

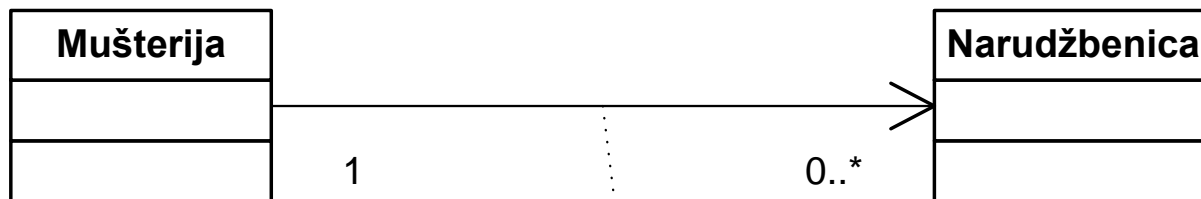


# OOAD-Priprema za ispit

---

# Zadatak 1

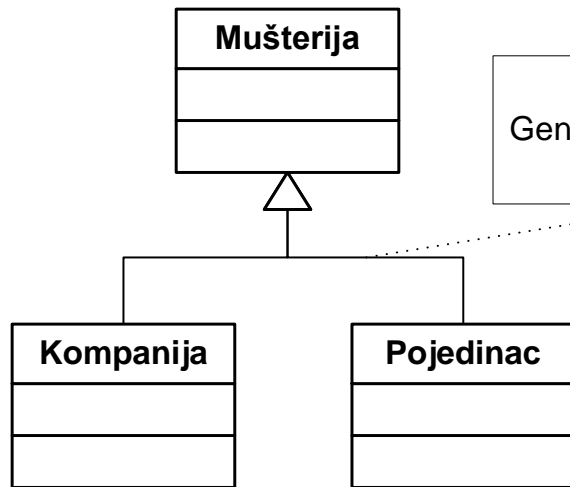
Za jednu mušteriju možemo da vežemo više narudžbenica.



Za JEDNU Mušteriju MOŽEMO (0 ILI) da vežemo VIŠE Narudžbenica.

# Zadatak 2

Mušterija može biti kompanija ili pojedinac.

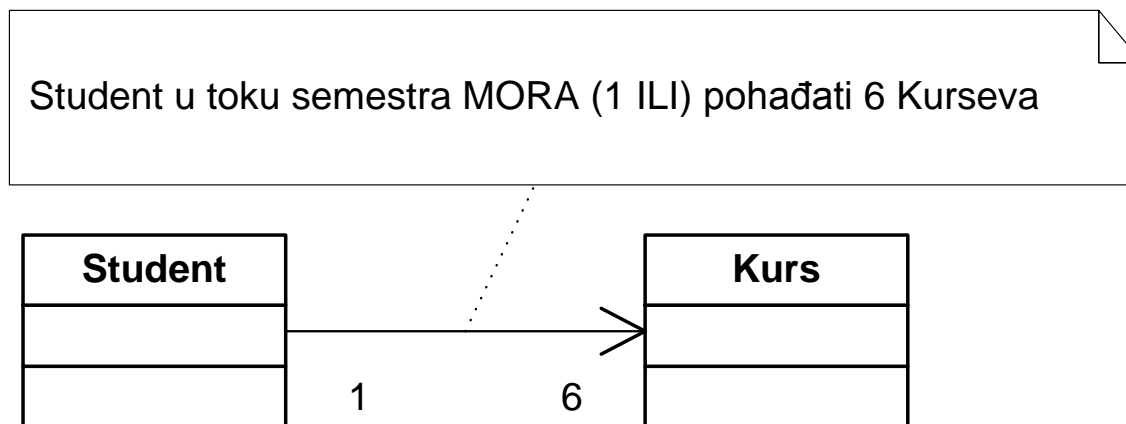


Generalizacija - Klase Kompanija i Pojedinac su izvedene klase od klase Mušterija.

