

# Osnovi softverskog inženjerstva

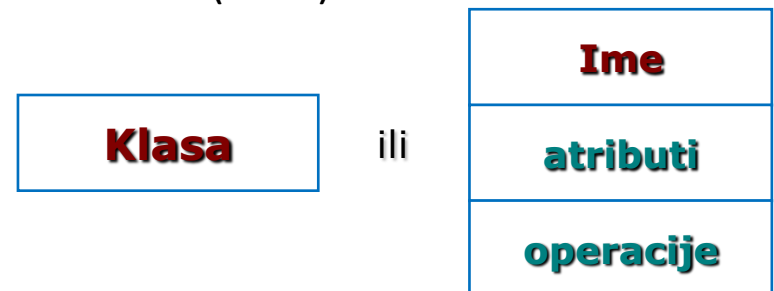
## P-07: Dijagram klasa

# Dijagram klasa

- Strukturni UML dijagram – modelovanje statičke strukture softverskog sistema
- Najčešće korištena notacija u softverskom inženjerstvu, koja se koristi za različite namjene i u različitim fazama razvoja softverskog sistema
- Omogućava prikaz klasa i njihovih međusobnih veza, bez mogućnosti modelovanja vremenski zavisnog ponašanja sistema

## Klase

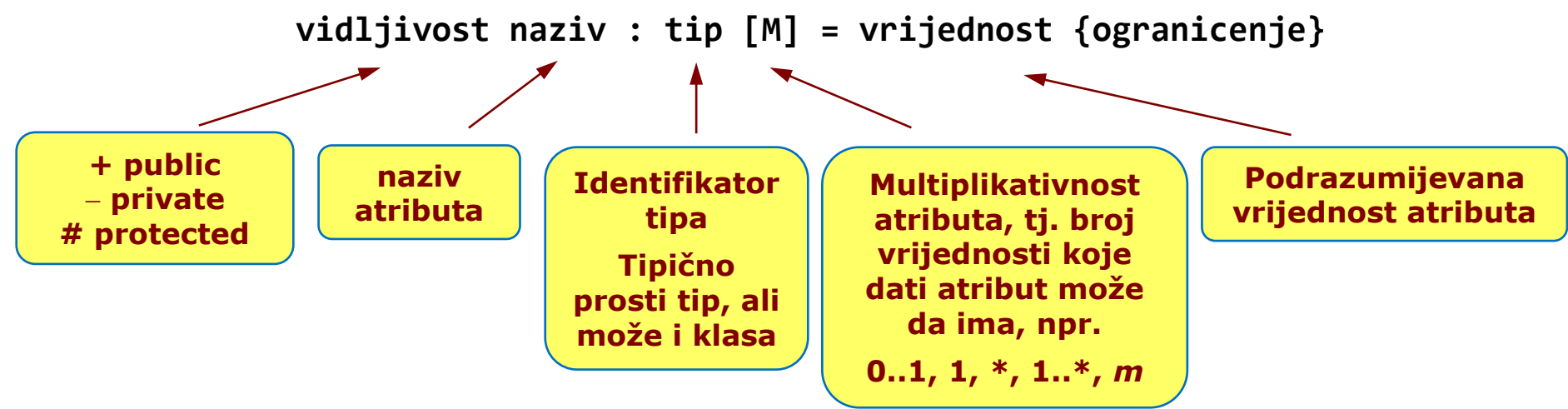
- **Klasa je deskriptor za skup entiteta sa sličnom strukturom, ponašanjem i vezama sa drugim entitetima**
- Priroda entiteta može biti različita (stvari, pojmovi itd.)
- Klasa se na dijagramu klasa prikazuje u obliku trodjelnog pravougaonika, gornja sekcija sadrži naziv klase, srednja atributi, a donja operacije
- Kad detalji nisu neophodni, atributi i operacije ne moraju da se prikazuju
- Naziv apstraktne klase se prikazuje kurzivom (italik)



# Dijagram klasa

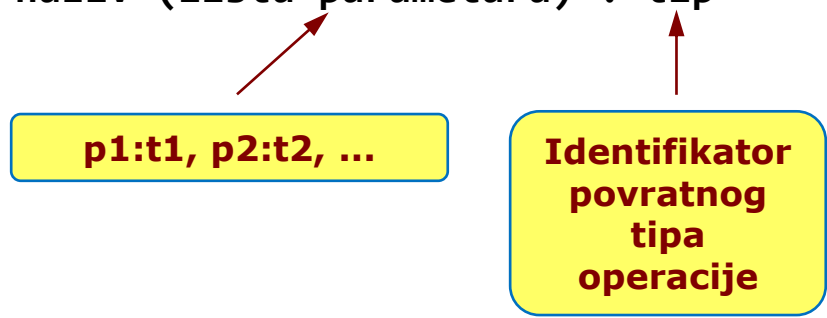
## Klase – specifikacija atributa i operacija

