

Klasni dijagrami

# Statički model sistema

- Prikazuje se korišćenjem
  - Klasnih dijagrama
  - Objekatskih dijagrama (statički dijagram objekata)
- Statičkim modelom sistema opisuju se entiteti sistema
- Dijagram klasa prikazuje jedan pogled na sistem
- Dijagram objekata mora biti saglasan sa odgovarajućim dijagramom klasa

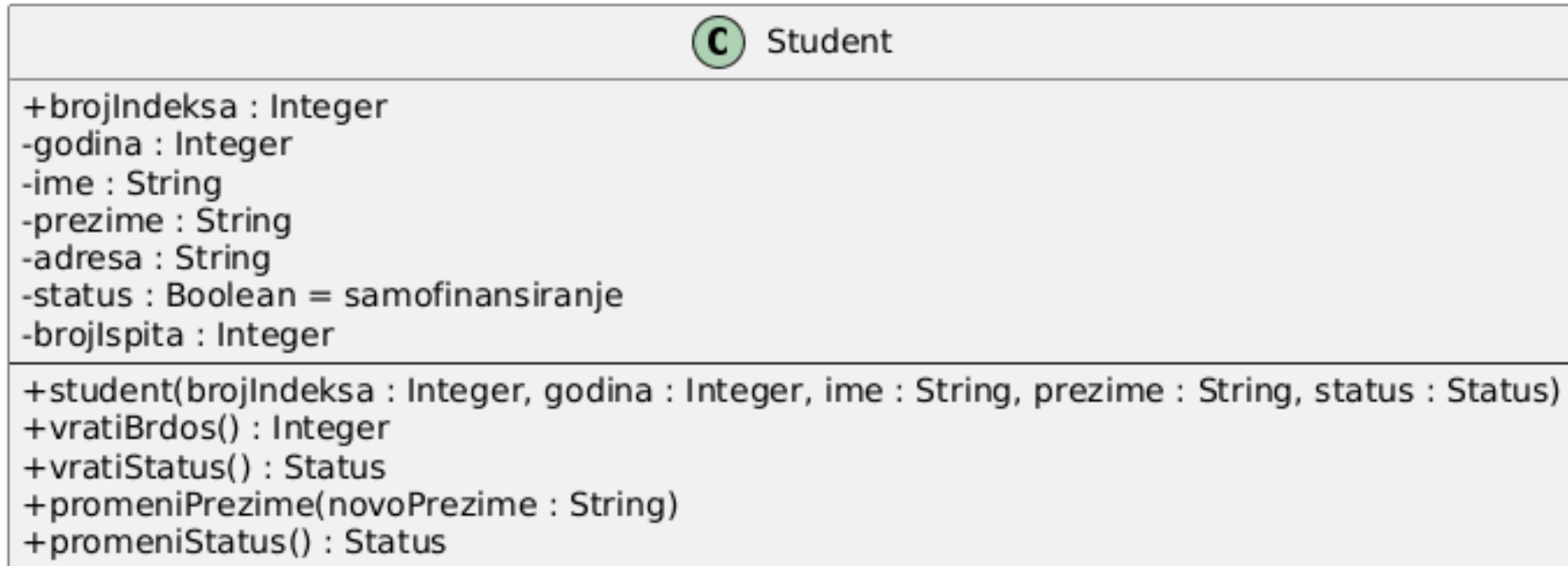
# Klasa

- Predstavlja skup atributa i operacija kojima se opisuje struktura (stanje) i ponašanje objekata posmatrane klase
- Klasa se predstavlja pravougaonikom podjeljenim na tri dela
- U gornjem pravougaoniku prikazuje se ime klase, stereotip i pripadnost paketu (biblioteci)
  - Stereotipom se označava namena klase, npr. entitetska klasa
- U srednjem delu se navode atributi klase
- U donjem delu se navode operacije klase