# Zadatak 3

Student u toku semestra mora pohađati 6 kurseva, a pri tome je broj studenata po kursu proizvoljan.

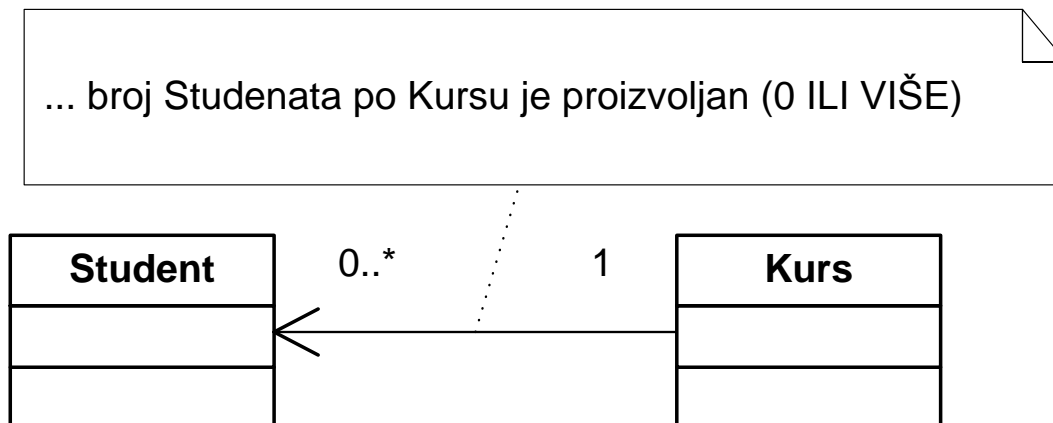
Razmišljanje:



# Zadatak 3

Student u toku semestra mora pohađati 6 kurseva, a pri tome je broj studenata po kursu proizvoljan.

Razmišljanje:

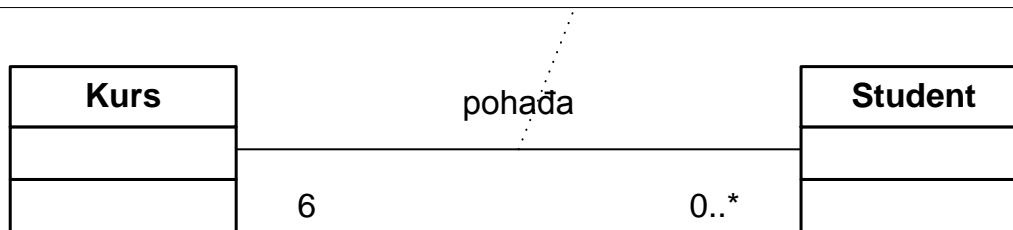


# Zadatak 3

Student u toku semestra mora pohađati 6 kurseva, a pri tome je broj studenata po kursu proizvoljan.

Spajanje:

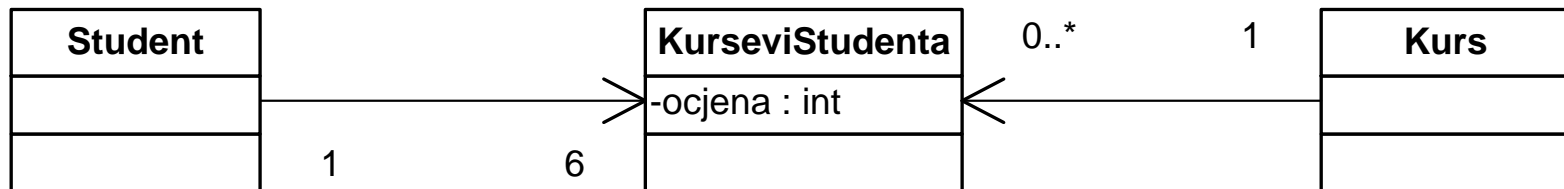
Klasa Student ima osobinu kursevi:Kursevi[6], a klasa Kurs ima osobinu studenti:Student[\*] - dvosmijerna asocijacija



# Zadatak 4

Student u toku semestra mora pohađati 6 kurseva, a pri tome je broj studenata po kursu proizvoljan. Za svakog studenta treba voditi podatke o ocjenama.

Klasa asocijacija - KurseviStudenta. Dodajemo atribut ocjena:int i razbijamo dvosmijernu asocijaciju.





# Zadatak 5

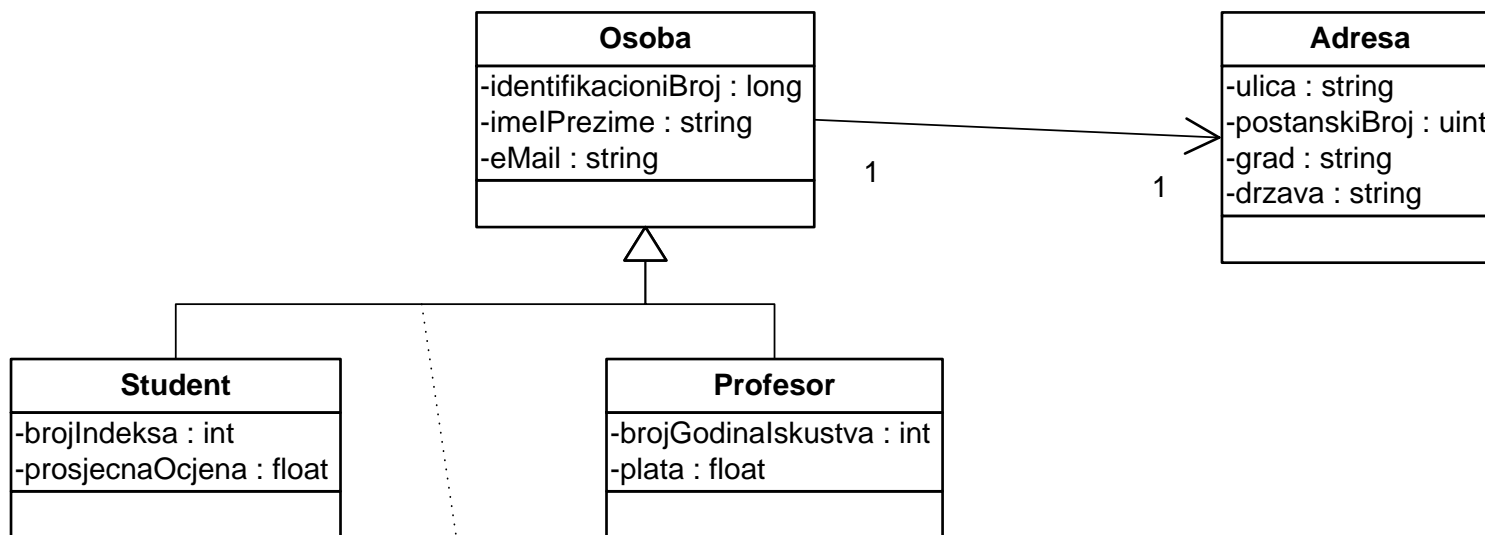
---

Prikazati dijagram klasa koji bi se odnosio na sljedeće:

Želimo da osobe opišemo atributima: identifikacioni broj, ime i prezime, e-mail a onda da vežemo podatke o adresama stanovanja: ulica, poštanski broj, grad, država. Osobe mogu biti studenti i profesori. Studenti pored osnovnih atributa vezanih za osobu imaju dodatne: broj indeksa i dosadašnja prosječna ocjena studija, a profesori pored osnovnih atributa vezanih za osobu imaju dodatne: godine iskustva i plata.



# Zadatak 5



Generalizacija - Klase Student i Profesor su izvedene klase od klase Osoba.  
Uočiti dodatne atribute.

