metod ***<<Korisnik>>::vratiStanje()***, koja vraća Objekat Stanje na osnovu koga se štampaju informacije o trenutnom stanju zadatog korisnika.

Primer izlaza na klijentskoj strani:

Izaberite mod rada klijenta:

- a) Administratorski mod
- b) Korisnicki mod

/>b

Dobrodosli u korisnicki servis mobilnog operatera. Za nastavak izaberite opciju:

- a) Uplata Minuta
- b) Uplata Poruka
- c) Uplata Interneta
- d) Provera stanja
- e) Kraj

/>a

Izbrali ste opciju za uplatu dodatnih minuta:

Unesite broj telefona korisnika:

/>060123456

Unesite broj dodatnih minuta:

/>100

Dobrodosli u korisnicki servis mobilnog operatera. Za nastavak izaberite opciju:

- a) Uplata Minuta
- b) Uplata Poruka
- c) Uplata Interneta
- d) Provera stanja
- e) Kraj

/>d

Izbrali ste opciju za proveru stanja:

Unesite broj telefona korisnika:

>060123456

Vas racun iznosi:

500 din: