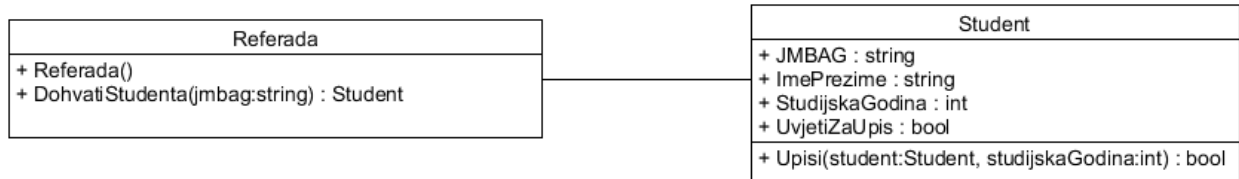


## ZADATAK

Programeri u vašem timu upravo su završili iteraciju razvoja komponente za provedbu upisa studenata u više studijske godine (.dll datoteka). Vi ste kao tester dobili zadatak da je testirate prema priloženim scenarijima.

Potrebno je kreirati novi **Solution** sa **Unit Test projektom**, dodati **referencu** na komponentu za upise studenata, te napisati jedinične (Unit) testove za klasu **Referada** i **Student**. Svaki od navedenih scenarija treba pokriti s posebnom testnom metodom (dakle, ne sve scenarije u jednu metodu).

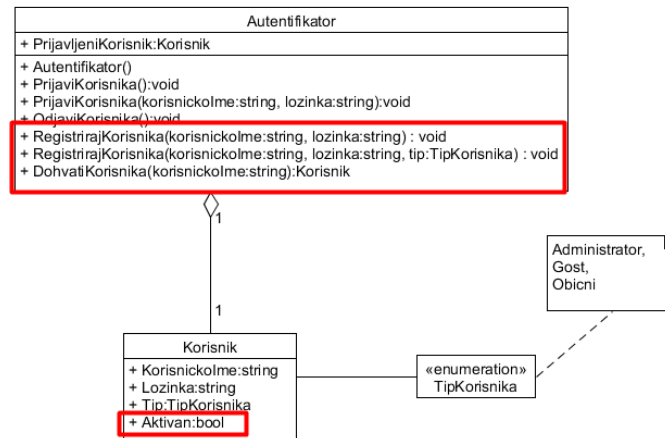


```
JMBAG = "11", ImePrezime = "Pero Perić", StudijskaGodina = 1, UvjetiZaUpis = true
JMBAG = "12", ImePrezime = "Ivan Perić", StudijskaGodina = 2, UvjetiZaUpis = false
JMBAG = "13", ImePrezime = "Matej Horvat", StudijskaGodina = 3, UvjetiZaUpis = false
JMBAG = "14", ImePrezime = "Ana Novak", StudijskaGodina = 4, UvjetiZaUpis = true
JMBAG = "15", ImePrezime = "Marija Horvat", StudijskaGodina = 3, UvjetiZaUpis = false
JMBAG = "16", ImePrezime = "Ivana Ivić", StudijskaGodina = 2, UvjetiZaUpis = true
```

### Scenariji koje treba pokriti testovima:

1. Ukoliko uz pomoć metode **DohvatiStudenta** klase **Referada** pokušamo dohvatiti studenta sa nepostojećim JMBAG-om (npr. **123**), metoda vraća **null**.
2. Ukoliko uz pomoć metode **DohvatiStudenta** klase **Referada** pokušamo dohvatiti studenta sa postojećim JMBAG-om (npr. **13**), metoda vraća objekt različit od **null**.
3. Ukoliko uz pomoć metode **Upisi** klase **Student** pokušamo upisati studenta (npr. **11**) u nepostojeću studijsku godinu (npr. **0**), metoda neće uspjeti izvršiti upise, te će baciti iznimku **ArgumentException**.
4. Ukoliko ima uvjete za upis, studenta je moguće upisati isključivo u sljedeću studijsku godinu, tj. nije moguće preskakati godine. Stoga, ukoliko uz pomoć metode **Upisi** klase **Student** pokušamo studenta **11** (trenutno 1. godina) upisati u 3. godinu, metoda neće uspjeti izvršiti upis, te će baciti iznimku **ArgumentException**.
5. Ukoliko nema uvjete za upis, student se ne može upisati u sljedeću studijsku godinu. Stoga, ukoliko uz pomoć metode **Upisi** klase **Student** pokušamo upisati studenta **12** (trenutno je 2. godina, ali nema uvjete za upis) u 3. godinu, metoda neće uspjeti izvršiti upis, te će izbaciti iznimku **ConditionsNotMetException**.
6. Ukoliko uz pomoć metode **Upisi** klase **Student** pokušamo upisati studenta **11** u studijsku godinu **2** (svi uvjeti su zadovoljeni), metoda će uspjeti izvršiti upis, vratiti će true, a student **11** mora biti na studijskoj godini **2** i uvjeti za potpis mu moraju biti resetirani na **false**.