# Notacija klase u UML-u

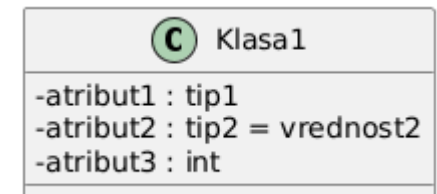


# Primer klase u UML-u



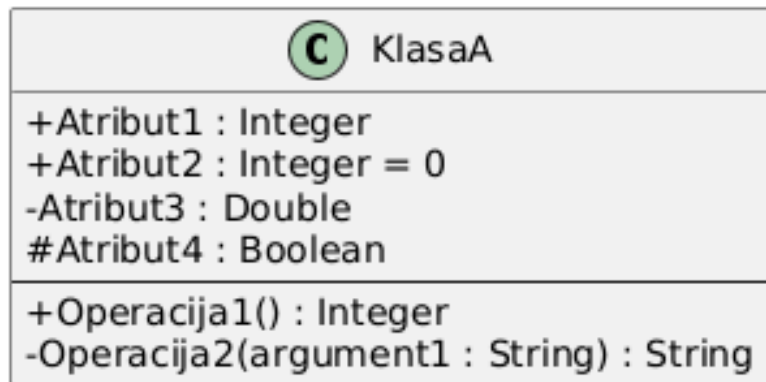
# Modeliranje vidljivosti

- Za svaki atribut je potrebno specifi vidljivost, tj. da li je ostalim objektima dozvoljeno da vide atribut
  - Public(+) atribut znači da tom atributu može da se pristupi spolja preko objekta te klase
  - Private(-) atributu se može pristupiti samo preko operacije te klase
  - Protected(#) atributu se pristupa preko objekata osnovne i izvedene klase
  - Package(~) atributu se pristupa unutar paketa kome ta klasa pripada
- Može se definisati početna vrednost atributa
- Podrazumevana vidljivost atributa je private
- Ako se drugačije ne naznači atributi su privatni



# Modeliranje operacija

- Operacijama se kao i atributima dodeljuje vidljivost
- Podrazumevana vidljivost operacija je **public**
- Definišu se ulazni argumenti operacije kao i povratni tip operacije



# Veze između klasa

- Asocijacija (association)
- Agregacija (aggregation)
- Nasleđivanje (inheritance)
- Zavisnost (dependency)

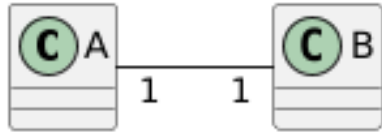
# Asocijacija

- Pretstavlja semantičku vezu između dve ili više klasa
- Asocijacije mogu biti bidirekzione ili unidirekzione
- Asocijacija dozvoljava objektu jedne klase da zna o javnim atributima i operacijama objekta druge klase
- Klasa može imati vezu asocijacije sama sa sobom tzv. refleksivna asocijacija

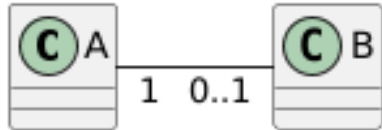
# Karakteristike asocijacije

- Multiplikativnost određuje koliko instanci jedne klase može biti povezano sa instancom druge klase
- Asocijacija dozvoljava jednoj klasi da zna o javnim atributima i operacijama druge klase
- Naziv asocijacije se definiše opciono (može ali ne mora) i ukazuje na značenje same asocijacije
- Krajevi asocijije predstavljaju uloge (role) koje pojedine klase imaju u asocijaciji

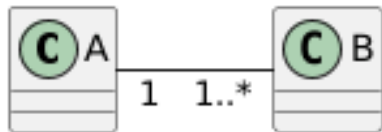
# Multiplikativnost asocijacije



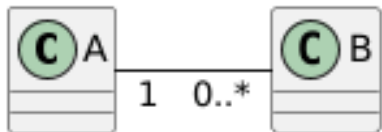
Svakoj instanci klase A odgovara tačno jedna instanca klase B