- Specifikacija atributa je oblika



- Specifikacija operacija je oblika

**vidljivost naziv (lista parametara) : tip**



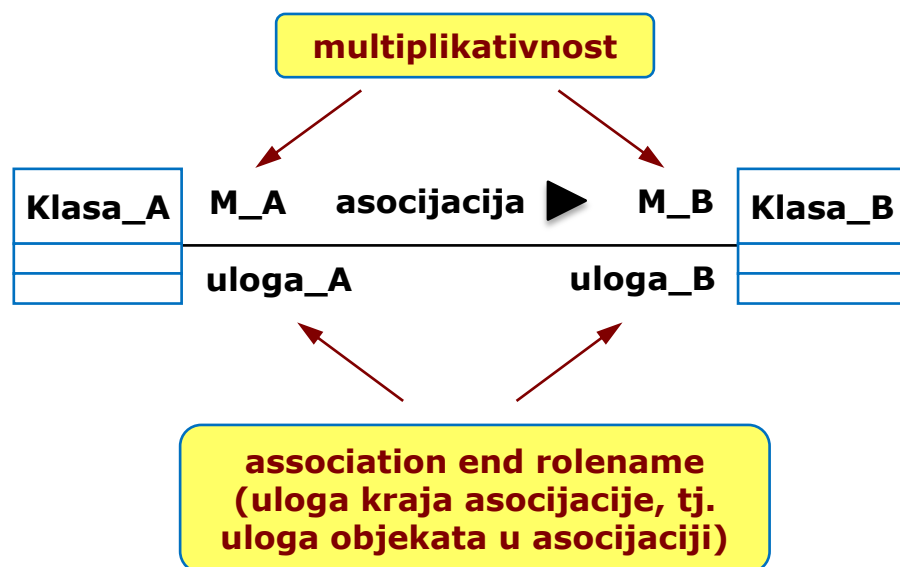
Osoba
jmb
ime
datumRodjenja
adresa [1..2]
telefon [*]

Predmet
idPredmeta
nazivPredmeta
ects =6 {ects>0}

# Veze između klasa

## Asocijacija

- Asocijacija je najopštiji tip veze između klasa, koji reprezentuje trajne veze između objekata
- Trajna veza između objekata je veza koja reprezentuje svojstva objekata, npr. osoba je rođena u nekom mjestu, osoba ima prebivalište u nekom mjestu, profesor predaje predmet, student je upisao studije na nekom studijskom programu itd.



**Naziv asocijacije oslikava vrstu veze između objekata, najbolje jedna riječ, npr. "prebivaliste" između klasa Osoba i Mjesto, ili "prijavljuje" između klasa Student i Ispit**

**Trougao ► ukazuje na smjer čitanja asocijacije (npr. "Motor ima rotor", a ne "Rotor ima motor").  
Navođenje simbola ► nije obavezno.**

**Objekti imaju uloge u asocijacijama, npr. u asocijaciji "prebivaliste", objekti klase Mjesto imaju ulogu "mjePreb" (mjesto prebivališta).  
Uloge se prikazuju po potrebi.**

**Multiplikativnost pokazuje koliko objekata date klase učestvuje u vezi sa jednim objektom druge klase, npr. M\_A pokazuje broj objekata klase A koji su u vezi sa jednim objektom klase B**

# Veze između klasa

## Asocijacija

**Multiplikativnost pokazuje koliko objekata date klase učestvuje u vezi sa jednim objektom druge klase, npr.  $M_A$  pokazuje broj objekata klase A koji su u vezi sa jednim objektom klase B**

simbol	značenje
0	nijedan
1	jedan
$m$	neki cijeli broj
0..1	nijedan ili jedan
$m..n$	najmanje $m$ , a najviše $n$
*	više (proizvoljno)
0..*	nijedan ili više
1..*	jedan ili više

Primjer:



Vežu između neke osobe i nekog mjesta po osnovu činjenice da je ta osoba rođena u datom mjestu reprezentujemo asocijacijom “rodjenje” između klase “Osoba” i “Mjesto”.

Objekti klase “Mjesto” u datoj asocijaciji igraju ulogu “mjeRod” (mjesto rođenja).

U jednom mjestu može biti rođeno više osoba pa je multiplikativnost objekata klase Osoba jednaka \*

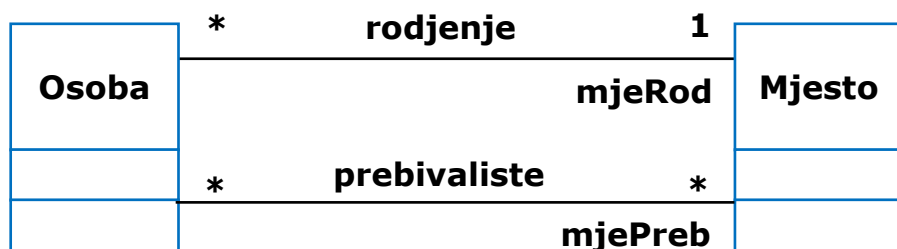
Svaka osoba mora i može biti rođena u jednom i samo jednom mjestu pa je multiplikativnost objekata klase Mjesto jednaka 1