# Zadatak 6

Prikazati dio dijagrama klasa koji bi se odnosio na sljedeće:  
Zgrada se sastoji od više stanova a stanovi se sastoje od 4 sobe.

Kompozicija - Cjelina u potpunosti posjeduje svoje dijelove.



# Zadatak 7

Koje relacije u dijagramu klasa se nalaze između Help sistema, GUI-a i podataka.

Pošto zadatak nije u potpunosti određen onda moramo prepostaviti neke stvari ili ih saznati.  
Pretpostavimo: Help sistem je u vidu aplikacije koja ima jedan GUI putem kojeg prezentira jedan ili više podataka.  
Pod ovom pretpostavkom je to kompozicija.



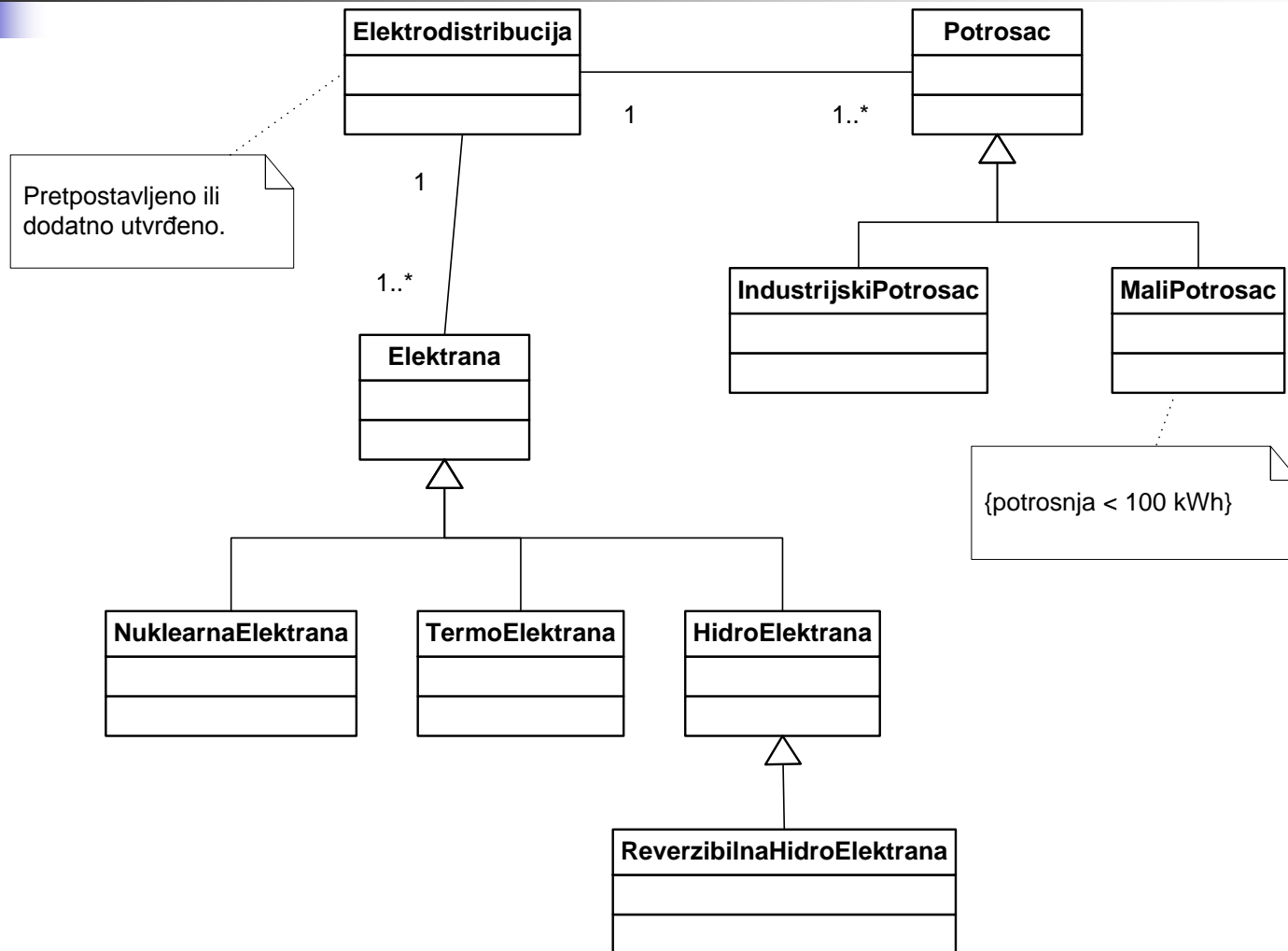