Svakoj instanci klase A može odgovarati nijedna ili jedna instanca klase B

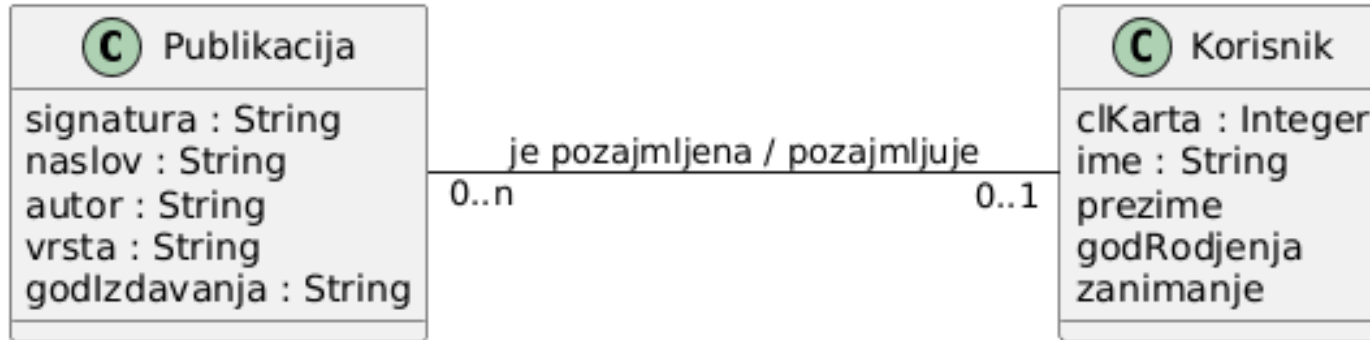


Svakoj instanci klase A odgovara jedna ili više instanca klase B



Svakoj instanci klase A odgovara nijedna ili više instanca klase B

# Primer: dijagram klasa

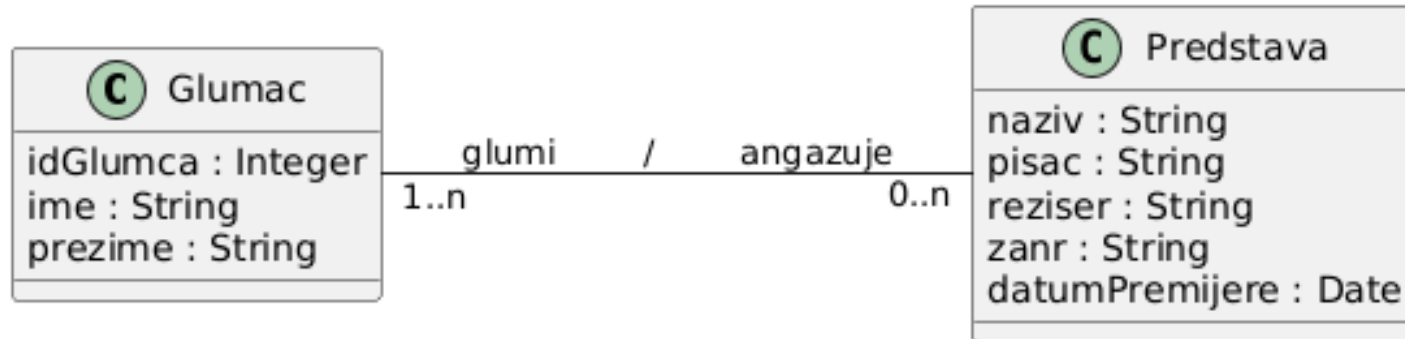


**Publikacija** (*signatura, naslov, vrsta, autor, godIzdavanja*) i **Korisnik** (*cKarta, ime, prezime, godRodjenja, zanimanje*).

- Publikacija ne mora biti pozajmljena od strane nijednog korisnika, a u jednom trenutku može biti pozajmljena od strane najviše jednog korisnika
- Korisnik istovremeno može pozajmiti više publikacija, a ne mora ni jednu

# Primer : asocijacije

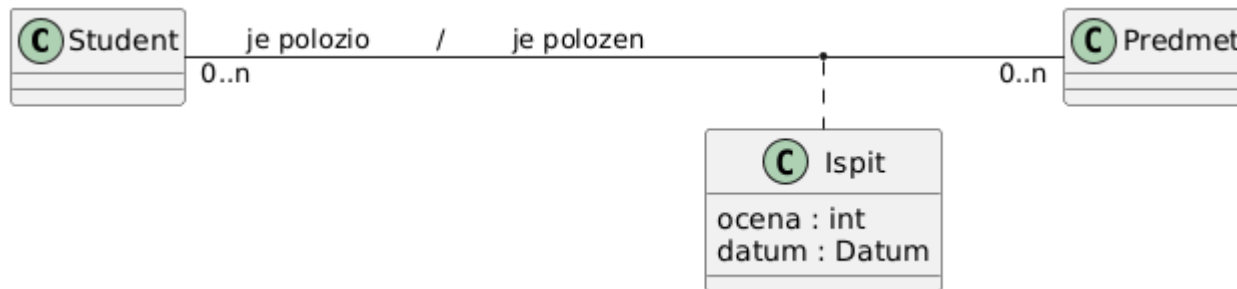
**Predstava** (*naziv, pisac, reziser, zanr, datumPremijere*)  
**Glumac** (*idGlumca, ime, prezime*).



