

# Baze Podataka

## UML/EER dijagram

---

# **UML Modelovanje Podataka: 5 koncepta**

---

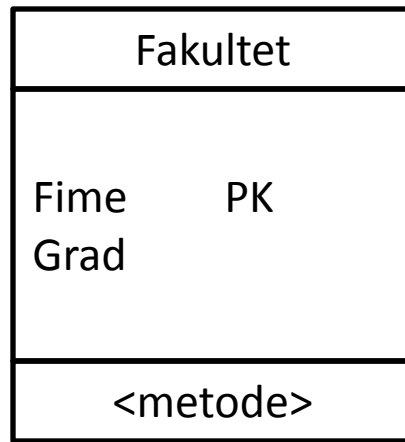
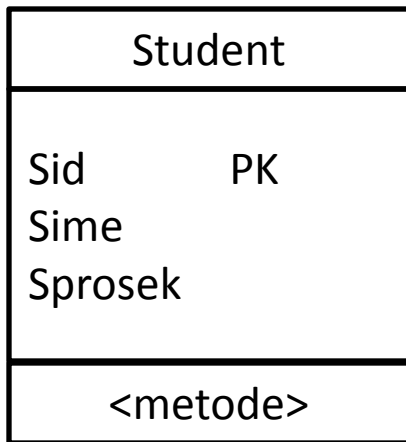
- (1) Klase (Classes)
- (2) Veze (Associations)
- (3) Klase povezivanja (Association Classes)
- (4) Podklase
- (5) Kompozicija & Agregacija

# UML modelovanje podataka: Klase

---

Imena, atributi, metode

*Kod modelovanja podataka: dodati "pk", metode se ne koriste*



# UML Modelovanje Podataka: 5 koncepta

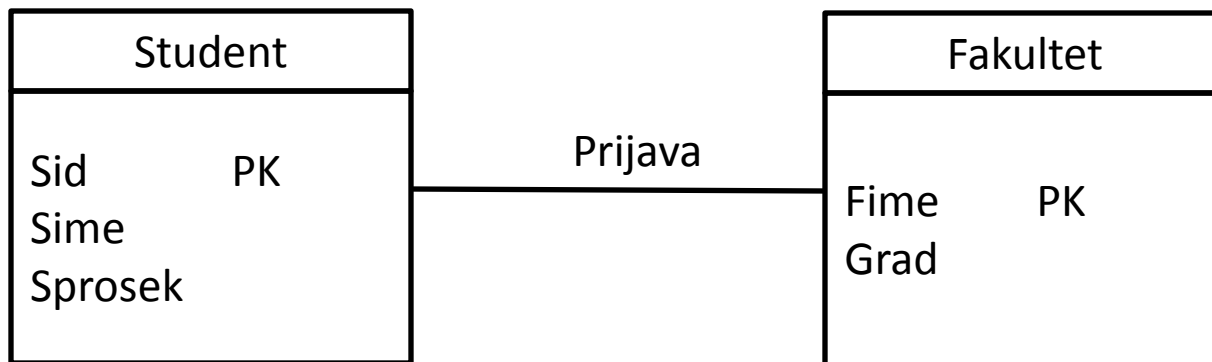
---

- (1) Klase (Classes)
- (2) Veze (Associations)
- (3) Klase povezivanja (Association Classes)
- (4) Podklase
- (5) Kompozicija & Agregacija

# UML modelovanje podataka: **Veze**

---

Veze između objekata dve klase

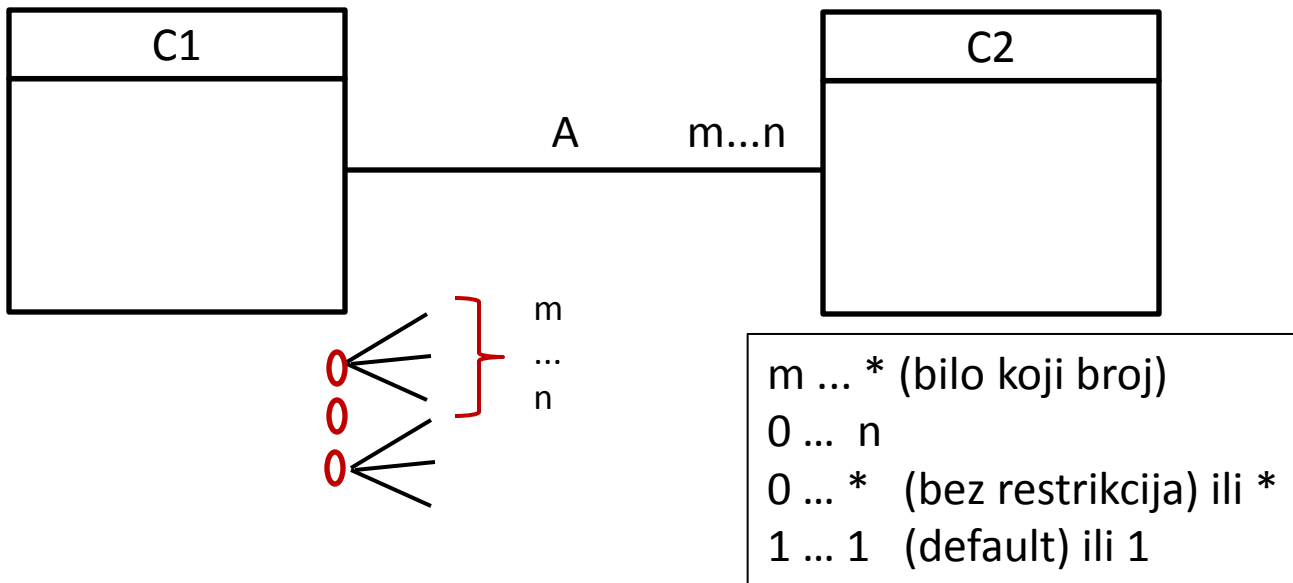


## Mnoštvo veza

Veze između objekata iz različitih klasa

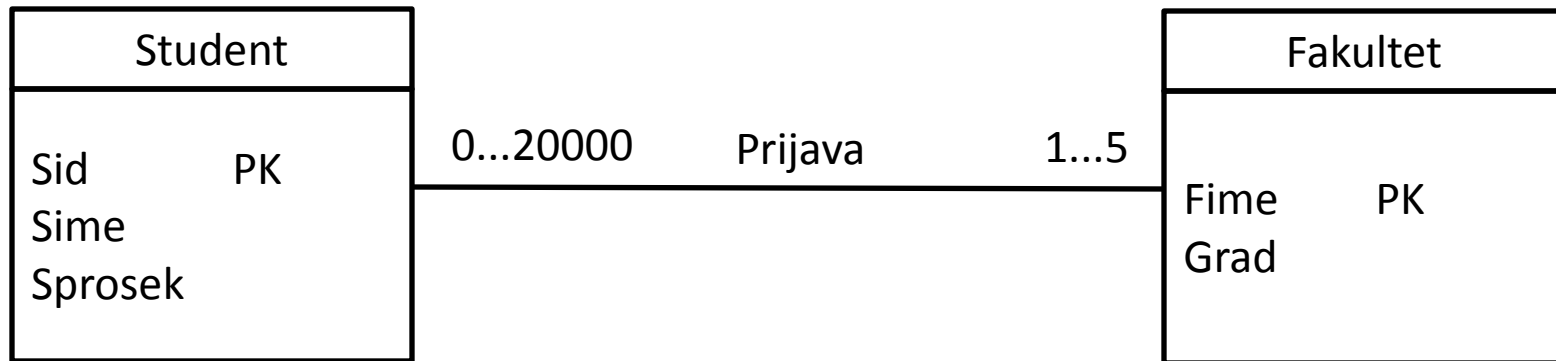
Koliko objekata jedne klase je povezano sa objektima druge klase

*Svaki objekat klase  $C_1$  je povezan sa najmanje  $m$  i najviše  $n$  objekata klase  $C_2$*



## Mnoštvo veza: Primer

*Studenti moraju da se prijave na bilo kom fakultetu ali ne mogu da se prijave na više od 5 fakulteta. Nijedan fakultet ne prihvata više od 20000 prijava.*