# Zadatak 8

---

Modelirajte (nacrtajte dijagram nasljeđivanja) jednostavan elektroenergetski sistem u kojem postoje 4 tipa elektrana: nuklearna, termoelektrana, hidroelektrana i reverzibilna hidroelektrana. Također ubacite i hijerarhiju klasa za potrošače, gdje ćete napraviti razliku između industrijskih potrošača (koji troše veće količine struje) i malih potrošača (domaćinstava).

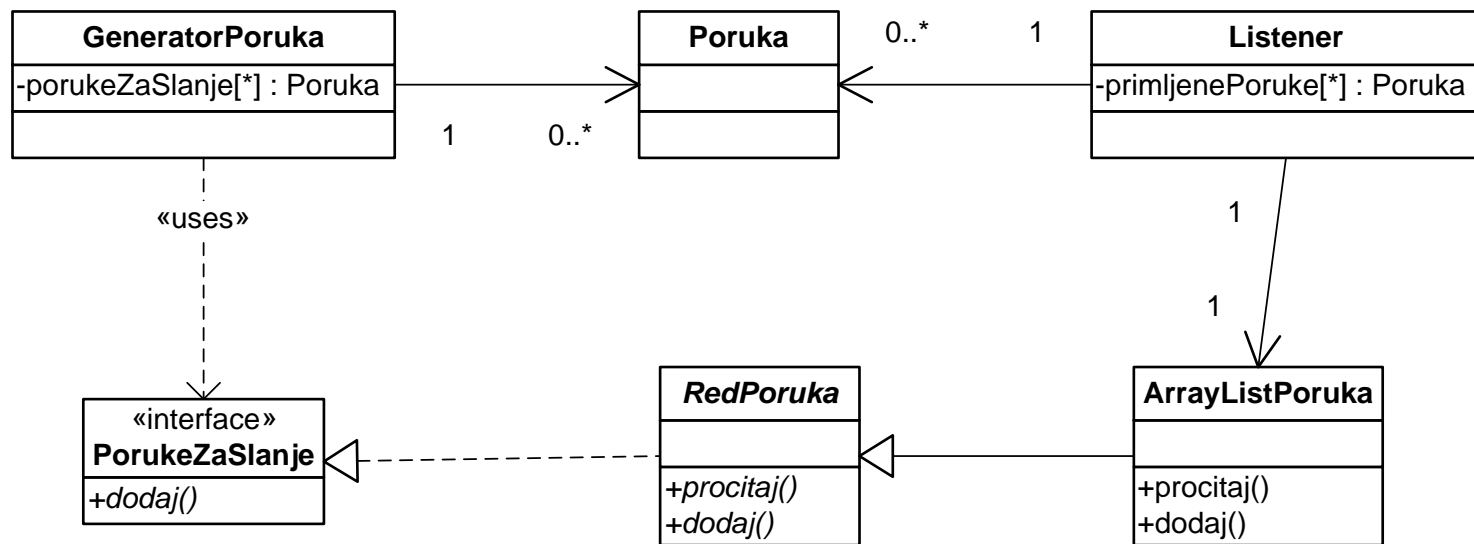
# Zadatak 8



# Zadatak 9

Modelirati *event-based* komunikaciju između generatora poruka i *listenera*, pod pretpostavkom da se komunikacija odvija putem interfejsa.