- Predstava angažuje 1 ili više glumaca
- Glumac ne mora da glumi ni u jednoj, a može istovremeno i u više predstava.

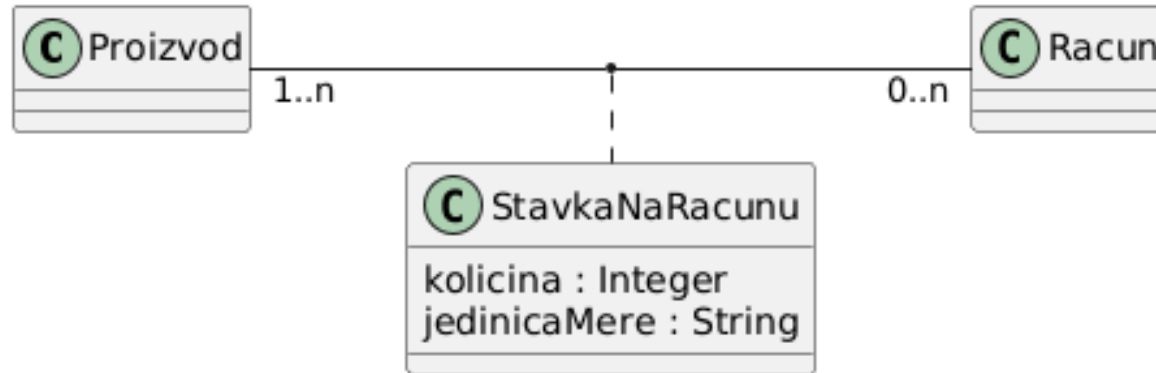
# Klasa asocijacije

- Predstavlja asocijaciju koja ima karakteristike klase, ima atribute i operacije a moguće su i veze sa drugim klasama u sistemu
- Klasa asocijacije je vezana za asocijaciju a ne za bilo koju od klasa koje su povezane asocijacijom



Za svaki položen ispit od strane svakog studenta zna se ocena koja je dobijena na ispitu kao i datum polaganja ispita

# Primer klase asocijacija

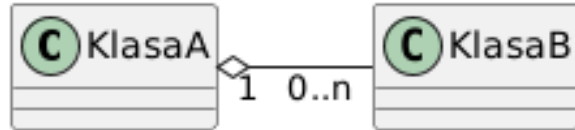


- Proizvod se ne mora nalaziti ni na jednom računu, ali se može nalaziti i na više njih
- Račun sadrži najmanje jedan, a može sadržati i više proizvoda
- Za svaki prodat proizvod se zna količina u kojoj je prodat i jedinica mere.

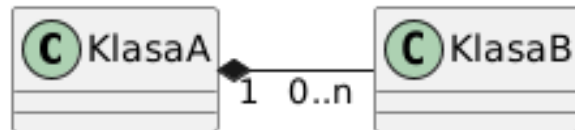
# Agregacija i kompozicija

- Tip veze asocijacije sa semantikom celina-deo
- Agregacija je **slabiji** oblik odnosa "deo-celina" (whole-part).
- Kod agregacije objekti mogu postojati nezavisno jedan od drugog
- Kompozicija je **jači** oblik odnosa "deo-celina".
- Kod kompozicije objekat ne može postojati bez objekta koji ga sadrži

# Oznake agregacije i kompozicije

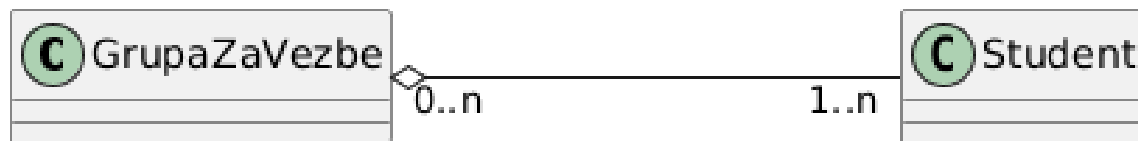


Objekat klase A predstavlja **agregaciju** objekata klase B (slab odnos “deo-celina”)