Potrebno je u dobivenom Solution-u kreirati novi **Unit Test projekt**, dodati **referencu** na projekt za autentifikaciju, te napisati jedinične (Unit) testove za klasu **Autentifikator**. Svaki od navedenih scenarija treba pokriti s posebnom testnom metodom. Testne metode nazvati po redu kako su navedene: Test1, Test2, Test3... **Testovi koji su obojeni crvenom bojom će inicijalno pasti, pa treba napraviti ispravke/dorade u implementaciji kako bi testovi prošli.**



```

{ KorisnickoIme = "mmijac", Lozinka="abcde", Tip = TipKorisnika.Administrator });
{ KorisnickoIme = "pivic", Lozinka = "12345", Tip = TipKorisnika.Obicni });
{ KorisnickoIme = "valic", Lozinka = "ab234", Tip = TipKorisnika.Obicni });
{ KorisnickoIme = "anovak", Lozinka = "qetzs", Tip = TipKorisnika.Obicni });
{ KorisnickoIme = "thorvat", Lozinka = "hfdz", Tip = TipKorisnika.Obicni, Aktivan = false });
{ KorisnickoIme = "gtudor", Lozinka = "nrtgs", Tip = TipKorisnika.Obicni });
{ KorisnickoIme = "btomas", Lozinka = "kgdrt", Tip = TipKorisnika.Administrator });
{ KorisnickoIme = "gost", Lozinka = "gost", Tip = TipKorisnika.Gost });

```

#### Scenariji koje treba pokriti testovima:

**Scenarij 1.** Ako pozivom metode **DohvatiKorisnika** pokušamo dohvatiti nepostojećeg korisnika („pperic“), metoda vraća **null**.

**Scenarij 2.** Ako pozivom metode **DohvatiKorisnika** pokušamo dohvatiti **neaktivnog** korisnika („thorvat“), metoda vraća **null**.

**Scenarij 3.** Ako pozivom metode **DohvatiKorisnika** pokušamo dohvatiti aktivnog korisnika (npr. „anovak“), metoda vraća vrijednost koja nije null i vraćeni korisnik ima **KorisnickoIme** jednako „anovak“.

**Scenarij 4.** Ako pozovemo metodu **RegistrirajKorisnika** i proslijedimo joj **lozinku** = „abc“ (kraće od 5 znakova), tada metoda baca iznimku **IncorrectAuthenticationData**.

**Scenarij 5.** Ako pozovemo metodu **RegistrirajKorisnika** i proslijedimo joj samo ispravno **korisničko ime** i **lozinku** (npr. „pperic“, „123456“), tada autentifikator registrira novog korisnika, i mora postojati korisnik sa korisničkim imenom „pperic“, lozinkom „123456“, te tipom **obični korisnik**.

**Scenarij 6.** Ako pozovemo metodu **RegistrirajKorisnika** i proslijedimo joj **korisničko ime** = „btomas“ (već postojeće korisničko ime) i **lozinku** = „abc123“, tada metoda baca iznimku **IncorrectAuthenticationData**.

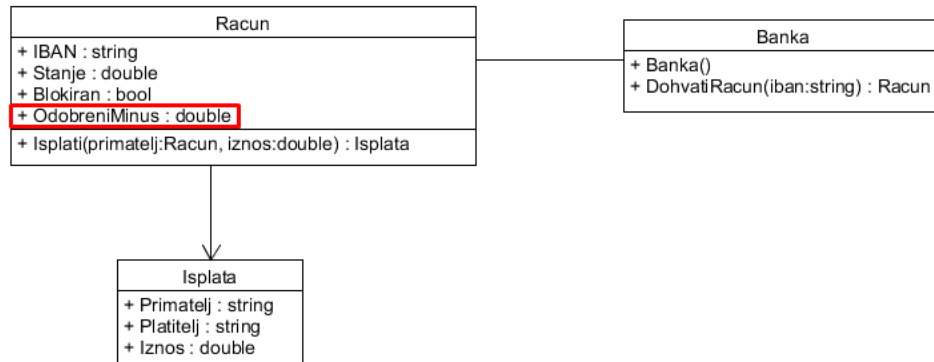
**Scenarij 7.** Ako pozivom metode **RegistrirajKorisnika** pokušam registrirati korisnika koji je administrator (npr. „pperic“, „123456“, **TipKorisnika.Administrator**), a prije toga nije već prijavljen neki drugi administrator, metoda baca iznimku **UnauthorizedRegistrationException**. (Registriranje novog administratorskog korisnika može obaviti samo administrator).