## Vrste relacija

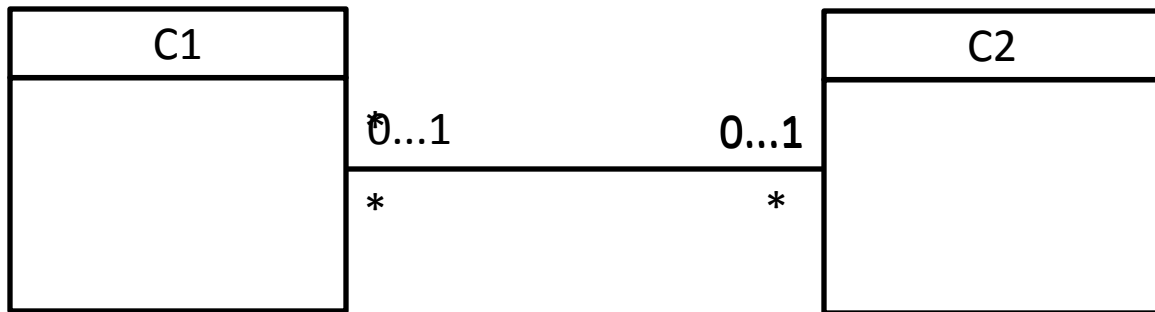
■ Jedan-prema-Jedan

■ Više-prema-Jedan

■ Više-prema-Više

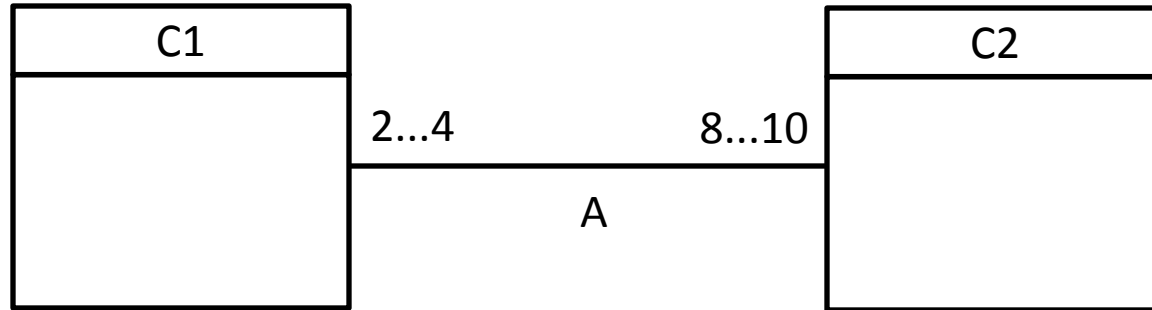
- Potpuna

- Default je potpuna 1...1 na obe strane





UML dijagram je prikazan na slici. Pretpostavka je da nijedna klasa nije prazna. Koji je minimalni broj objekata u obe klase?



2 u C1 i 8 u C2

8 u C1 i 2 u C2

16 u C1 i 16 u C2

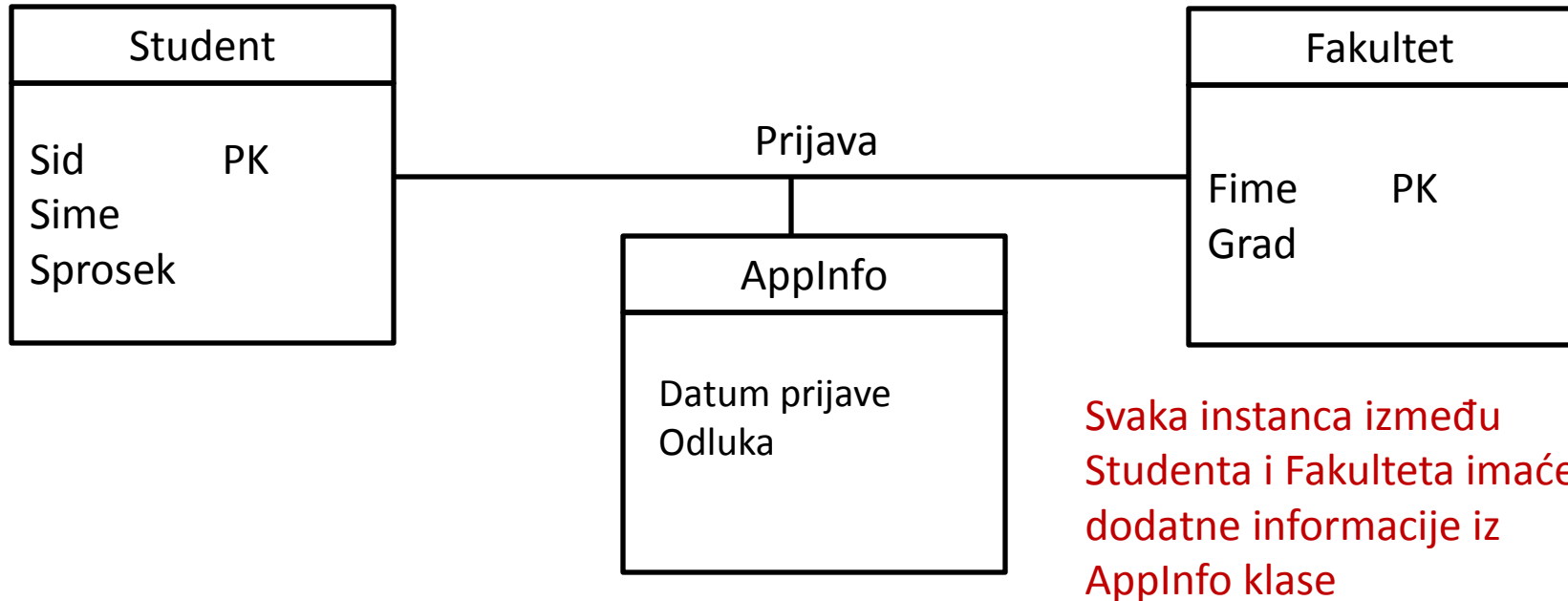
10 u C1 i 10 u C2

# UML Modelovanje Podataka: 5 koncepta

- (1) Klase (Classes)
- (2) Veze (Associations)
- (3) Klase povezivanja (Association Classes)
- (4) Podklase
- (5) Kompozicija & Agregacija

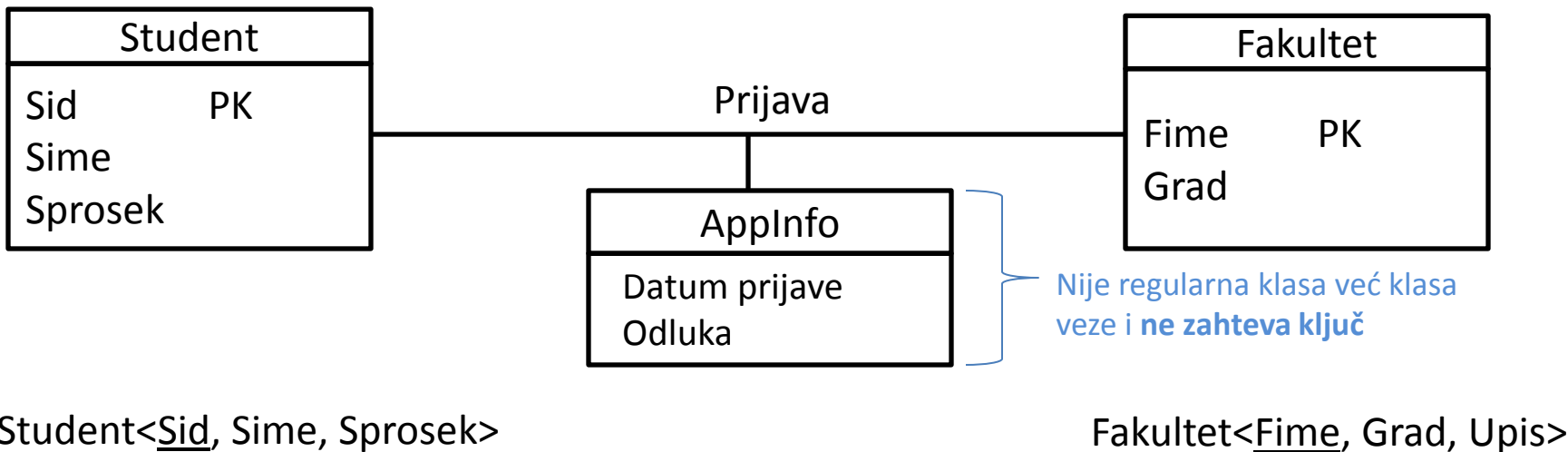
# UML modelovanje podataka: Klase povezivanja