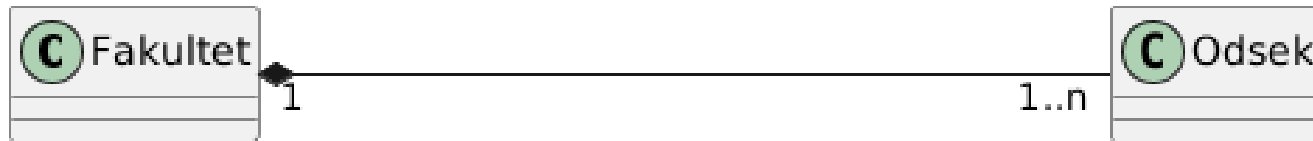
Objekat klase A predstavlja **kompoziciju** objekata klase B (jak odnos “deo-celina”)

# Primer agregacije



- Grupa za vežbe iz nekog predmeta sačinjena je od jednog ili više studenata
- Svaki student može da prisutvuje vežbama iz nijednog ili više predmeta
- Ako student prisustvuje vežbama onda je on član odgovarajuće grupe za vežbe
- Student postoji nezavisno od bilo koje grupe za vežbe

# Primer kompozicije

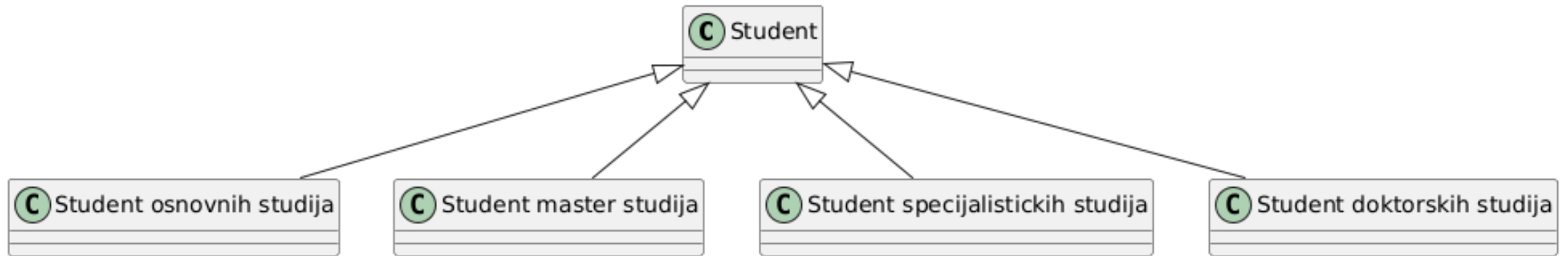


- Svaki fakultet sačinjen je od jednog ili više odseka
- Svaki odsek pripada jednom i samo jednom fakultetu
- Odsek ne može postojati bez svog fakulteta

# Generalizacija

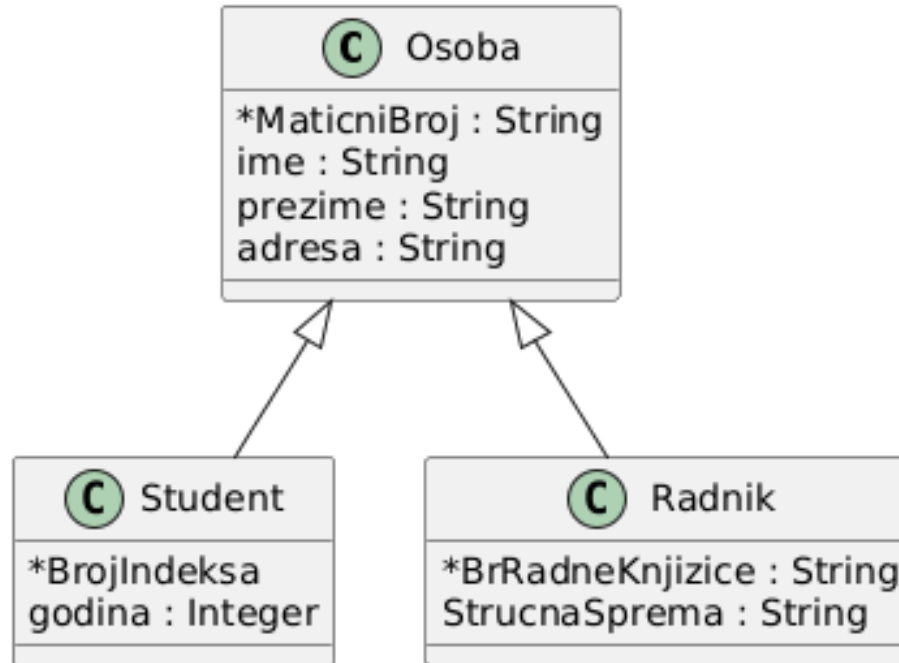
- Nasleđivanje je hijerarhijska veza između klasa
- Naziva se generalizacija ili specijalizacija, u zavisnosti u kom smeru se posmatra veza
- Podređeni element je u saglasnosti sa nadređenim elementom i poseduje sve njegove osobine, a uz to može posedovati i dodatne osobine
- Nadređena klasa je superklasa a podređena klasa je potklasa