**Scenarij 8.** Ako pozivom metode **RegistrirajKorisnika** pokušam registrirati korisnika koji je administrator (npr. „pperic“, „123456“, **TipKorisnika.Administrator**), a prije toga jeste već prijavljen neki drugi administrator (npr. *mmijac*), metoda uspijeva registrirati korisnika, pa je moguće dohvatiti korisnika „pperic“ koji nije **null** i koji je **administrator**.

**Scenarij 9.** Ako pozovemo metodu **RegistrirajKorisnika** i proslijedimo joj ispravno **korisničko ime** i **lozinku**, te **tip Gost** (npr. „pperic“, „123456“, **TipKorisnika.Gost**). S obzirom da ne želimo da je moguće dodavati korisnike koji su tipa gost, autentifikator neće registrirati novog korisnika, tj. dohvaćanje korisnika „pperic“ će vratiti **null**.

**Scenarij 10.** Ako pozivom metode **RegistrirajKorisnika** i proslijeđivanjem korisničkih podataka za neaktivnog korisnika („thorvat“, „hfdz“), ne registrira se novi korisnik, nego se postojeći korisnik „thorvat“ aktivira (**Aktivan = true**). To znači da dohvaćanjem korisnika „thorvat“ dobijemo korisnika koja nije null, i koji je aktivan (**Aktivan == true**).

U dobivenom **Solution-u** kreirati novi **Unit Test projekt**, dodati **referencu** i **using** izraz na postojeći projekt **Financije**, te napisati jedinične (Unit) testove za klase **Racun** i **Banka**. Svaki od navedenih scenarija treba pokriti s posebnom testnom metodom (dakle, ne sve scenarije u jednu metodu). Testne metode nazovite **Test1**, **Test2**, **Test3...**, po redu kako su navedeni scenariji. **Testovi koji su obojeni crvenom bojom će inicijalno pasti, pa treba napraviti ispravke/dorade u implementaciji Financija kako bi testovi prošli.**



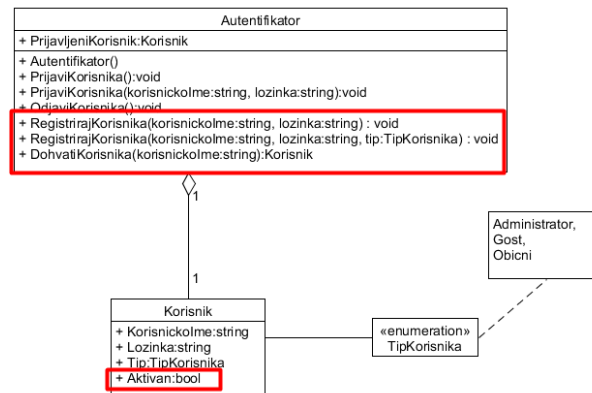
```

Racuni.Add(new Racun { IBAN = "HR11", Stanje = 100000, Blokirani = false });
Racuni.Add(new Racun { IBAN = "HR22", Stanje = 50000, Blokirani = false });
Racuni.Add(new Racun { IBAN = "HR33", Stanje = 12000, Blokirani = false });
Racuni.Add(new Racun { IBAN = "HR44", Stanje = 36000, Blokirani = true });
Racuni.Add(new Racun { IBAN = "HR55", Stanje = 8000, Blokirani = false });
Racuni.Add(new Racun { IBAN = "HR66", Stanje = 2000, Blokirani = false, OdobreniMinus = 3000 });
  