Veza između objekata dve klase, sa atributima relacije



## UML -> Relacije: Klase veze

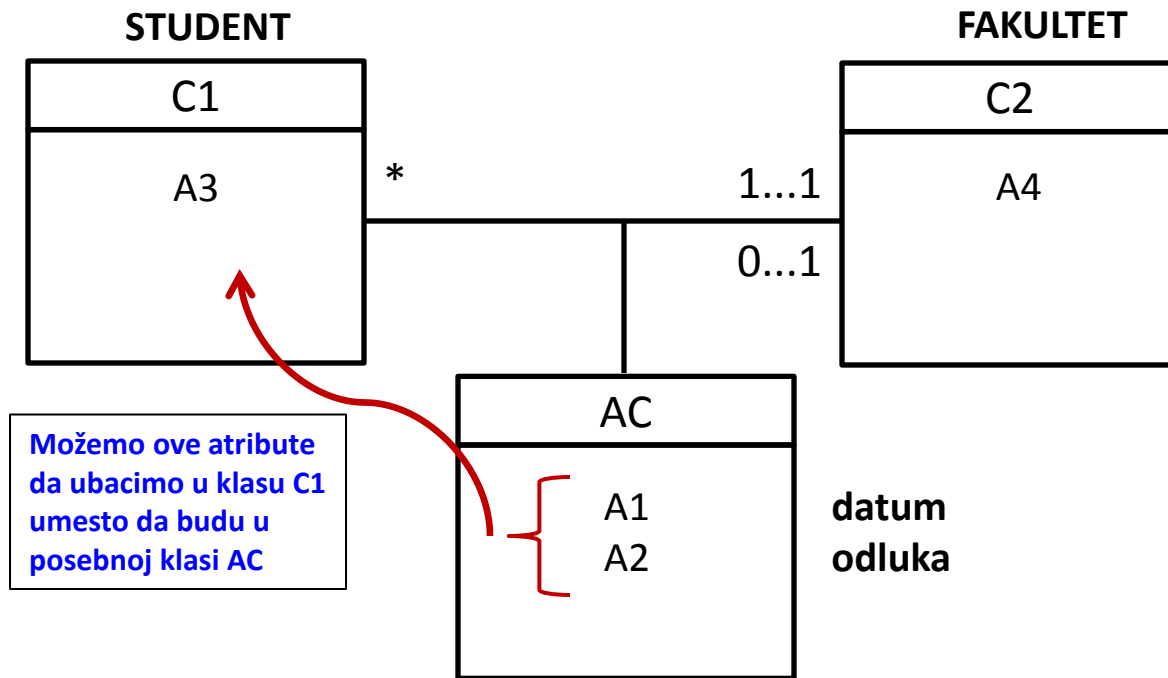
Atributi veze se dodaju relaciji koja opisuje vezu



Prijava<Sid, Fime, DatumPrijave, Odluka>

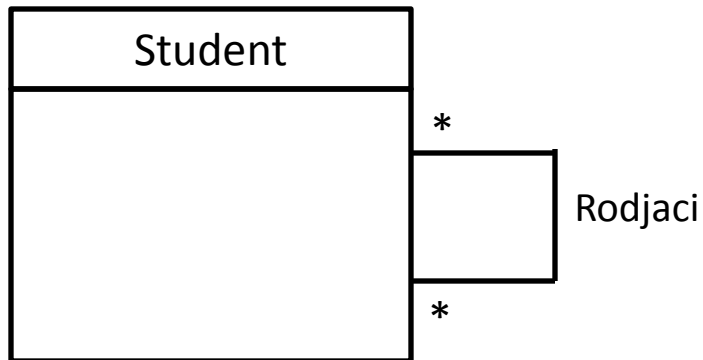
## Eliminisanje Klase povezivanja

Nije potrebna ukoliko je 0..1 ili 1..1

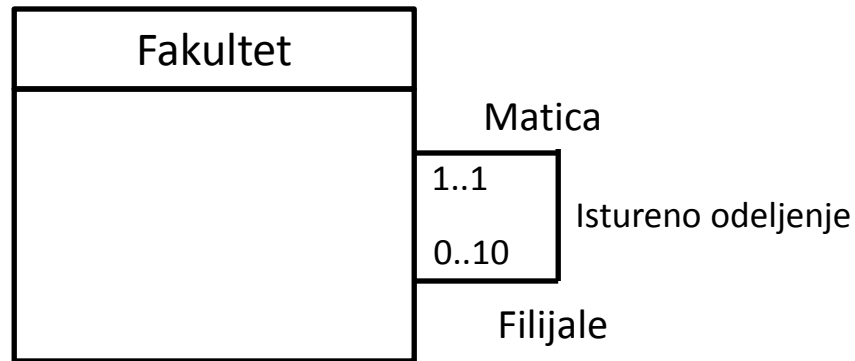


# Self-Associations

## Veze unutar iste klase



Svaki student može da ima više rođaka

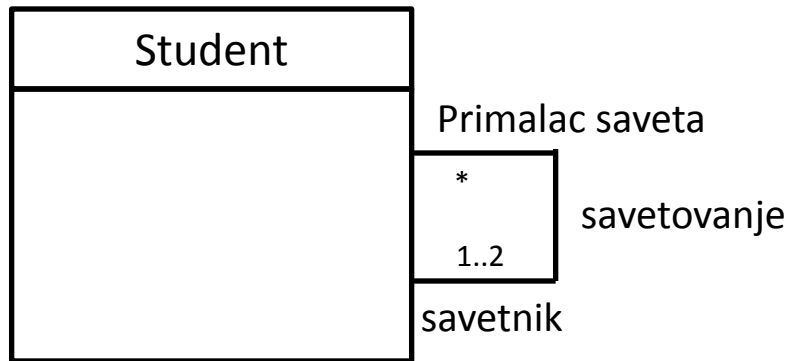


Matični fakultet može da ima najviše 10 isturenih odeljenja a svako istureno odeljenje samo jedan matični fakultet

## Veze unutar same klase - Zadatak

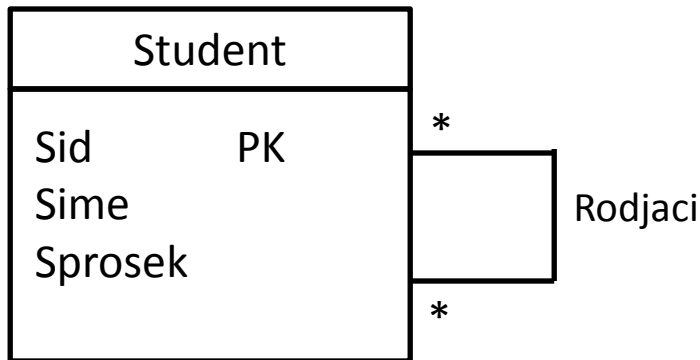
Odrediti relacije na osnovu sledećeg zahteva. Student mora da ima kolegu savetnika i može da ih ima najviše dvoje. Savetnik može da ima više studenata koje savetuje.

