Projektovanje baza podataka

Model klasa

14. april 2026.

Sadržaj

1. Model klasa podataka
2. Dijagram klasa podataka

Napomena

Slajdovi su nastali obradom slajdova prof. dr Saše Malkova za predmet Projektovanje baza podataka

Sadržaj

1 Model klasa podataka

2 Dijagram klasa podataka

UML i modeliranje baza podataka

- Motivacija i cilj:
  - UML je opšte poznat jezik za modeliranje softvera
  - Približavanje jezika za modeliranje podataka jeziku za modeliranje informacionih sistema i programa
- U više navrata je predlagano da se napravi poseban "profil" za modeliranje podataka
  - Postoje nezvanične radne verzije 2001-2005, ali bez zvaničnih koraka nakon toga
  - Ispostavilo se da nije neophodan – sasvim dobro je i bez njega

Model klasa podataka

- UML dijagram klasa već opisuje strukturu klasa
- Ideja je da se isti dijagram iskoristi za modeliranje podataka:
  - bolje predstavlja semantiku odnosa od dijagrama tabela RM
  - bolje predstavlja semantiku odnosa čak i od pravog ER dijagrama
  - ima manje nedostatke koji mogu da se prevaziđu

Razlika u konceptu modela

- Modeli klasa i relacija se konceptualno razlikuju:
  - relacije (tabele) su skupovi podataka
    - kao i skupovi entiteta i odnosa u modelu ER
  - klase su tipovi podataka

Razlika u konceptu modela

- Može da se prevaziđe u praksi:
  - a) ako nam je potrebno da modeliramo više skupova istog tipa, možemo da uvedemo više naslednika klase koja određuje tip
    - to nije semantički sasvim ispravno, ali se tako jasno vidi "sličnost" tipova pojedinačnih skupova entiteta
    - bazna klasa, koja služi samo kao tip, može da se posebno označi, npr. kao "apstraktna"
  - b) svaka "klasa"može da se označi – da bi se znalo da li je tip ili skup
    - tipovi/klase se označavaju sa *«type»*
    - skupovi entiteta se označavaju sa *«persistent»*
    - to dodatno opterećuje dijagram, pa se radi samo kada je neophodno
  - v) problem se ignoriše i smatra da je klasa *skup* a ne *tip*
    - u redu je samo za neke manje domene, gde se ne očekuje više skupova istog tipa

Sadržaj

1 Model klasa podataka

2 Dijagram klasa podataka

UML dijagram klasa podataka

- Za projektovanje modela baze podataka koristi se tzv. **konceptualni dijagram klasa**
  - u UML-u se obično naziva ***dijagram klasa domena***
  - u radu sa BP se naziva i ***dijagram klasa podataka***
- Za razliku od uobičajenog dijagrama klasa:
  - u prvom planu su atributi i odnosi
  - ponašanje se skoro potpuno zanemaruje
- Za specifične oznake koriste se različite dopune UML-a

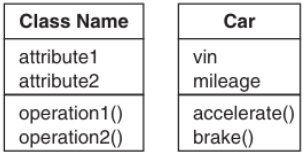
Osnovna notacija dijagrama klasa

- UML ikonica za klasu je **pravougaonik**.
- Kada se klasa prikazuje sa atributima i operacijama, pravougaonik je podeljen na tri horizontalna odeljka.
  - Gornji odeljak sadrži ime klase.
  - Srednji odeljak sadrži nazive atributa.
  - Donji odeljak sadrži nazive operacija.
- Pri modelovanju baza podataka klasa se prikazuje samo sa imenom klase i odeljkom za atribute.

UML dijagram klasa

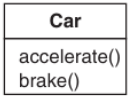
Classes

Notation and Example

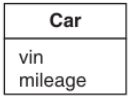


Notational Variations

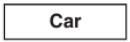
Emphasizing Operations



Emphasizing Attributes



Emphasizing Class



Odnosi

- Asocijacija
- Agregacija
- Kompozicija
- Nasleđivanje

u osnovi nije drugačije nego kod uobičajenih dijagrama klasa

Asocijacija

- Odnos između dve klase
- Može da bude:
  - funkcionalan – "uradi nešto za mene"
  - strukturalan – "budi nešto za mene"
- U kontekstu modela podataka važnija je i skoro isključivo se koristi ***strukturalna*** asocijacija
- Označava da jedna klasa "zna" za neke objekte druge klase i na neki način upravlja njima ili komunicira sa njima
  - u kontekstu podataka, omogućava da se komunicira sa njima

Agregacija

- Posebna vrsta asocijacije
- Implicira da se objekat jedne klase *sastoji* od (jednog ili više) objekata druge klase
- Predstavlja slabiji oblik strukturalne asocijacije:
  - isti deo može da bude *sadržan* u više složenih objekata, tj. da bude deljen
  - ako se složeni objekat obriše, delovi mogu da nastave da postoje
- Ponekad se praktično svaka strukturalna asocijacija, čija je kardinalnost "1-više", posmatra kao forma agregacije
  - ali to nije dobro raditi ako semantika nije odgovarajuća