```

#### Scenariji koje treba pokriti testovima:

1. Ukoliko pozovemo metodu **DohvatiRacun** klase **Banka** te joj proslijedimo IBAN **HR12** (nepostojeći račun), metoda vraća **null**.
2. Ukoliko pozovemo metodu **DohvatiRacun** klase **Banka** te joj proslijedimo IBAN **HR11** (postojeći račun), metoda vraća objekt račun čije je svojstvo IBAN jednako **HR11**.
3. Ukoliko pozovemo metodu **DohvatiRacun** klase **Banka** te joj proslijedimo IBAN **HR44** (blokirani račun), metoda baca iznimku **BankAccountBlockedException**.
4. Ukoliko pozivom metode **Isplati** klase **Racun** pokušamo prebaciti iznos od **30 000 kn** sa računa **HR11** na račun **HR21** (nepostojeći račun), metoda baca iznimku **TransactionFailedException**.
5. Ukoliko pozivom metode **Isplati** klase **Racun** pokušamo prebaciti iznos od **30 000 kn** sa računa **HR11** na račun **HR22**, metoda uspijeva izvršiti transakciju, te je stanje računa **HR11 = 70 000 kn**, a stanje računa **HR22 = 80 000**.
6. Ukoliko pozivom metode **Isplati** klase **Racun** pokušamo prebaciti iznos od **30 000 kn** sa računa **HR11** na račun **HR22**, metoda uspijeva izvršiti transakciju, te je u dobivenom objektu **Isplata** platitelj jednak „**HR11**“, primatelj jednak „**HR22**“, a iznos jednak **30 000 kn**.
7. Ukoliko pozivom metode **Isplati** klase **Racun** pokušamo prebaciti iznos od **130 000 kn** sa računa **HR11** na račun **HR22**, metoda ne uspijeva izvršiti transakciju (jer je iznos prevelik), te baca iznimku **TransactionFailedException**.
8. Ukoliko pozivom metode **Isplati** klase **Racun** pokušamo prebaciti iznos od **3 000 kn** sa računa **HR66** (više nego ima na računu) na račun **HR55**, metoda uspijeva izvršiti transakciju jer račun ima dozvoljeni minus, i zbog toga je stanje računa **HR66 = -1 000 kn**, računa **HR55 = 11 000 kn**, a iznos u objektu **Isplata** je **3 000 kn**.
9. Ukoliko pozivom metode **Isplati** klase **Racun** pokušamo prebaciti iznos od **5 500 kn** sa računa **HR66** (više nego ima na računu i dozvoljenog minusa) na račun **HR55**, metoda ne uspijeva izvršiti transakciju, te baca iznimku **TransactionFailedException**.
10. Ukoliko pozivom metode **Isplati** klase **Racun** pokušamo prebaciti iznos od **-5 00 kn** sa računa **HR11** na račun **HR22**, metoda ne uspijeva izvršiti transakciju (jer je iznos negativan), te baca iznimku **TransactionFailedException**.

Potrebno je u dobivenom Solution-u kreirati novi **Unit Test projekt**, dodati **referencu** na projekt za autentifikaciju, te napisati jedinične (Unit) testove za klasu **Autentifikator**. Svaki od navedenih scenarija treba pokriti s posebnom testnom metodom. Testne metode nazvati po redu kako su navedene: Test1, Test2, Test3... **Testovi koji su obojeni crvenom bojom će inicijalno pasti, pa treba napraviti ispravke/dorade u implementaciji kako bi testovi prošli.**



```

Korisnici.Add(new Korisnik { Korisnickolme = "mmijac", Lozinka="abcde", Tip = TipKorisnika.Administrator });
Korisnici.Add(new Korisnik { Korisnickolme = "pivic", Lozinka = "12345", Tip = TipKorisnika.Obicni });
Korisnici.Add(new Korisnik { Korisnickolme = "valic", Lozinka = "ab234", Tip = TipKorisnika.Obicni });
Korisnici.Add(new Korisnik { Korisnickolme = "anovak", Lozinka = "qetzs", Tip = TipKorisnika.Obicni });
Korisnici.Add(new Korisnik { Korisnickolme = "thorvat", Lozinka = "hfdz", Tip = TipKorisnika.Obicni });
Korisnici.Add(new Korisnik { Korisnickolme = "gtudor", Lozinka = "nrtgs", Tip = TipKorisnika.Obicni, Aktivan = false });
Korisnici.Add(new Korisnik { Korisnickolme = "btomas", Lozinka = "kgdrt", Tip = TipKorisnika.Administrator });
Korisnici.Add(new Korisnik { Korisnickolme = "gost", Lozinka = "gost", Tip = TipKorisnika.Gost });

```

#### Scenariji koje treba pokriti testovima:

**Scenarij 1.** Ako pozivom metode **DohvatiKorisnika** pokušamo dohvatiti aktivnog korisnika (npr. „**valic**“), metoda vraća vrijednost koja nije null i vraćeni korisnik ima **Korisnickolme** jednako „**valic**“.

**Scenarij 2.** Ako pozivom metode **DohvatiKorisnika** pokušamo dohvatiti nepostojećeg korisnika („**mmarkic**“), metoda vraća null.

**Scenarij 3.** Ako pozivom metode **DohvatiKorisnika** pokušamo dohvatiti **neaktivnog** korisnika („**gtudor**“), metoda baca **InactiveUserException**.

