

eBanka

Opis:

Korišćenjem Java RMI tehnologije implementirati sistem koji omogućava korisnicima prebacivanje novca sa dinarskog na devizni račun i obrnuto po trenutno važećem kursu putem eBanka korisničkog servisa.

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 podatke jednog korisnika u banci.
 - *Stanje vratiStanje()* - metoda koja vraća stanje datog korisnika.
 - *void transferDinarskiNaDevizni(float iznos, float kurs)* - metoda koja prebacuje zadati iznos sa dinarskog na devizni račun.
 - *void transferDevizniNaDinarski(float iznos, float kurs)* - metoda koja prebacuje zadati iznos sa deviznog na dinarski račun.
- **KorisnikImpl** - klasa koja implementira interfejs klase Korisnik
 - *KorisnikImpl(String jbk, float iznosDinarski, float iznosDevizni)* - konstruktor koji uzima jedinstveni broj korisnika, inicijalni iznos na dinarskom racunu i inicijalni iznos na deviznom racunu
- **Stanje** - interfejs klase koja treba da sadrži podatke sa iznosima na dinarskom i deviznom računu.
 - *float vratiDinarskiIznos()* - metoda koja vraća iznos na dinarskom računu.
 - *float vratiDevizniIznos()* - metoda koja vraća iznos na deviznom računu.
- **StanjeImpl** - klasa koja implementira interfejs klase Stanje
 - *StanjeImpl(float iznosDinarski, float iznosDevizni)* - konstruktor koji uzima iznos na dinarskom racunu i iznos na deviznom racunu
- **EBankaManager** - interfejs klasa koja sadrži (bazu) niz korisnika.
 - *Korisnik vratiKorisnika(String jbk)* - metoda koja vraća referencu na korisnika za zadatim identifikatorom.
 - *void posaljiObavestenje(String poruka)* - metoda koja salje obavestenje svim registrovanim klijentima.
- **EBankaManagerImpl** - klasa koja implementira interfejs klase Banka
- **EBankaCallback** - interfejs klasa koja treba da sadrži callback metode.

Sistem treba da prati sledeći scenario izvršavanja:

1. Vršiti se startovanje servera na kome se kreira instanca klase **EBankaManagerImpl**. U okviru klase **EBankaManagerImpl** se inicijalizuje niz od deset objekata tipa Korisnik sa

odgovarajucim jedinstvenim brojevima korisnika, inicijalnim iznosima na dinarskom i deviznom računu.

2. Vršiti se startovanje klijenta u okviru koga se vrši konktovanje na server i referenciranje objekta tipa EBanka 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 odgovarajuca opcija iz menija (npr. a) nakon čega se u okviru komandne linije zahteva unos poruke i korišćenjem metode << **EBankaManagerImpl** >>:: **posaljiObavestenje** svim registrovanim klijentima puterm callback interfejsa salje obaveštenje koje sadrži unetu poruku.

6. Ukoliko je selektovana opcija "Korisnicki mod" na klijentskoj strani se u okviru komandne linije ispisuje meni sa opcijama:

- a) Transfer sa dinarskog na devizni račun
- b) Transfer sa deviznog na dinarski račun
- c) Provera stanja
- d) Kraj

7. Na klijentskoj strani se u okviru komandne linije unosi odgovarajuca opcija iz menija (npr. a) nakon čega se u okviru komandne linije zahteva unos broja korisnika kao i iznosa za transfer praćeno porukama "Unesite jedinstveni broj korisnika: " i "Unesite iznos:". Korišćenjem metode <<EBanka>>::vratiKorisnika() se prvo preuzima referenca na odgovarajućeg korisnika a zatim se pozivom metoda <<Korisnik>>::transferDinarskiNaDevizni() vrši odgovarajući transfer novčanih sredstava. .

8. Ukoliko se iz menija odabere opcija c) u okviru komandne linije se zahteva unos broja korisnika koji se prosleđuje pozivu metode << Korisnik >>:: vratiStanje() nakon čega se na podaci o stanju korisnika štampaju u komandnoj liniji.

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

Primer izlaza na klijentskoj strani:

Izaberite mod rada klijenta:

- a) Administratorski mod
- b) Korisnicki mod

/>b

Dobrodošli u eBank korisnicki servis. Za nastavak izaberite opciju:

- a) Transfer sa dinarskog na devizni račun
- b) Transfer sa deviznog na dinarski račun
- c) Provera stanja
- d) Kraj

>a

Izbrali ste opciju za Transfer sa dinarskog na devizni račun:

Unesite jedinstveni broj korisnika:

>JBK123456

Unesite iznos:

>100

Dobrodošli u eBank korisnicki servis. Za nastavak izaberite opciju:

- a) Transfer sa dinarskog na devizni račun
- b) Transfer sa deviznog na dinarski račun
- c) Provera stanja
- d) Kraj

>c

Izbrali ste opciju za proveru stanja:

Unesite jedinstveni broj korisnika:

>JBK123456

Vase stanje je:

Iznos na dinarskom računu: 120

Iznos na deviznom računu: 5000