# Veze između klasa

## Asocijacija

Između dvije klase može da postoji više asocijacija

Primjer:



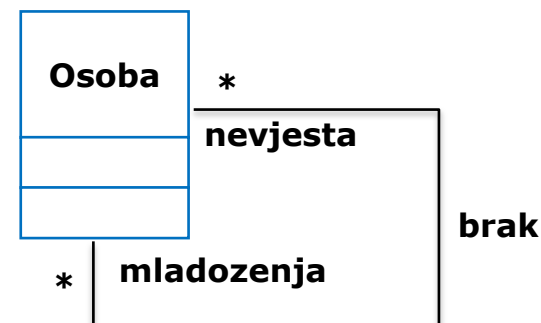
Pored asocijacije “rodjenje” (prethodni primjer), između objekata klase Osoba i klase Mjesto postoji i asocijacija “prebivaliste”, koja predstavlja vezu između neke osobe i nekog mjesta po osnovu činjenice da ta osoba ima (ili je imala) prebivalište u datom mjestu.

Objekti klase “Mjesto” u datoj asocijaciji igraju ulogu “mjePreb” (mjesto prebivališta).

U jednom mjestu prebivalište može da ima više osoba pa je multiplikativnost objekata klase Osoba jednaka \*, a svaka osoba tokom vremena može da ima prebivalište u više mjesta pa je multiplikativnost na strani Mjesto \*

REFLEKSIVNA ASOCIJACIJA – asocijacija između objekata iste klase

Primjer:



Između dvije osobe mogu da postoje različite veze (veza po osnovu braka, veza roditeljstva, veza starateljstva, veza nadređenosti/podređenosti itd)

U konkretnom primjeru, asocijacija “brak” reprezentuje činjenicu da između dvije osobe može da postoji veza po osnovu sklopljenog braka.

Tokom života, neka osoba može da bude u više različitih brakova pa je multiplikativnost krajeva \*

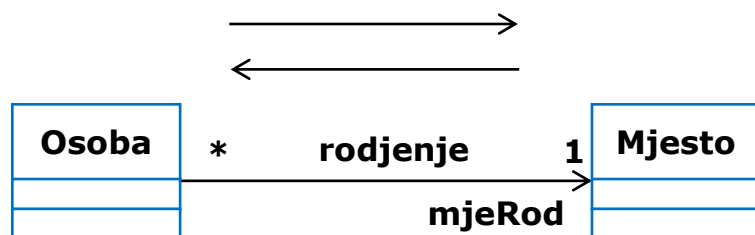
# Veze između klasa

## Asocijacija

- Asocijacije mogu biti:  
**neusmjerene (bidirekzione)**



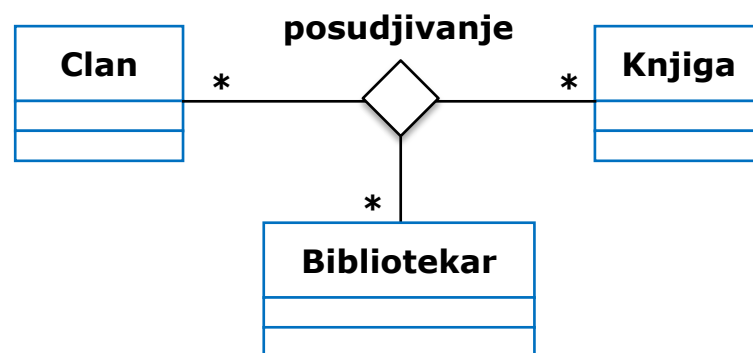
### usmjerene (unidirekzione)



**Ako je asocijacija usmjerena, tada su samo objekti jedne klase "svjesni" objekata druge klase, npr. svaka osoba ima atribut mjeRod koji reprezentuje mjesto rođenja date osobe, ali objekti klase Mjesto nemaju niz objekata sa podacima o osobama koje su rođene u datom mjestu.**

## N-arna asocijacija

- Asocijacije sa dva kraja (sve prethodne) su binarne asocijacije
- N-arna asocijacija je asocijacija koja ima više od dva kraja