**Scenarij 4.** Pozivom metode **RegistrirajKorisnika** i proslijeđivanjem korisničkih podataka za neaktivnog korisnika („**gtudor**“, „**nrtgs**“), ne registrira se novi korisnik, nego se postojeći korisnik „**gtudor**“ aktivira (**Aktivan = true**). To znači da dohvaćanjem korisnika „**gtudor**“ sada dobijemo korisnika koja nije null, i koji je aktivan (**Aktivan == true**).

**Scenarij 5.** Ako pozovemo metodu **RegistrirajKorisnika** i proslijedimo joj samo ispravno **korisničko ime** i **lozinku** (npr. „**mmarkic**“, „**101010**“), tada autentifikator registrira novog korisnika, i mora postojati korisnik sa korisničkim imenom „**mmarkic**“, lozinkom „**101010**“, te tipom **obični korisnik**.

**Scenarij 6.** Ako pozovemo metodu **RegistrirajKorisnika** i proslijedimo joj **korisničko ime** = „**thorvat**“ (već postojeće korisničko ime) i **lozinku** = „**121212**“, tada metoda baca iznimku **IncorrectAuthenticationData**.

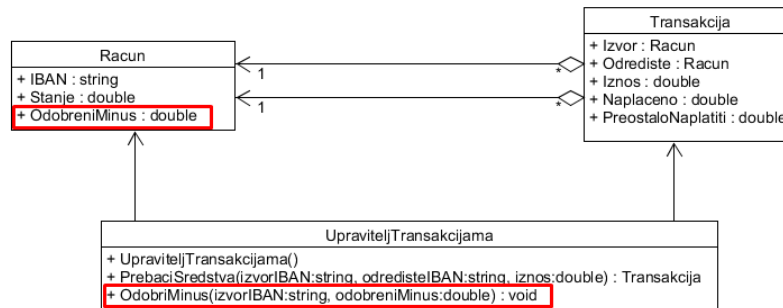
**Scenarij 7.** Ako pozivom metode **RegistrirajKorisnika** pokušam registrirati korisnika koji je administrator (npr. „**mmarkic**“, „**101010**“, **TipKorisnika.Administrator**), a prije toga je prijavljen obični korisnik (npr. *pivic*), metoda baca iznimku **UnauthorizedRegistrationException**. (Registriranje novog administratorskog korisnika može obaviti samo administrator).

**Scenarij 8.** Ako pozivom metode **RegistrirajKorisnika** pokušam registrirati korisnika koji je administrator (npr. „**mmarkic**“, „**101010**“, **TipKorisnika.Administrator**), a prije toga jeste već prijavljen neki drugi administrator (npr. *btomas*), metoda uspijeva registrirati korisnika, pa je moguće dohvatiti korisnika „**mmarkic**“ koji nije **null** i koji je **administrator**.

**Scenarij 9.** Pozovimo metodu **RegistrirajKorisnika** i proslijedimo joj ispravno **korisničko ime** i **lozinku**, te tip **Gost** (npr. „**mmarkic**“, „**101010**“, **TipKorisnika.Gost**). S obzirom da ne želimo da je moguće dodavati korisnike koji su tipa **gost**, autentifikator neće registrirati novog korisnika, tj. dohvaćanje korisnika „**mmarkic**“ će vratiti **null**.

**Scenarij 10.** Ako pozovemo metodu **RegistrirajKorisnika** i proslijedimo joj vrijednost „**test**“ kao **korisničko ime** (kraće od 5 znakova), metoda baca **IncorrectAuthenticationData** iznimku.

U dobivenom **Solution-u** kreirati novi **Unit Test projekt**, dodati **referencu** na projekt za bankovne transakcije, te napisati jedinične (Unit) testove za klasu **UpraviteljTransakcijama**. Svaki od navedenih scenarija treba pokriti s posebnom testnom metodom (dakle, ne sve scenarije u jednu metodu). **Testovi koji su obojeni crvenom bojom će inicijalno pasti, ili ih uopće neće biti moguće kreirati, pa treba napraviti ispravke/dorade u implementaciji kako bi testovi prošli.**



```

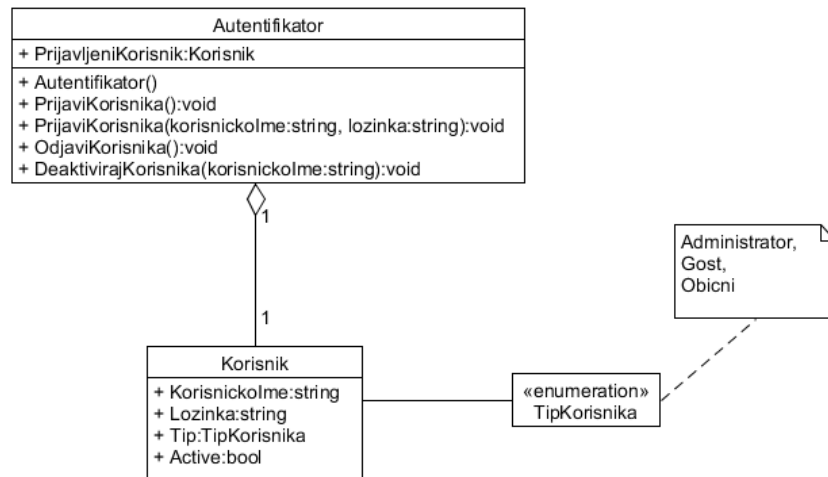
IBAN = "HR11", Stanje = 100000
IBAN = "HR22", Stanje = 50000
IBAN = "HR33", Stanje = 12000
IBAN = "HR44", Stanje = 36000
IBAN = "HR55", Stanje = 8000
IBAN = "HR66", Stanje = 2000
  