- a) primalac saveta 1..2 i savetnik \*
- b) savetnik 1..2 i primalac saveta \***
- c) primalac saveta 2..\* i savetnik 1..\*
- d) primalac saveta 1..\* i savetnik 2..\*



# Self-Associations

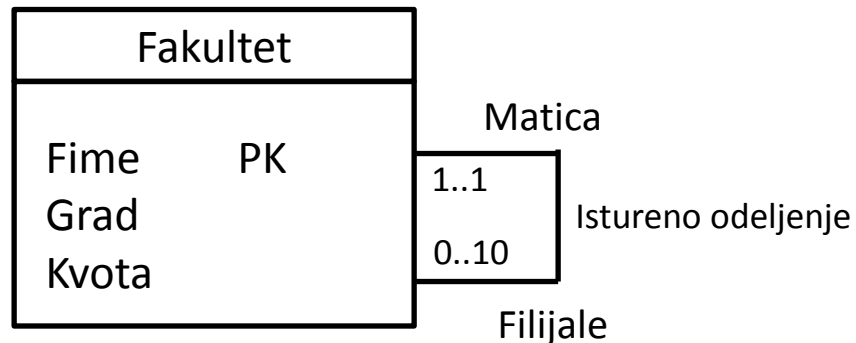
Svaki student može da ima više rođaka



Student < Sid, Sime, Sprosek >

Rodjaci < Sid1, Sid2 >

Matični fakultet može da ima najviše 10 isturenih odeljenja a svako istureno odeljenje samo jedan matični fakultet



Fakultet < Fime, Grad, Kvota >

Odeljenja < Matica, Filijala >

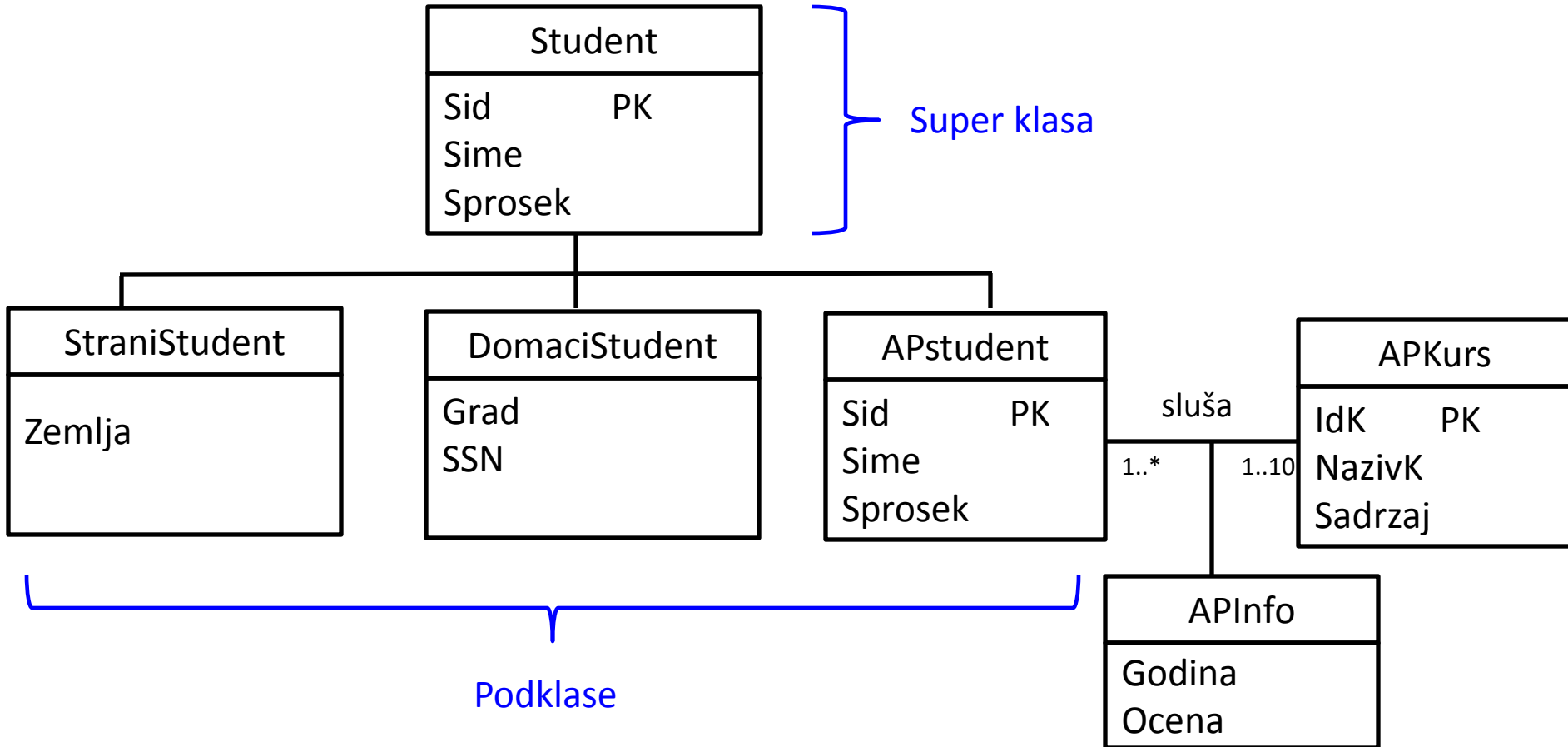


# UML Modelovanje Podataka: 5 koncepta

---

- (1) Klase (Classes)
- (2) Veze (Associations)
- (3) Klase povezivanja (Association Classes)
- (4) Podklase
- (5) Kompozicija & Agregacija

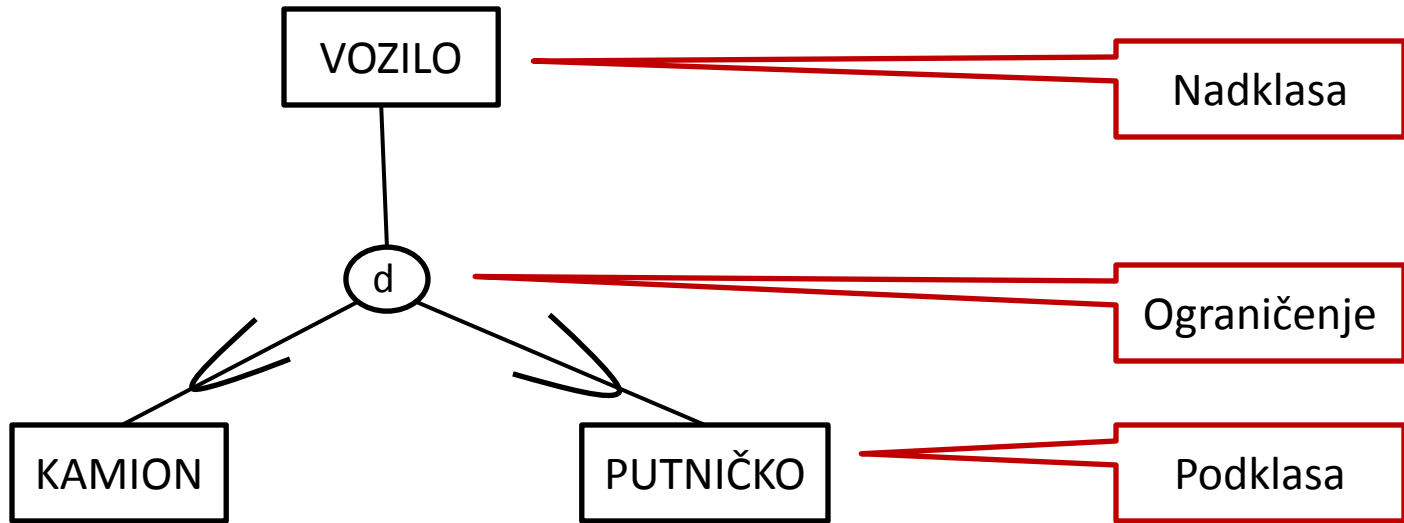
# UML modelovanje podataka: Podklase (subclasses)



# GENERALIZACIJA/SPECIJALIZACIJA

---

- Klasa VOZILO sa dve disjunktne specijalizacije KAMION i PUTNIČKO
  - Znači da vozilo može biti ili KAMION ili PUTNIČKO ali ne oba istovremeno