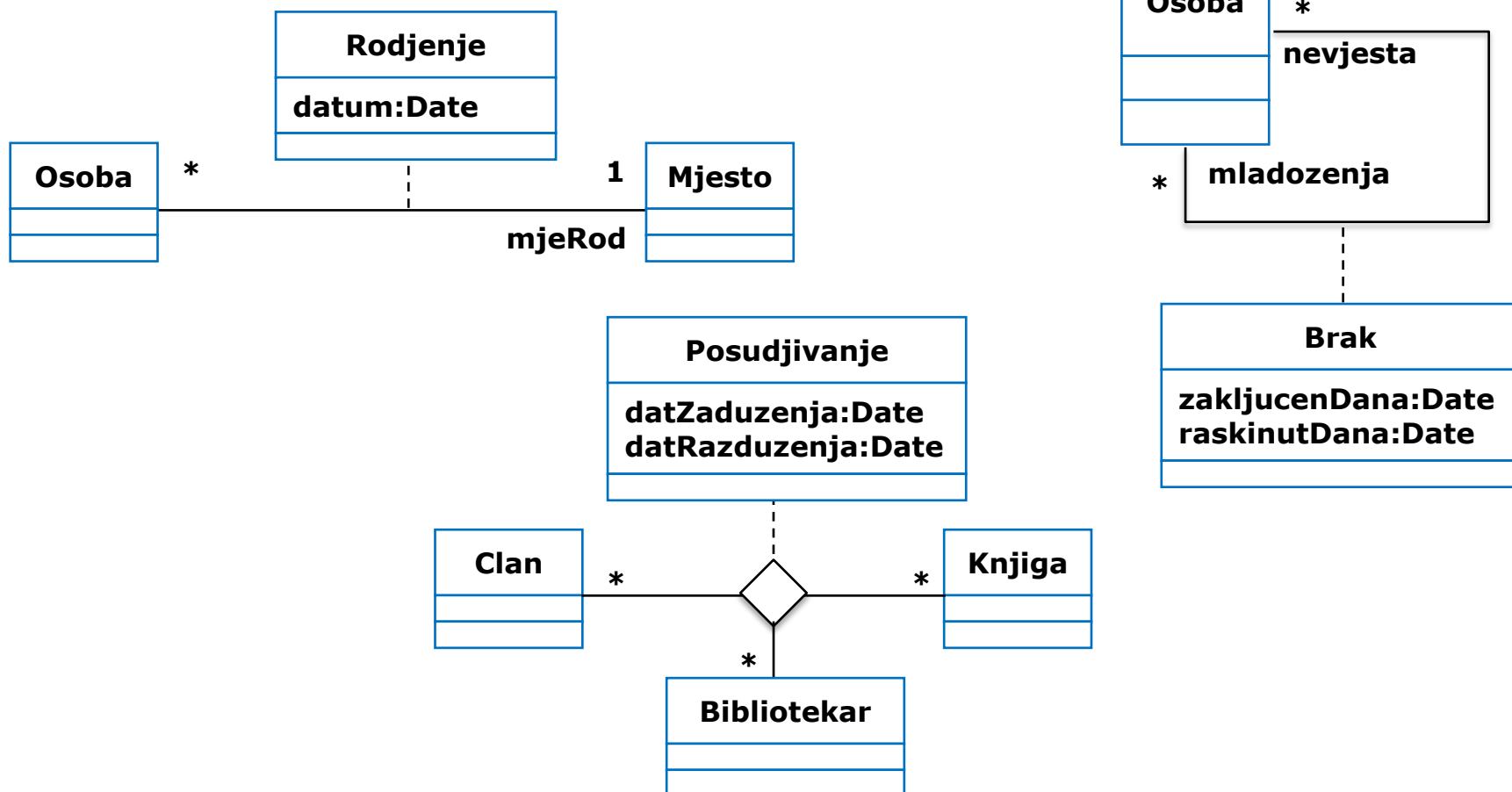


N-arna asocijacija "posudjivanje" reprezentuje činjenicu da je neki član posudio neku knjigu, pri čemu je neki bibliotekar evidentirao to posuđivanje. Svaki član može da ima više posuđivanja (\* na strani klase Clan), svaka knjiga može biti više puta posuđena (\* na strani klase Knjiga), a svaki bibliotekar može da evidentira više posuđivanja (\* na strani klase Bibliotekar)

# Veze između klasa

## Vezna (pridružena) klasa (Association class)

- Asocijacija koja ima atribute reprezentuje se klasom koja je pridružena toj asocijaciji

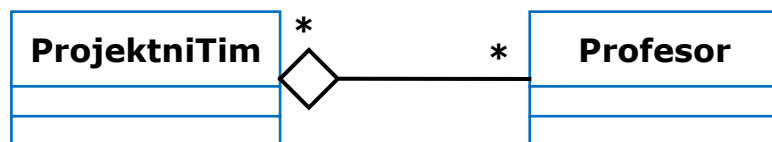
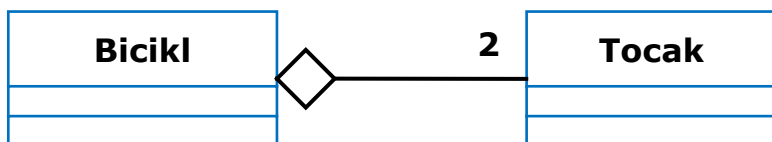




