

ZADATAK za 23 poena

Paket `net.simulation`:

Apstraktan tip `RacunarNaMrezi` koji sadrži:

- privatnu promenljivu `adresa` tipa `String`;
- javni konstruktor koji prihvata adresu računara i setuje je;
- javni getter za atribut `adresa`;
- prepisan metod `toString` koji vraća vrednost atributa `adresa`;
- javni metod `equals` koji prihvata objekat tipa `String` i poredi ga sa vrednošću atributa `adresa`;
- javni metod `equals` koji prihvata objekat tipa `RacunarNaMrezi` i poredi njegovu adresu sa atributom `adresa`;

Interfejs `Klijent` sadrži sledeće funkcionalnosti:

- metod `prijaviSe` koji prihvata objekat tipa `String` (predstavlja adresu servera) i nema povratnu vrednost;
- metod `odjaviSe` koji prihvata objekat tipa `String` (predstavlja adresu servera) i nema povratnu vrednost;
- metod `posaljiZahtev` koji prihvata objekat tipa `String` (predstavlja adresu servera) i nema povratnu vrednost;
- metod `prihvatiOdgovor` koji prihvata objekat tipa `String` (predstavlja adresu servera) i nema povratnu vrednost;
- metod `getAdresa` koji ne prihvata argumente i vraća objekat tipa `String`;

Tip `VebKlijent` koji predstavlja računar na mreži koji je klijent i sadrži:

- atribut `imrBrauzera` tipa `String`, sa package vidljivošću;
- javni konstruktor koji prihvata adresu i imeBrauzera i setuje ih;
- javni metod `prijaviSe` koji prihvata adresu servera (tipa `String`) i mreži prosleđuje zahtev sa porukom `prijava`;
- javni metod `odjaviSe` koji prihvata adresu servera (tipa `String`) i mreži prosleđuje zahtev sa porukom `odjava`;
- javni metod `posaljiZahtev` koji prihvata adresu servera (tipa `String`) i mreži prosleđuje zahtev sa porukom `ping`;
- javni metod `prihvatiOdgovor` koji prihvata poruku tipa `String` i štampa je u formatu `stringReprezentacija:poruka`;
- prepisan metod `toString` koji vraća string reprezentaciju oblika `adresa [imeBrauzera]`;

Tip `Server` koji predstavlja računar na mreži i sadrži:

- protected promenljivu `prijavljeniKlijenti` tipa `Klijent[]`;
- protected promenljivu `brPrijavljenihKlijenata` tipa `int` koji predstavlja broj prijavljenih klijenata;
- javni konstruktor koji prihvata adresu i setuje je;
- javnu metodu `prihvatiZahtev` koja prihvata objekat tipa `Klijent` i objekat tipa `string` koji predstavlja poruku koju šalje klijent. U slučaju da je poruka:
 - `prijava` - klijent se dodaje u niz prijavljenih klijenata;
 - `odjava` - klijent se uklanja iz niza prijavljenih klijenata;
 - `ping` - ukoliko je klijent prijavljen, vrši obradu zahteva;
- protected metodu `obradi` koja prihvata objekat tipa `Klijent` koji predstavlja klijenta čiji se zahtev obrađuje i preko mreže šalje odgovor na odgovarajuću adresu oblika „Server `adresa` šalje poruku.”;

Tip `MulticastServer` koji je specijalna vrsta servera kojii sadrži:

- javni konstruktor koji prihvata adresu i setuje je;
- prepisanu protected metodu `obradi` koja prihvata objekat tipa `Klijent` koji predstavlja klijenta čiji se zahtev obrađuje i preko mreže na adrese svih prijavljenih klijenata šalje odgovor oblika „Server `adresa` svima šalje poruku, jer se javio `podaciKlijenta`.”;

Tip `Mreza` koji sadrži:

- privatni statički atribut `dostupniUredjaji` tipa `RacunarNaMrezi[]`;
- privatni statički atribut `brUredjaja` tipa `int` koji predstavlja broj prijavljenih uređaja;
- javnu statičku metodu `registrujUredjaj` koji prihvata objekat tipa `RacunarNaMrezi` i dodaje ga u niz dostupnih uređaja i nema povratnu vrednost
- privatnu statičku metodu `pronadji` koja prihvata argument tipa `String`, pronalazi `RacunarNaMrezi` u nizu dostupnih uređaja sa datom adresom i vraća ga kao povratnu vrednost;
- javnu statičku metodu `proslediZahtev` koja prihvata string adresu na koju se šalje zahtev, objekat tipa `Klijent` koji šalje zahtev i poruku tipa `String`, pronalazi server sa odgovarajućom adresom i ukoliko postoji prosleđuje zahtev tom serveru;
- javnu statičku metodu `proslediOdgovor` koji prihvata string adresu klijenta kojem se šalje odgovor i string poruku koja se šalje, pronalazi klijenta sa odgovarajućom adresom i ukoliko postoji prosleđuje mu odgovor;

Default paket:

Testna klasa treba da:

- kreira tri servera, dva obična (imi,pmf) i jedan multikast (facebook), dva veb klijenta (ivica i marica) i sve ih smešta u niz i registruje ih na mreži;
- ivicu prijavljuje na imi i facebook, a maricu na facebook;
- ivica salje zahteve sledećim redosledom:
 - imi
 - facebook
 - pmf
 - facebook
- ivica i marica se odjavljuju.