# Subclass Terminologija & Osobine

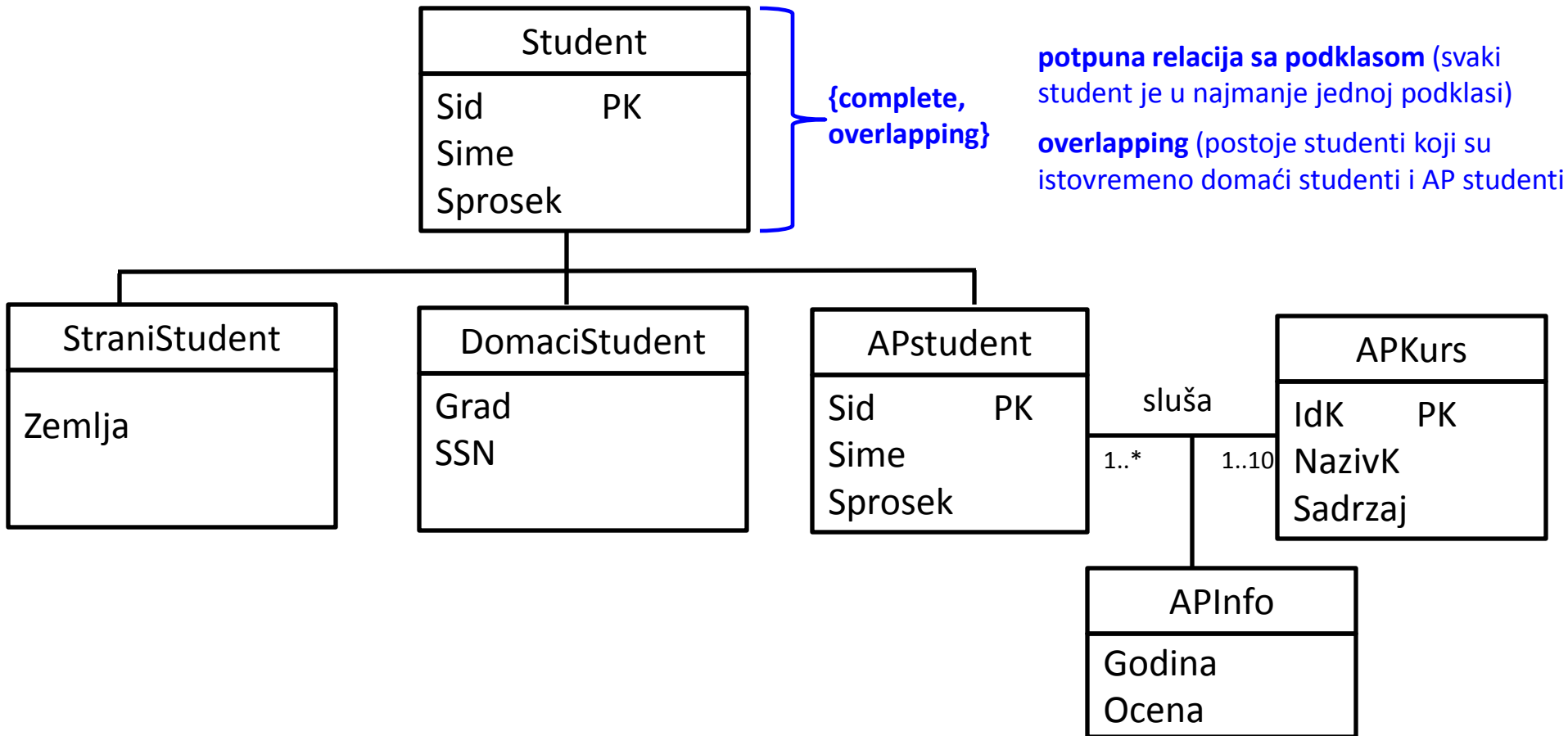
- *Superclass = Generalizacija*
- *Subclass = Specijalizacija*
- *Nepotpuna (Partial) vs. Kompletna*
  - *Kompletna - ako je svaki objekat u super klasi prisutan najmanje u jednoj podklasi*
  - *Nepotpuna – ukoliko postoje objekti kojih nema ni u jednoj podklasi*
- *Disjoint (Exclusive) vs. Overlapping*
  - *Disjoint - svaki objekat se nalazi samo u jednoj podklasi*
  - *Overlapping – postoje isti objekti koji se nalaze u više od jedne podklase*

# GRAFIČKA NOTACIJA EER DIJAGRAMA

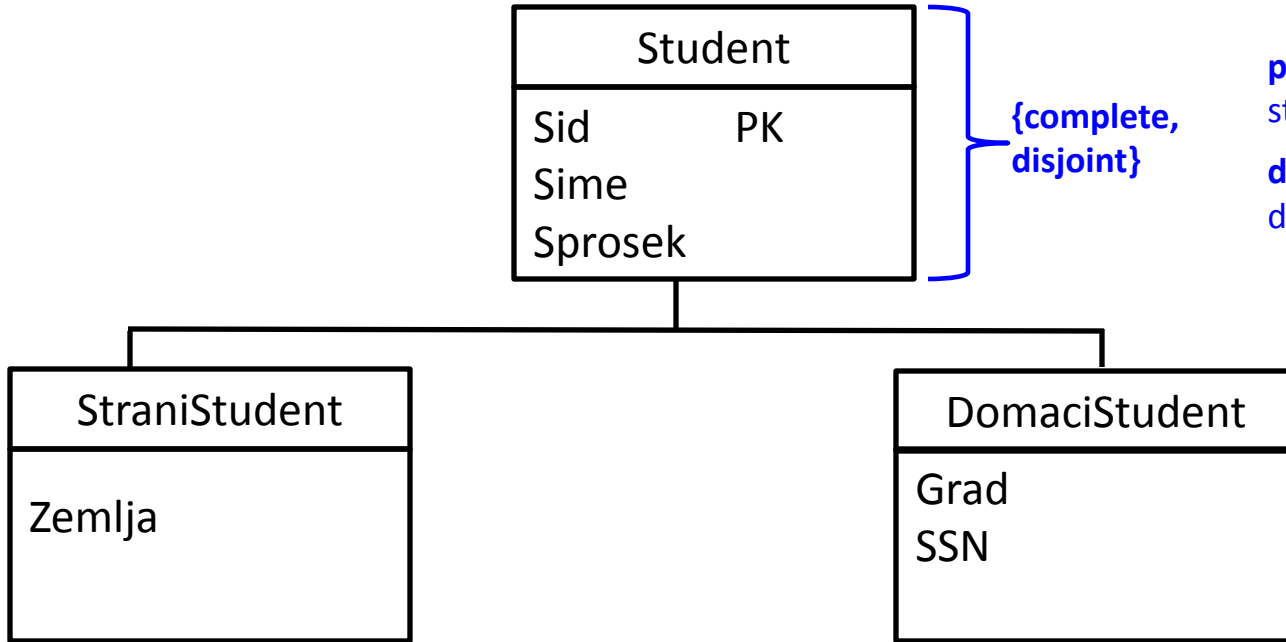
---

- EER dijagram koristi grafičku notaciju za ER dijagram i dodatke koji se predstavljaju na sledeći način.
- **Klasa** se predstavlja na isti način kao i entitet, pravugaonikom sa upisanom imenom klase
- Ograničenja
  - Oznake za učešće podklasa
    - **Disjoint** 
    - **Overlap** 
  - Oznake za kompletnost
    - **Totalna** 
    - **Parcijalna** 
  - Oznaka za **kategorije** 