```

#### Scenariji koje treba pokriti testovima:

1. Ukoliko pozovemo metodu **PrebaciSredstva** te joj proslijedimo IBAN nepostojećeg računa izvorišnog računa, metoda baca iznimku **InvalidBankAccountException**.
2. Ukoliko pozovemo metodu **PrebaciSredstva** sa ispravnim IBAN-ovima (npr. „HR11“ i „HR22“), ali proslijedimo nepravilan iznos (negativan ili 0), metoda baca iznimku **InvalidAmountException**.
3. Ukoliko uz pomoć metode **PrebaciSredstva** prebacimo sa bankovnog računa „HR11“ na bankovni račun „HR22“ iznos od **30 000 kn**, nastala transakcija bi trebala pokazati da je novo stanje računa izvorišnog računa **70 000 kn**, a stanje odredišnog računa **80 000 kn**.
4. Ukoliko uz pomoć metode **PrebaciSredstva** prebacimo sa bankovnog računa „HR11“ na bankovni račun „HR22“ iznos od **30 000 kn**, nastala transakcija bi trebala pokazati da je naplaćeno **30 000 kn**, a preostalo za naplatiti **0 kn**.
5. Ukoliko uz pomoć metode **PrebaciSredstva** pokušamo prebaciti sa bankovnog računa „HR66“ na bankovni račun „HR55“ iznos od **3 500 kn**, nastala transakcija bi trebala pokazati da je novo stanje računa izvorišnog računa **0 kn**, a stanje odredišnog računa **10 000 kn**.
6. Napišite test koji bi utvrdio ispravnost transakcije, odnosno njenih vrijednosti **Naplaceno** i **PreostaloNaplatiti** u scenariju kada uz pomoć metode **PrebaciSredstva** pokušamo prebaciti sa bankovnog računa „HR66“ na bankovni račun „HR55“ iznos od **3 500 kn**. Imajte na umu nedovoljno stanje na računu „HR66“, te da će metoda **PrebaciSredstva** naplatiti onoliko koliko može a da bankovni račun ne ode u minus.
7. Ukoliko pozivom metodom **OdobriMinus** pokušamo odobriti minus nepostojećem računu, tj. proslijedimo joj nepostojeći **IBAN**, metoda baca iznimku **InvalidBankAccountException**.
8. Ukoliko uz pomoć metode **PrebaciSredstva** pokušamo prebaciti sa bankovnog računa „HR66“ na bankovni račun „HR55“ iznos od **3 500 kn**, a bankovni račun ima odobreni minus = **2000 kn**, nastala transakcija bi trebala pokazati da je novo stanje računa izvorišnog računa **-1 500 kn**, a stanje odredišnog računa **11 500 kn**.
9. Ukoliko uz pomoć metode **PrebaciSredstva** pokušamo prebaciti sa bankovnog računa „HR66“ na bankovni račun „HR55“ iznos od **4 500 kn**, a bankovni račun ima odobreni minus = **2 000 kn**, nastala transakcija bi trebala pokazati da je novo stanje računa izvorišnog računa **-2000 kn**, stanje odredišnog računa **12 000 kn**, da je naplaćeno **4 000 kn**, a preostalo za naplatiti **500 kn**.