Moguće rješenje:





# Zadatak 9

---

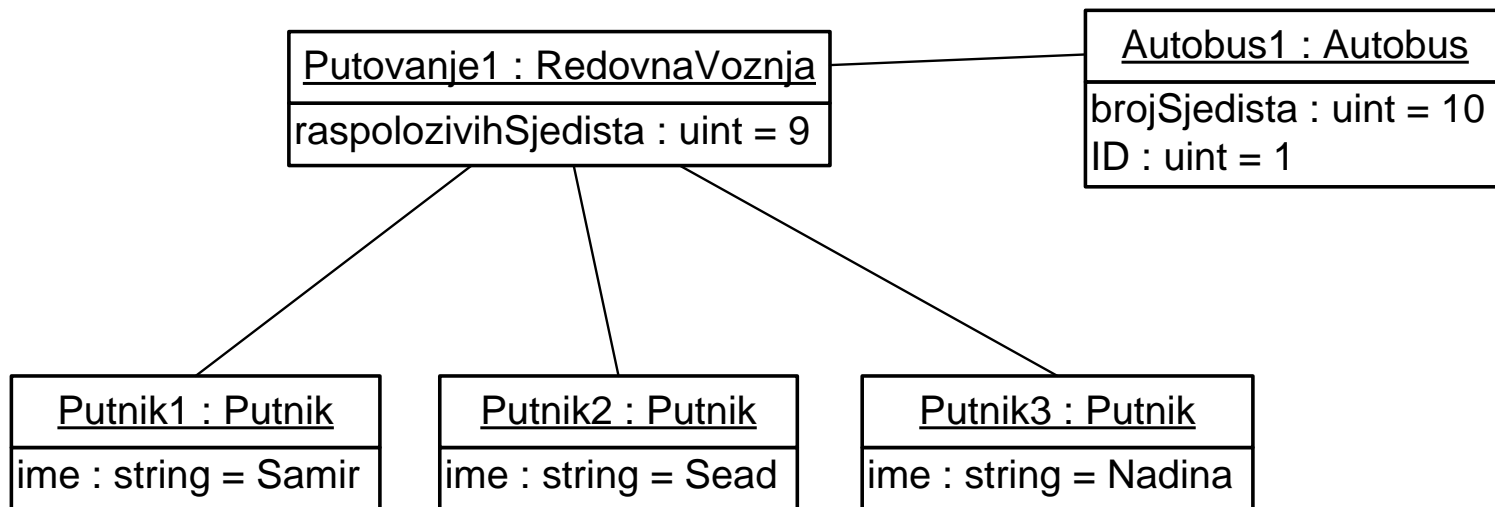
Objašnjenje:

**GeneratorPoruka** sadrži listu objekata klase **Poruka** i koristi (zahtijeva) interfejs **PorukeZaSlanje** da bi upotrebom metoda *dodaj()* poslao jednu od poruka **Listeneru**. Listener realizuje interfejs, tako što koristi primjerak klase **ArrayListPoruka** koja je naslijedila apstraktnu klasu **RedPoruka** i njene apstraktne metode *procitaj()* i *dodaj()* implemetirala.

Listener također radi sa objektima klase Poruka pa otuda asocijacija sa tom klasom.