# UML modelovanje podataka: Podklase (subclasses)



# UML modelovanje podataka: Podklase (subclasses)



**potpuna relacija sa podklasom** (svaki student je u najmanje jednoj podklasi)

**disjoint** (svaki student može da bude domaći ili strani)

## Podklase - Zadatak

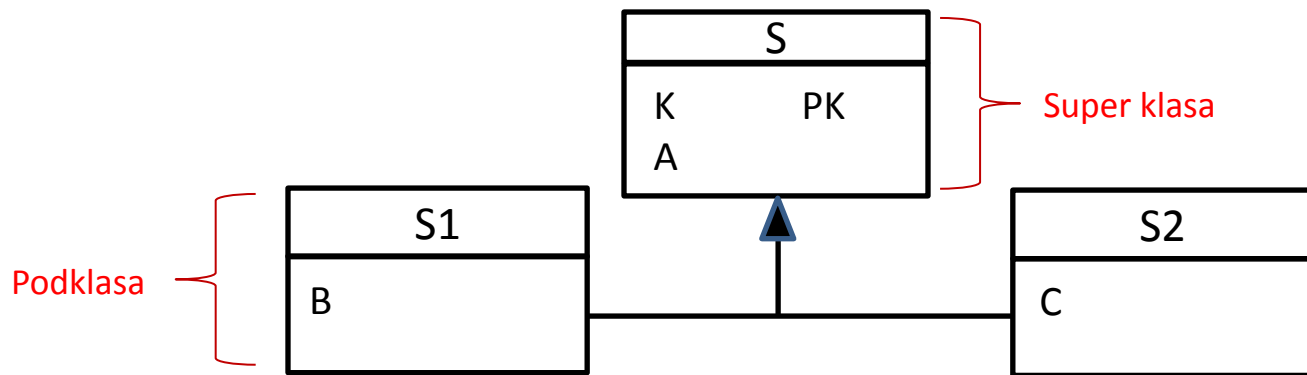
Posmatramo super klasu S sa tri podklase S1, S2 i S3. Pretpostavimo da je veza između super klase i podklasa potpuna (complete) i razdvojena (disjoint). Ako S ima 12 objekta koji je minimalni i maksimalni broj objekta u podklasi S1.

- a) minimum 4 i maksimum 4
- b) minimum 0 i maksimum 4
- c) minimum 0 i maksimum 12
- d) minimum 4 i maksimum 12



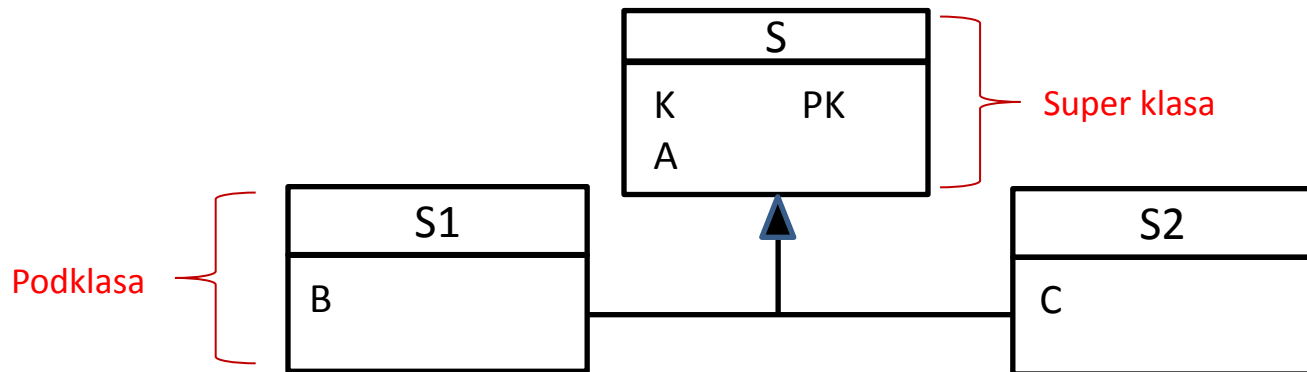
# Tri načina prevodjenja ovog UML dijagrama u relacije

- 1) Relacije podklase sadrže ključ superklase + svoje atribute.
- 2) Relacije podklase sadrže sve atribute
- 3) Jedna mega relacija koja sadrži sve atribute super klase i podklasa



# I varijanta (Podklase)

Relacije podklase sadrže ključ superklase + svoje atribute.



$S < \underline{K}, A >$

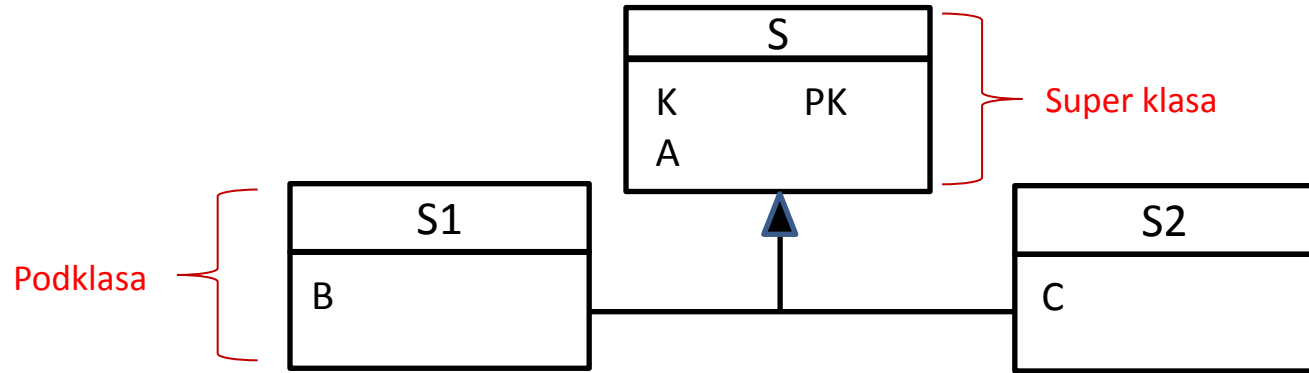
$S1 < \underline{K}, B >$

$S2 < \underline{K}, C >$

Relacije formirane na ovaj način zahtevaju dodatno povezivanje da bi se došlo do svih podataka

## II varijanta (Podklase)

Relacije podklase sadrže sve atribute



$S < \underline{K}, A >$

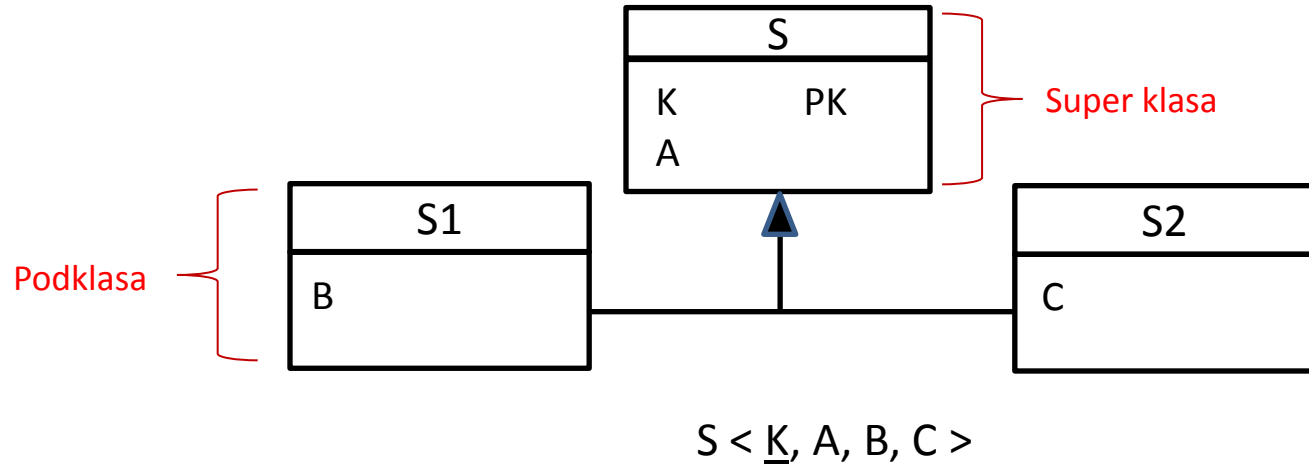
$S1 < \underline{K}, A, B >$

$S2 < \underline{K}, A, C >$

Super klasa sadržiće samo objekte koji se ne nalze u podklasama

# III varijanta (Podklase)

Jedna mega relacija koja sadrži sve atribute super klase i podklasa



U ovom slučaju mogu da se jave null vrednosti za one objekte koji su prisutni u klasi S1 a ne u klasi S2 ili u klasi S a ne u podklasama S1 i S2