## ZADATAK

Potrebno je u dobivenom Solution-u kreirati novi **Unit Test projekt**, dodati **referencu** na projekt za autentifikaciju, te napisati jedinične (Unit) testove za klasu **Autentifikator**. Svaki od navedenih scenarija treba pokriti s posebnom testnom metodom (dakle, ne sve scenarije u jednu metodu). Testne metode nazvati po redu kako su navedene: Test1, Test2, Test3... **Testovi koji su obojeni crvenom bojom će inicijalno pasti, ili ih uopće neće biti moguće kreirati, pa treba napraviti ispravke/dorade u implementaciji kako bi testovi prošli.**



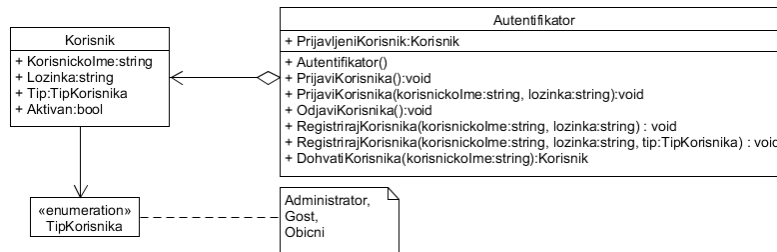
```
KorisnickoIme = "mmijac", Lozinka="abcde", Tip = TipKorisnika.Administrator
KorisnickoIme = "pivic", Lozinka = "12345", Tip = TipKorisnika.Obicni
KorisnickoIme = "valic", Lozinka = "ab234", Tip = TipKorisnika.Obicni
KorisnickoIme = "anovak", Lozinka = "qetzs", Tip = TipKorisnika.Obicni
KorisnickoIme = "thorvat", Lozinka = "hfdz", Tip = TipKorisnika.Obicni
KorisnickoIme = "gtudor", Lozinka = "nrtgs", Tip = TipKorisnika.Obicni
KorisnickoIme = "btomas", Lozinka = "kgdrt", Tip = TipKorisnika.Administrator
KorisnickoIme = "gost", Lozinka = "gost", Tip = TipKorisnika.Gost
```

### Scenariji koje treba pokriti testovima:

1. Ukoliko pozovemo metodu **PrijaviKorisnika** i proslijedimo joj ispravnu kombinaciju **korisničkog imena** i **lozinke**, tada mora vrijediti da svojstvo **PrijavljeniKorisnik** nije **null**, te da je u svojstvu **PrijavljeniKorisnik** zaista onaj korisnik koji se prijavio.
2. Ukoliko pozovemo metodu **PrijaviKorisnika** i proslijedimo joj prazno („“) **korisničko ime** i **lozinku**, tada bi se automatski trebao prijaviti gost, pa svojstvo **PrijavljeniKorisnik** ne smije biti **null**, i u njemu mora biti korisnik koji ima svojstvo **Tip** sa vrijednošću **Gost**.
3. Ukoliko pozovemo metodu **PrijaviKorisnika** i proslijedimo joj neispravnu kombinaciju **korisničkog imena** i **lozinke**, tada metoda baca iznimku **FailedAuthenticationException**.
4. U slučaju je da neki korisnik prijavljen, poziv metode **OdjaviKorisnika** rezultira time da svojstvo **PrijavljeniKorisnik** postaje **null**.
5. Prijava korisnika (npr. *pivic*) nakon što je još uvijek prijavljen drugi korisnik (npr. *anovak*) rezultira iznimkom **InvalidOperationException**.
6. Poziv metode **PrijaviKorisnika** bez parametara prijavljuje gosta, pa svojstvo **PrijavljeniKorisnik** ne smije biti **null**, i u njemu mora biti korisnik koji ima svojstvo **Tip** sa vrijednošću **Gost**.
7. Poziv metode **OdjaviKorisnika** kada korisnik nije prijavljen rezultira iznimkom **InvalidOperationException**.
8. Pokušaj deaktiviranja korisnika (npr. *pivic*) uz pomoć metode **DeaktivirajKorisnika** kada je prijavljen korisnik koji nije administrator (npr. *anovak*) rezultira iznimkom **InvalidOperationException**.
9. Kada se administrator prijavi (npr. *mmijac*), deaktivira korisnika (npr. *pivic*) uz pomoć metode **DeaktivirajKorisnika**, te zatim odjavi, pokušaj prijave deaktiviranog korisnika *pivic* rezultira iznimkom **UserDeactivatedException**.