# Zadatak 10

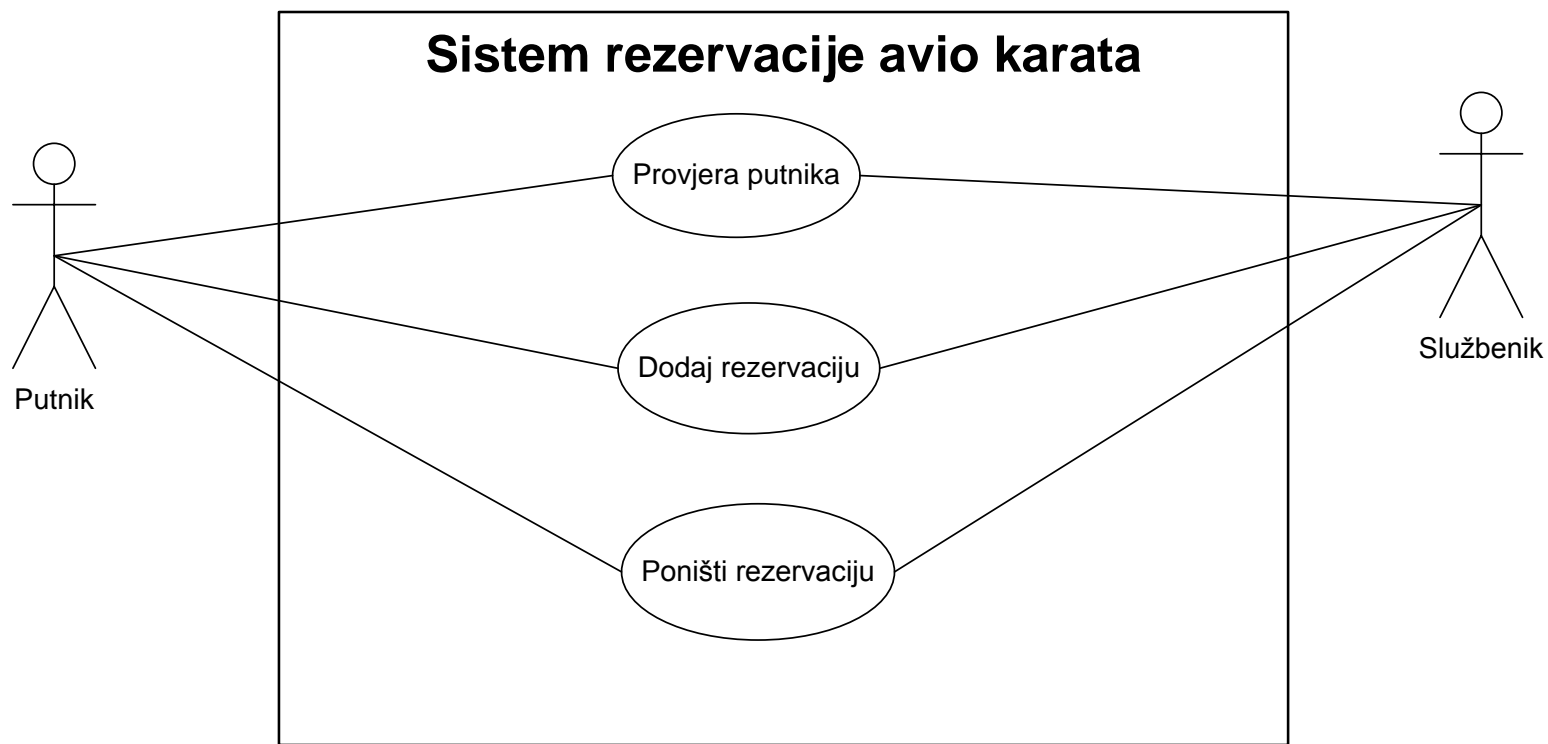
Dijagram objekata...