## I varijanta (Podklase)

$S < \underline{K}, A >$   
 $S1 < \underline{K}, B >$   
 $S2 < \underline{K}, C >$

## II varijanta (Podklase)

$S < \underline{K}, A >$   
 $S1 < \underline{K}, A, B >$   
 $S2 < \underline{K}, A, C >$

Ukoliko se radi o potpunoj/disjoint varijanti, tada je najbolje drugo rešenje sa podklasama bez super klase

## III varijanta (Podklase)

$S < \underline{K}, A, B, C >$

Ukoliko je u bazi prisutno ogromno preklapanje (overlapping) tada se preporučuje Dizajn 3

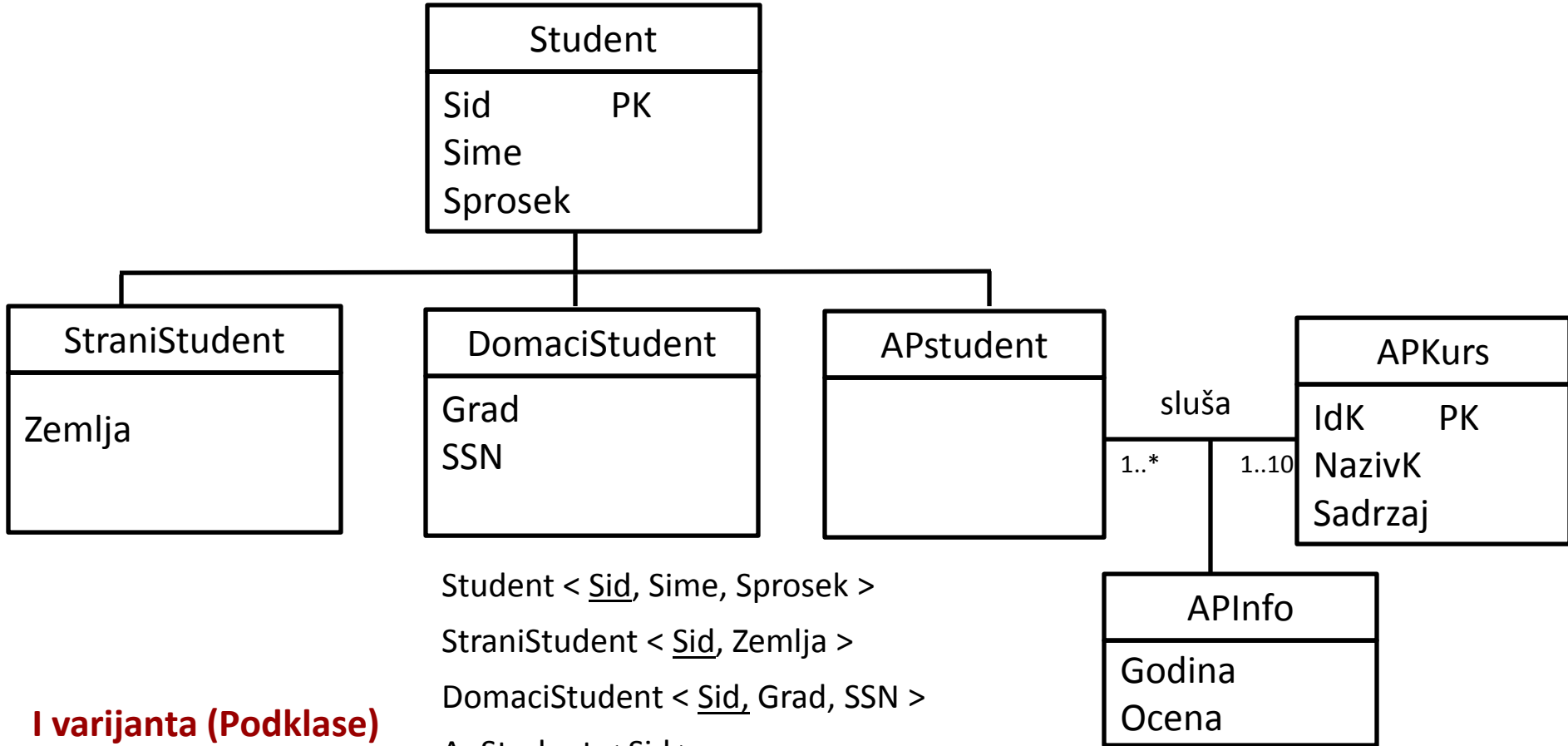
Izbor najboljeg prevođenja zavisi od osobina podklasa:

- **Overlapping vs Disjoint**

- *Disjoint* - svaki objekat se nalazi samo u jednoj podklasi
- *Overlapping* – postoje isti objekti koji se nalaze u više od jedne podklase

- **Potpuna vs Nepotpuna**

- *Potpuna* ako je svaki objekat u super klasi prisutan najmanje u jednoj podklasi
- *Nepotpuna* ukoliko postoje objekti kojih nema ni u jednoj podklasi



**I varijanta (Podklase)**

- Student < Sid, Sime, Sprosek >
- StraniStudent < Sid, Zemlja >
- DomaciStudent < Sid, Grad, SSN >
- ApStudent < Sid >
- ApKurs < IdK, NazivK, Sadržaj >
- Slusa < Sid , IdK, Ocena, Godina>

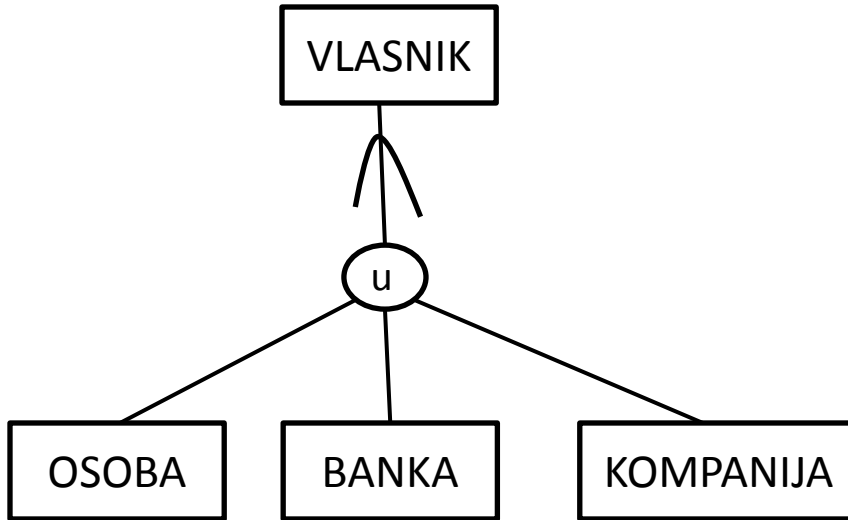
# UML Modelovanje Podataka: 5 koncepta

- (1) Classes (klase)
- (2) Associations (veze)
- (3) Association Classes (klase veze)
- (4) Subclasses (podklase)
- (5) Composition & Aggregation

# KATEGORIJE

---

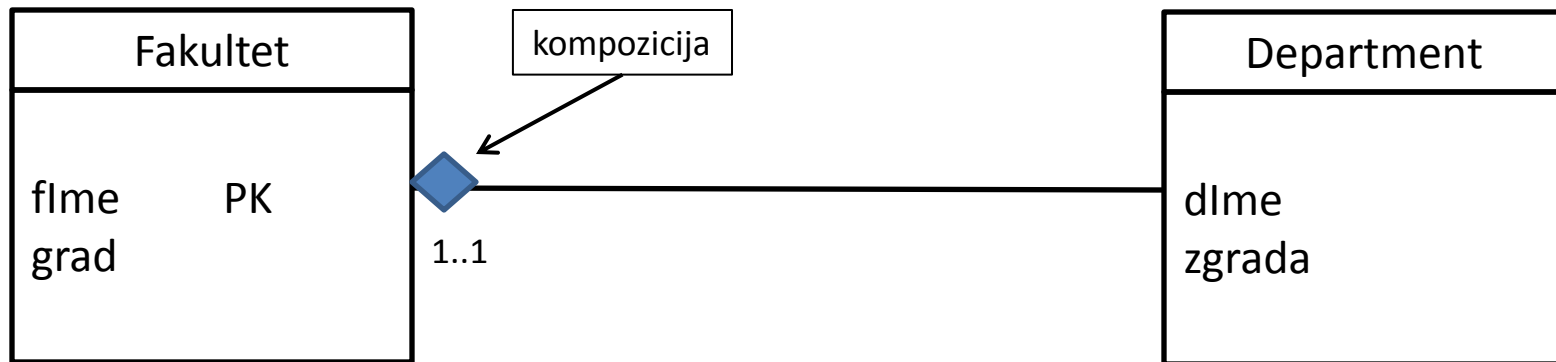
- Klasa VLASNIK vozila predstavlja kategoriju koja može biti jedna od klasa: OSOBA, BANKA ili KOMPANIJA.