# Veze između klasa

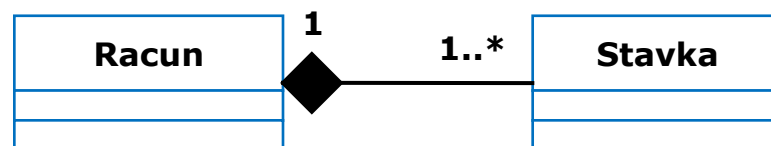
## Agregacija

- Asocijacija koja predstavlja odnos cjelina – dio (*part – whole*)
- "dijelovi" = entiteti koji mogu da imaju vlastiti identitet = mogu da egzistiraju nezavisno od "cjeline"
- Prikazuje se kao asocijacija koja na strani "cjeline" završava praznim romбом



## Kompozicija

- Snažniji (stroži) oblik asocijacije u odnosu na agregaciju
- "dijelovi" kompozicije ne mogu da egzistiraju nezavisno od "cjeline"
- Prikazuje se kao asocijacija koja na strani "cjeline" završava ispunjenim romбом



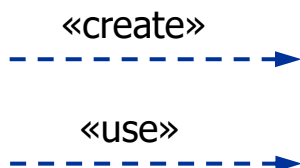
Stavka je jedan red na računu i reprezentuje činjenicu da je neka roba prodana (cijena, količina).

Svaka stavka mora da pripada nekom računu i ne može da postoji samostalno.

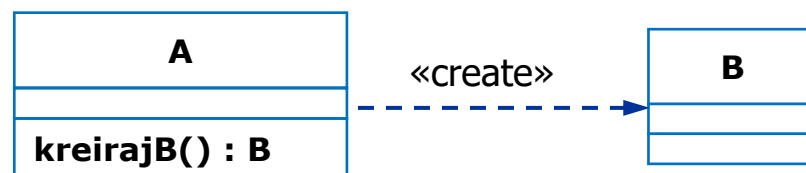
# Veze između klasa

## Zavisnost

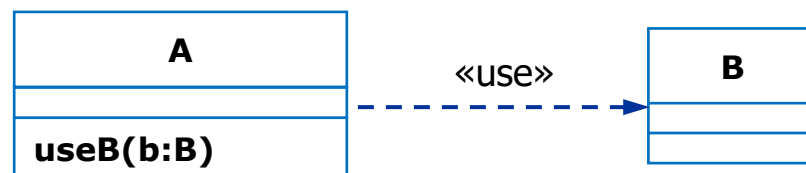
- Osim trajnih veza između objekata (zavisnosti), između objekata postoje i privremene veze
- Privremena veza između objekata reprezentuje se kao zavisnost (*dependency*) između korespondentnih klasa
- Privremene veze su veze kreiranja, korištenja, uništavanja, ...



**Operacija kreirajB() kreira i kao rezultat vraća objekat klase B**



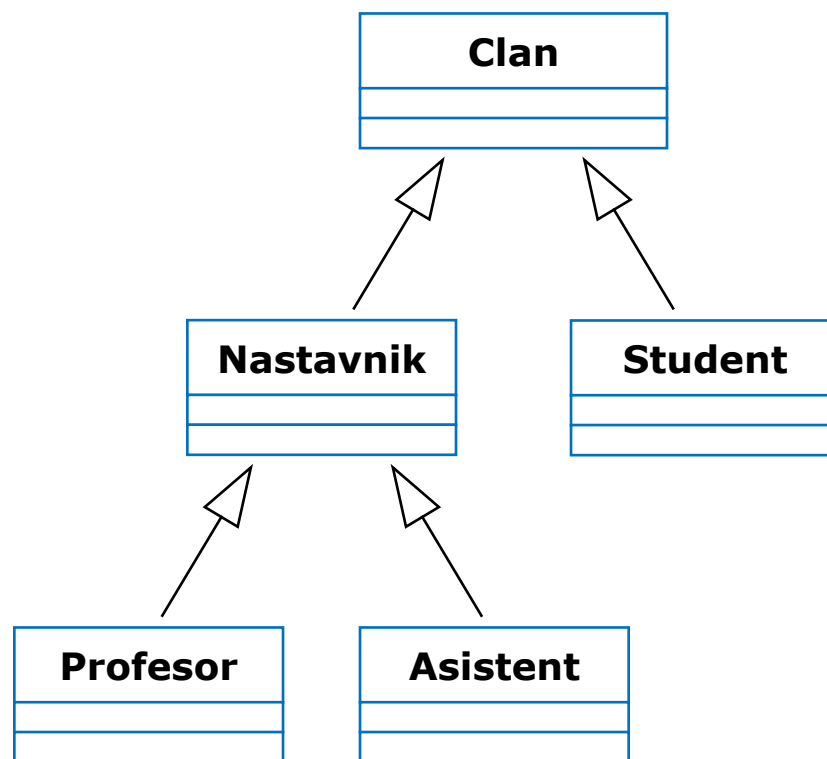
**Objekat klase B je argument poziva operacije useB(), što znači da objekti klase A koriste objekte klase B**