Potrebno je u dobivenom Solution-u kreirati novi **Unit Test projekt**, dodati **referencu** na projekt za autentifikaciju, te napisati jedinične (Unit) testove za klasu **Autentifikator**. Svaki od navedenih scenarija treba pokriti s posebnom testnom metodom (dakle, ne sve scenarije u jednu metodu). Testne metode nazvati po redu kako su navedene: Scenarij 1, Scenarij2, Scenarij3, i sl.... **Testovi koji su obojeni crvenom bojom će inicijalno pasti, ili ih uopće neće biti moguće kreirati, pa treba napraviti ispravke/dorade u implementaciji kako bi testovi prošli.**



```

{ KorisnickoIme = "mmijac", Lozinka="abcde", Tip = TipKorisnika.Administrator });
{ KorisnickoIme = "pivic", Lozinka = "12345", Tip = TipKorisnika.Obicni });
{ KorisnickoIme = "valic", Lozinka = "ab234", Tip = TipKorisnika.Obicni });
{ KorisnickoIme = "anovak", Lozinka = "qetzs", Tip = TipKorisnika.Obicni });
{ KorisnickoIme = "thorvat", Lozinka = "hfdrr", Tip = TipKorisnika.Obicni, Aktivan = false });
{ KorisnickoIme = "gtudor", Lozinka = "nrtgs", Tip = TipKorisnika.Obicni });
{ KorisnickoIme = "btomas", Lozinka = "kgdrt", Tip = TipKorisnika.Administrator });
{ KorisnickoIme = "gost", Lozinka = "gost", Tip = TipKorisnika.Gost });
  
```

#### Scenariji koje treba pokriti testovima:

**Scenarij 1.** Ukoliko pozivom metode **DohvatiKorisnika** pokušamo dohvatiti aktivnog korisnika (npr. „pivic“), metoda vraća vrijednost koja nije null.

**Scenarij 2.** Ukoliko pozivom metode **DohvatiKorisnika** pokušamo dohvatiti neaktivnog korisnika („thorvat“), metoda vraća null vrijednost.

**Scenarij 3.** Ukoliko pozivom metode **DohvatiKorisnika** pokušamo dohvatiti nepostojećeg korisnika (npr. „pperic“), metoda vraća null vrijednost.

**Scenarij 4.** Ukoliko pozovemo metodu **RegistrirajKorisnika** i proslijedimo joj **korisničko ime** = „mmijac“ (već postojeće korisničko ime) i **lozinku** = „abc123“, tada metoda baca iznimku **IncorrectAuthenticationData**.

**Scenarij 5.** Ukoliko pozovemo metodu **RegistrirajKorisnika** i proslijedimo joj **korisničko ime** kraće od 5 znakova, tada metoda baca iznimku **IncorrectAuthenticationData**.

**Scenarij 6.** Ukoliko pozovemo metodu **RegistrirajKorisnika** i proslijedimo joj samo ispravno **korisničkog ime** i **lozinku** (npr. „pperic“, „123456“), tada autentifikator registrira novog korisnika, i mora postojati korisnik sa korisničkim imenom „pperic“, lozinkom „123456“, te tipom **obični korisnik**. **HINT: Možete koristiti metodu DohvatiKorisnika za dohvaćanje korisnika.**

**Scenarij 7.** Ukoliko pozovemo metodu **RegistrirajKorisnika** i proslijedimo joj ispravno **korisničkog ime** i **lozinku**, te **tip Gost** (npr. „pperic“, „123456“, **TipKorisnika.Gost**). S obzirom da ne želimo da je moguće dodavati korisnike koji su tipa gost, autentifikator neće registrirati novog korisnika, tj. dohvaćanje korisnika „pperic“ će vratiti **null**. **HINT: Možete koristiti metodu DohvatiKorisnika za dohvaćanje korisnika.**

**Scenarij 8.** U slučaju da pozivom metode **RegistrirajKorisnika** pokušam registrirati korisnika koji je administrator (npr. „pperic“, „123456“, **TipKorisnika.Administrator**), a prije toga nije već prijavljen neki drugi administrator, metoda baca iznimku **UnauthorizedRegistrationException**. (Ideja je da registriranje administratorskog korisnika može obaviti samo već postojeći administrator).

**Scenarij 9.** U slučaju da pozivom metode **RegistrirajKorisnika** pokušam registrirati korisnika koji je administrator (npr. „pperic“, „123456“, **TipKorisnika.Administrator**), a prije toga jeste već prijavljen neki drugi administrator, metoda uspijeva registrirati korisnika, pa je moguće dohvatiti korisnika „pperic“ koji nije **null** i koji je **administrator**.

**Scenarij 10.** U slučaju poziva metode **RegistrirajKorisnika** i proslijeđivanjem korisničkih podataka za neaktivnog korisnika („thorvat“, „hfdrr“), ne registrira se novi korisnik, nego se postojeći korisnik „thorvat“ aktivira (**Aktivan = true**). To znači da dohvaćanjem korisnika „thorvat“ dobijemo korisnika koja nije null, i koji je aktivan (**Aktivan == true**).