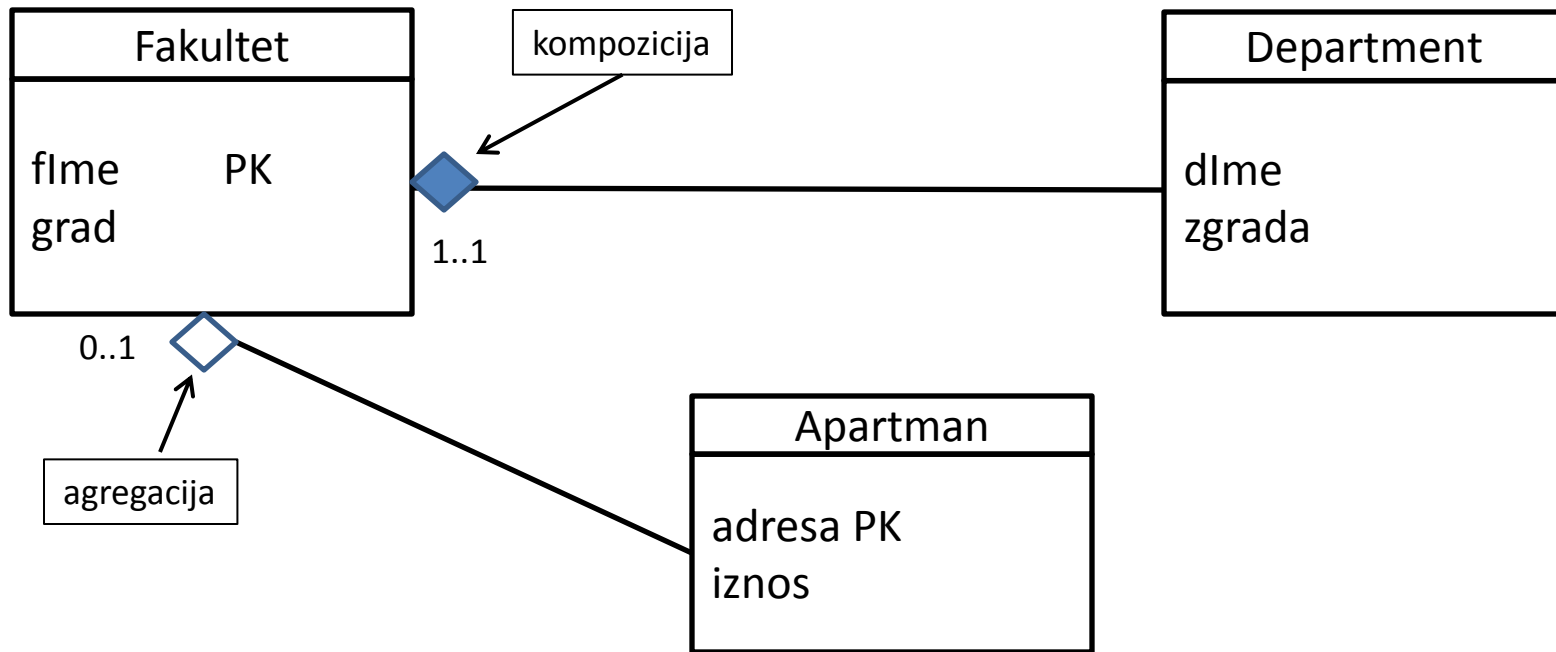
## UML Data Modeling: Composition (sastavljanje) & Aggregation

Objekti jedne klase pripadaju objektima druge klase



# UML Data Modeling: Composition (sastavljanje) & Aggregation

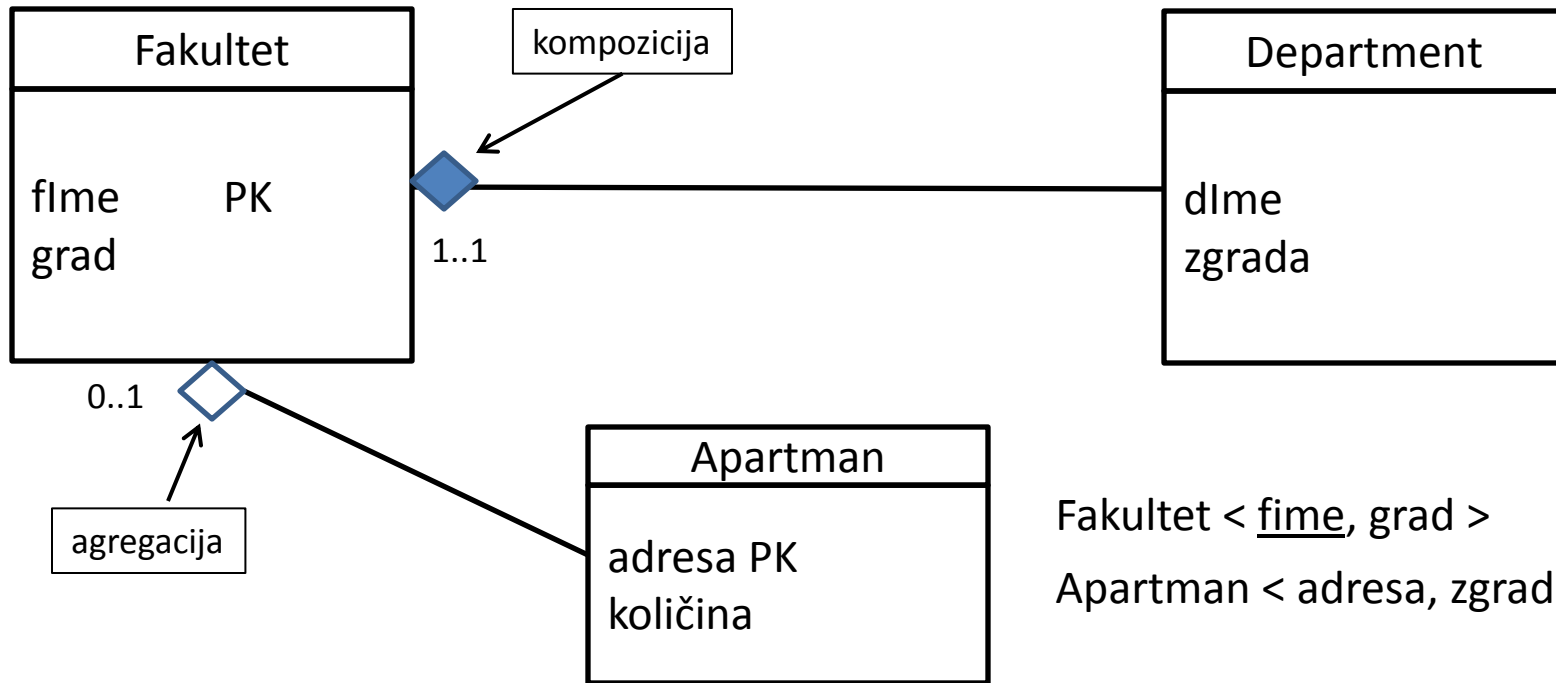
Objekti jedne klase pripadaju objektima druge klase



# Kompozicija & Agregacija

Fakultet < fime, grad >

Department < dlme, zgrada, fime >



Fakultet < fime, grad >

Apartman < adresa, zgrada, fime >