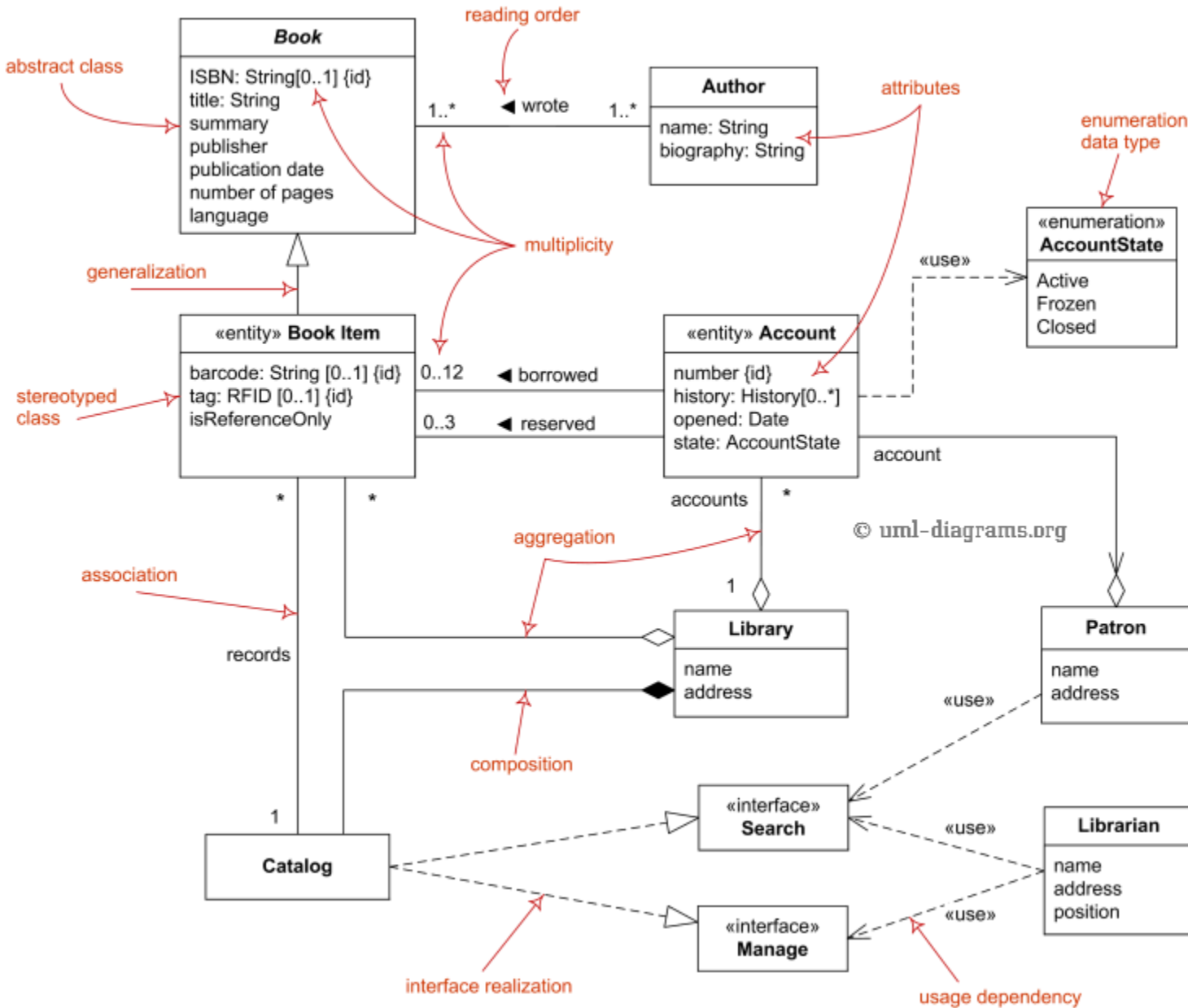
# Veze između klasa

## Generalizacija

- veza između uopštenijeg opisa nekog entitetskog skupa i specifičnijeg opisa na bazi uopštenog, kojim se uopšteni opis proširuje (specijalizuje)
- specijalizovani opis je konzistentan sa generalnijim opisom, jer uključuje sve karakteristike definisane opštijim opisom uz mogućnost definisanja dodatnih
- generalizovana specifikacija – **roditelj** (superklasa ili natklasa)
- specijalizovana specifikacija – **dijete** (subklasa ili potklasa)