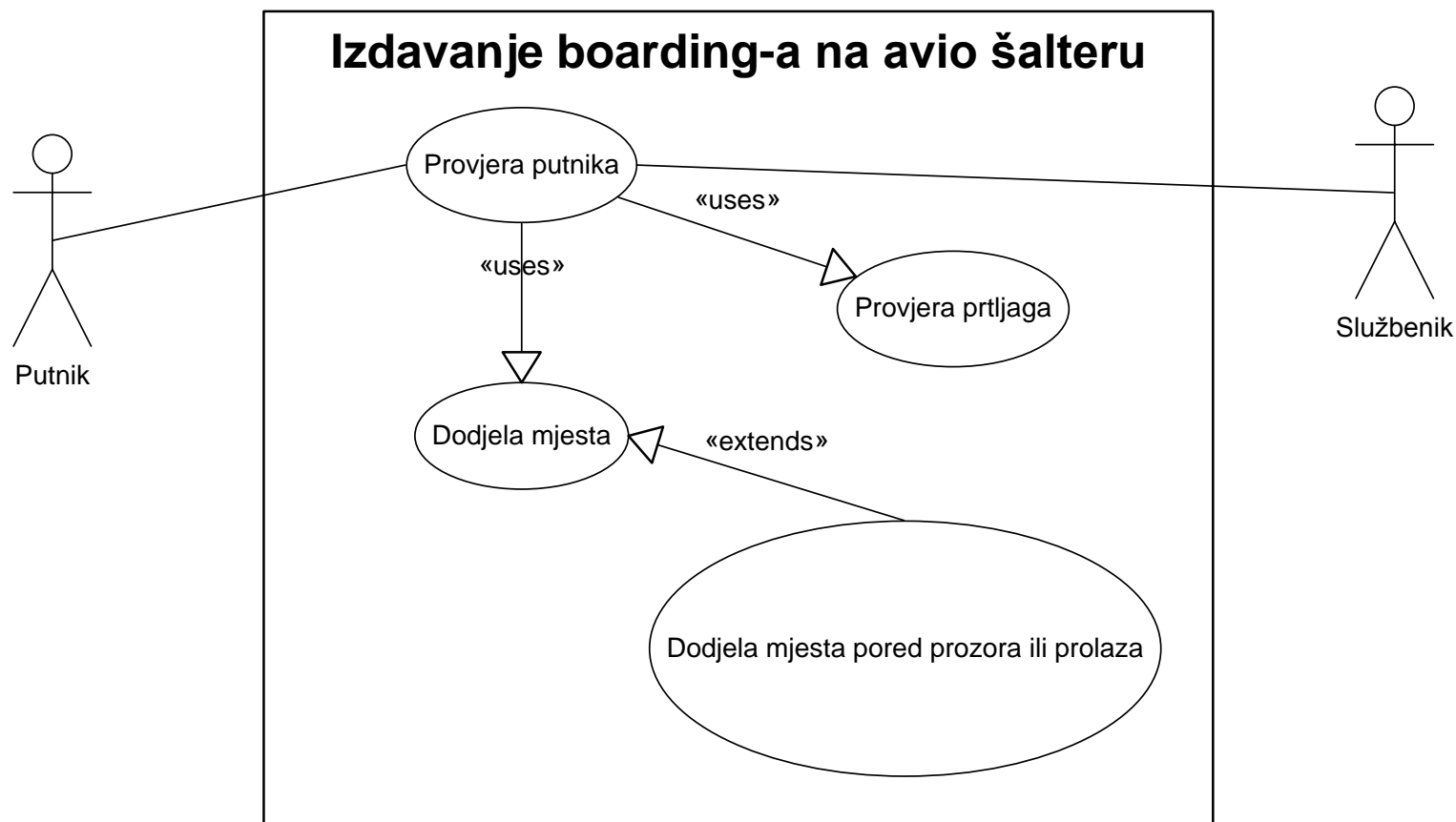
# Zadatak 11

Dijagrami slučajeva upotrebe ...



# Zadatak 12

Dijagrami slučajeva upotrebe ... <<uses>> <<include>>



# Zadatak 13

Nacrtati dijagram aktivnosti koji će realizovati narudžbu, čekati tri dana i poslati račun korištenjem *time event*-a