# Natklasa Student i njene specijalizacije



- Klasa Student ima potklase:
  - Student osnovnih studija
  - Student specijalističkih studija
  - Student magistarskih studija
  - Student doktorskih studija

# Generalizacija(nasleđivanje)

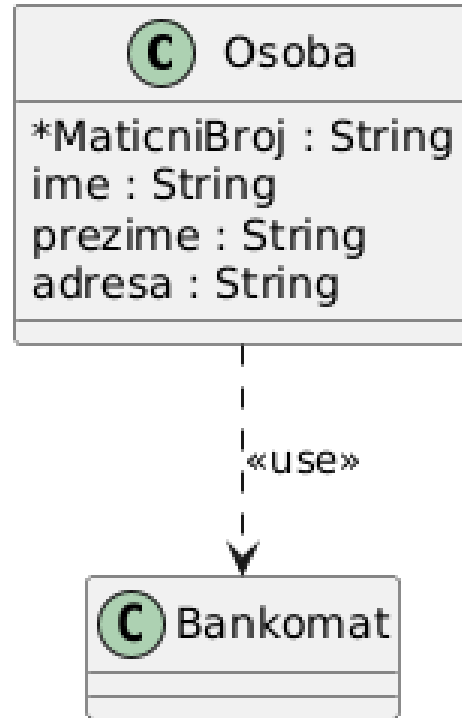


- Klase Student i Radnik specijalizuju klasu Osoba
- Klasa Osoba je njihova generalizacija

# Veza zavisnosti

- Veza zavisnosti (engl. dependency) označava situaciju u kojoj jedna klasa koristi drugu klasu za obavljanje neke funkcionalnosti, ali nije direktno zavisna od nje u smislu strukture (kao što bi bila kod agregacije ili kompozicije).
- Veza zavisnosti se prikazuje isprekidanom strelicom od klase koja koristi ka klasi koja se koristi. Na strelici piše **<<use>>**
- Klasa Osoba koristi bankomat (Bankomat) da bi podigla novac
- Osoba ne poseduje bankomat, već ga samo koristi kada joj zatreba

# Veza zavisnosti



# Objekatski dijagrami

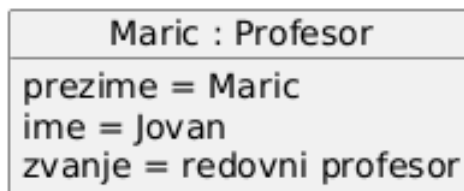
- Na objekatskom dijagramu prikazane su pojave klase, odnosno objekti i veze između njih
- Veze između objekata su pojave odgovarajućih veza između klasa kojima objekti pripadaju
- Prikazuju trenutno stanje sistema u određenom trenutku — konkretne instance klasa (objekte) i njihove međusobne veze
- Ilustruje primer strukture sistema, često kao dopuna klasnom dijagramu

# Objekti

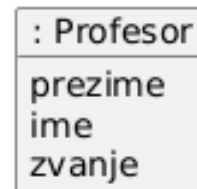
- Objekti odnosno pojave posmatrane klase sadrže sve atribute posmatrane klase i podržavaju operacije definisane posmatranom klasom
- Kada se kreira novi objekat on sadrži sve atribute posmatrane klase sa inicijalnim vrednostima
- Identitet objekta je jedinstven i automatski se dodeljuje svakom objektu pri njegovom kreiranju

# UML oznaka objekta

- Objekti se prikazuju pravougaonikom podeljenim u dva dela
- U prvom delu se navodi ime objekta i ime klase čiju pojavu pretstavlja objekat
- U drugom delu se navode atributi sa svojim vrednostima za posmatrani objekat
- Moguće je izostaviti ime objekta čime se specificira neimenovani objekat posmatrane klase



imenovani objekat



neimenovani objekat

# Primer objekatskog dijagrama

