

Napomena: Na Desktop-u napraviti direktorijum sa imenom oblika <Ime><Prezime> i njega izabrati za workspace pri pokretanju Eclipse. Projekat i paket nazvati isto tako.

Zadatak:

Napisati aplikaciju kojom se realizuju transakcije (uplate i isplate) nad računima u banci.

1. Klasa *Racun* opisuje račun u banci. Opis računa sadrži broj računa (String), oznaku valute (RSD ili EUR), tekuće stanje na računu, datum poslednje uplate i datum poslednje isplate. Potrebno je definisati klasu *Datum* za opis datuma. Takođe, opis računa sadrži i registrovane podatke o poslednjih 10 uplata, odnosno, poslednjih 10 isplata.
2. Klasa *OsnovniRacun* opisuje račun bez mogućnosti pozajmice, dok klasa *RacunSaPozajmicom* opisuje račun sa mogućom pozajmicom. Opis ove klase sadrži i podatak o vrednosti dozvoljenog minusa (pozajmice) i podatak o raspoloživom stanju na računu.

Definisati konstruktor koji formira račun sa inicijalnim stanjem 0, kao i konstruktor koji postavlja početno stanje na datu vrednost.

Obezbediti da se polimorfno izvršavaju metodi:

```
public void realizujUplatu(Uplata u) i  
public boolean realizujIsplatu(Isplata i)
```

Prilikom registrovanja nove transakcije (uplate ili isplate), ukoliko je već registrovano 10 transakcija tog tipa, nova transakcija postaje poslednja registrovana, a postojeće treba ažurirati na odgovarajući način.

3. Klasa *Uplata* opisuje uplatu na račun u banci. Opis klase sadrži podatak o računu na koji se uplata realizuje, sumu novca koja se uplaćuje na račun i datum uplate. Obezbediti metod za realizaciju uplate.
4. Klasa *Isplata* opisuje isplatu sa računa u banci. Opis klase sadrži podatak o računu sa kojeg se realizuje isplata, sumu novca koja se isplaćuje sa računa i datum isplate. Obezbediti metod za realizaciju isplate.

Prilikom ispisa podataka o računu, za datum poslednje uplate/isplate ispisati “—” ako nije bilo uplata/isplata.

Napisati test klasu u kojoj se sa ulaza učitava koliko računa je potrebno kreirati, nakon čega se unose podaci o računima. Tip računa se određuje na osnovu opisa koji se zadaje (“osnovni” ili “pozajmica”).

Potom se slučajno bira jedan račun i prikazuju se podaci o njemu.

Nakon toga se u petlji učitavaju podaci o transakcijama koje treba izvršiti nad izabranim računom sve dok korisnik ne unese “kraj”. Nakon svake izvršene transakcije treba prikazati podatke o računu, a ukoliko transakcija ne može da se realizuje i odgovarajuću poruku o tome. Na kraju prikazati sve realizovane uplate i sve realizovane isplate za izabrani račun.

Primer prikaza podataka za račun sa pozajmicom:

```
Broj racuna: 123-45  
Valuta: rsd  
Stanje: 10000.0  
Poslednja uplata: --  
Poslednja isplata: --  
Dozvoljeni minus: 30000.0  
Raspolozivo stanje: 40000.0
```

```
Tip transakcije:  
isplata  
Suma:  
20000  
dan: 10 mesec: 4 godina: 2010
```

```
Stanje na racunu nakon transakcije:  
Broj racuna: 123-45  
Valuta: rsd  
Stanje: -10000.0  
Poslednja uplata: --
```

```
Poslednja isplata: 10. 4. 2010.  
Dozvoljeni minus: 30000.0  
Raspolozivo stanje: 20000.0
```

```
Tip transakcije:  
isplata  
Suma:  
25000  
dan: 12 mesec: 4 godina: 2010
```

```
Isplata ne može da se realizuje!  
Stanje na racunu nakon transakcije:  
Broj racuna: 123-45  
Valuta: rsd  
Stanje: -10000.0  
Poslednja uplata: --  
Poslednja isplata: 10. 4. 2010.  
Dozvoljeni minus: 30000.0  
Raspolozivo stanje: 20000.0
```