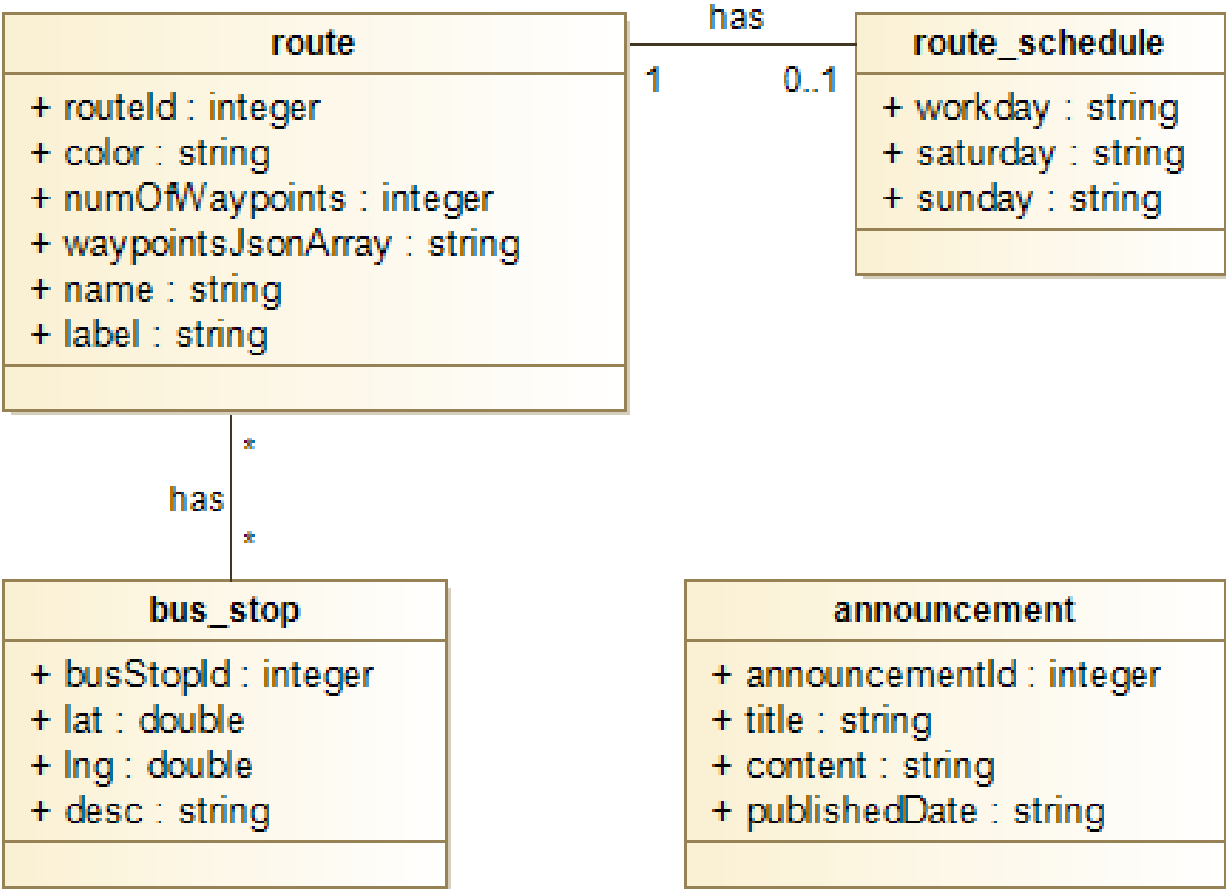


# Primjer dijagrama klasa



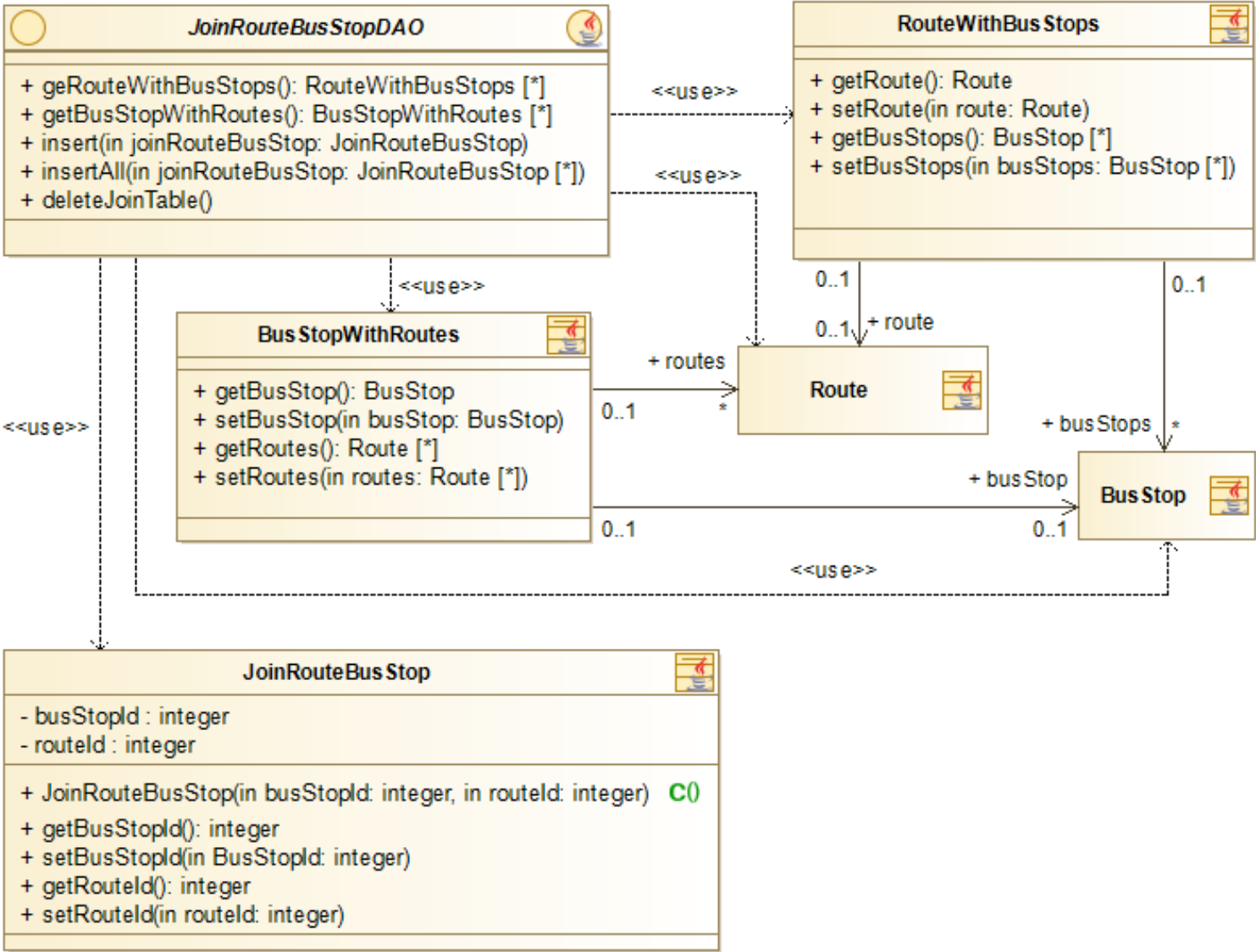
# Primjer dijagrama klasa

(dio konceptualnog modela baze podataka u BL BusTracker sistemu)



# Primjer dijagrama klasa

(struktura jednog modula u BL BusTracker sistemu)



# Kreiranje dijagrama klasa

- Ne postoji jedinstven i u potpunosti definisan način za kreiranje dijagrama klasa!
- Dijagram klasa kreira se ručno
- U literaturi postoje i neki prijedlozi, kao i neki alati, za (polu)automatsko generisanje dijagrama klasa na osnovu različitih artefakata:
  - na osnovu tekstualnih specifikacija,
  - na osnovu modela poslovnih procesa
  - ...

Za ilustraciju automatizovanog generisanja dijagrama klasa na osnovu modela poslovnih procesa pogledati AMADEOS sistem  
(<http://m-lab.etf.unibl.org:8080/amadeos/>)

Za ilustraciju automatizovanog generisanja dijagrama klasa na tekstualnih specifikacija pogledati TexToData sistem  
(<http://m-lab.etf.unibl.org:8080/Textodata/>)

## Kreiranje dijagrama klasa na osnovu tekstualne specifikacije

- Tekstualna specifikacija toka događaja u slučaju upotrebe je pogodan osnov za kreiranje dijagrama klasa
- **Analiza “imenica-glagol” (“noun-verb”)**
  - imenice su kandidati za objekte/klase
  - glagoli su kandidati za operacije (i asocijacije)
  - poznato kao **Abotova tehnika** (Abbott, 1983)

# Kreiranje dijagrama klasa

Neke heuristike za mapiranje elemenata tekstualne specifikacije u odgovarajuće koncepte u dijagramu klasa

(Abotova tehnika)

Tekstualna specifikacija	Korespondentni koncept u modelu	Primjer
vlastita imenica	objekat	Marko, preferans ...
opšta imenica	klasa/atribut	klijent, račun, transakcija ...
radni/prelazni glagol	operacija/asocijacija	ubacuje, ukucava, provjerava ...
neprelazni glagol	operacija/atribut/...	zavisi od ...
glagol biti/jeste	nasljeđivanje	teretni voz je(ste) voz za prevoz ...
glagol imati/sastojati/činiti	agregacija/kompozicija	račun ima stavke, voz se sastoji od ...
modalni glagol	ograničenje	PIN mora biti 4-cifren
pridjev	atribut	ispravan, nov, promijenjen, izdat ...



# Kreiranje dijagrama klasa

- Specifikacija zahtjeva je dokument koji sadrži zahtjeve organizovane po slučajevima upotrebe, pri čemu za svaki slučaj upotrebe postoji odgovarajuća specifikacija
- Na osnovu svake pojedinačne specifikacije slučaja upotrebe može se kreirati odgovarajući dijagram klasa
- Svaki od tih dijagrama klasa predstavlja samo dio cjelokupnog dijagrama klasa koji reprezentuje model čitavog sistema