Kompozicija

- Predstavlja jači oblik strukturalne asocijacije:
  - uvek je binarni odnos
  - odnos celina/deo
  - jedan deo može da pripada samo jednom složenom objektu
  - ako se složeni objekat obriše uobičajeno ponašanje je da se obrišu i svi delovi
- Odgovara konceptu "slabe entitske relacije" ER-modela
  - elementi kompozicije odgovaraju slabom entitetu

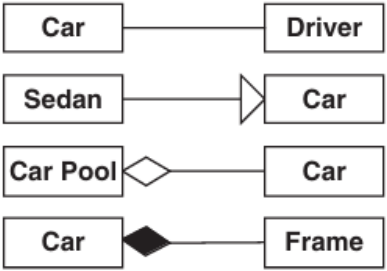
Nasleđivanje

- Odnos koji je suštinski za OO modeliranje, pa time i za model klasa podataka
- Osnovna klasa predstavlja opštiji slučaj izvedenih klasa
  - "*generalizacija*"
- Izvedene klase predstavljaju posebne slučajeve bazne klase
  - "*specijalizacija*"
- Relacioni model nema, a model entiteta i odnosa ima odgovarajući koncept
- Kao i u vezi hijerarhija u modelu ER, hijerarhije u dijagramima klasa podataka često se grade na osnovu strukture, a ne na osnovu ponašanja

Predstavljanje odnosa

Relationships

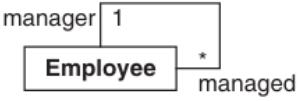
- Association
- Generalization
- Aggregation
- Composition



Predstavljanje odnosa - stepen

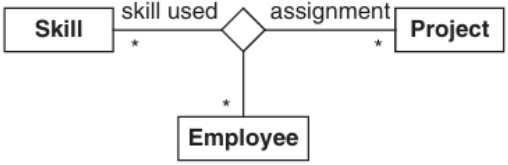
Degree

reflexive association



binary association

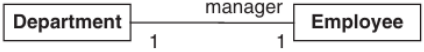
ternary association



Predstavljanje odnosa - kardinalnost

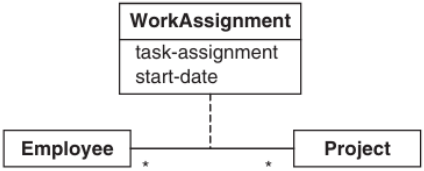
Multiplicities

one-to-one



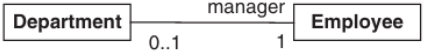
one-to-many

many-to-many

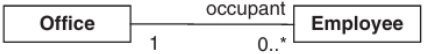


Existence

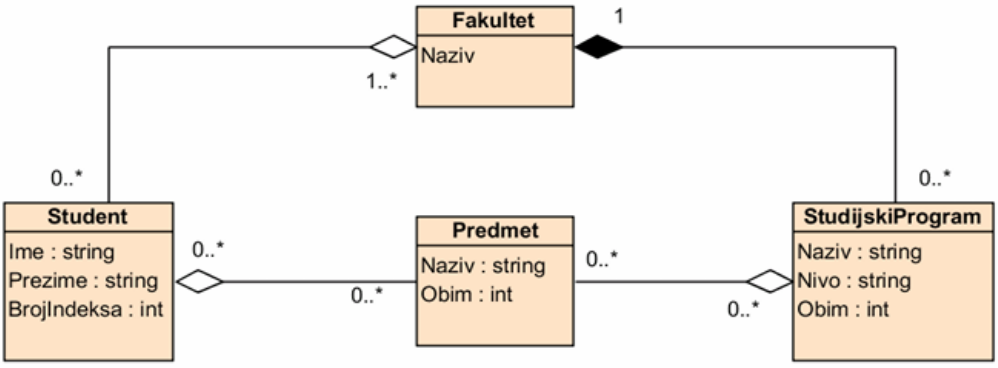
optional



mandatory



Primer dijagrama klasa podataka



Odnosi u modelu podataka

- U modelu klasa podataka većina odnosa ima strukturalnu prirodu
  - zbog toga većina asocijacija može da preraste u agregaciju, ili čak kompoziciju
  - ali ne uvek!!!

Asocijacija ili agregacija?

- Da li fakultet sadrži studente?
  - suštinski možemo da odgovorimo: *ne*
  - studenti nisu fizički deo fakulteta
  - ali na nivou podataka možemo da odgovorimo: *da*
  - podaci o upisanim studentima jesu deo podataka o fakultetu
  - štaviše, tu može da se govori i o kompoziciji
    - student je slab entitet, ne postoji ako nije upisan na fakultet
- Da li student sadrži predmete?
  - suštinski *ne*
  - na nivou podataka *da*, ali *upisane predmete*
- U oba slučaja možemo da koristimo agregaciju
  - fakultet ima skup upisanih studenata
  - student ima skup upisanih predmeta

Asocijacija ili agregacija?

- Da li je greška ako se koristi agregacija *fakultet sadrži studente*?
  - Ne! Ali samo ako se precizira da su to *upisani studenti*.
  - Čak može i kompozicija – ne postoji student koji nije upisan na fakultet
- Da li je greška ako se tu ne koristi agregacija?
  - Ne!
- Pa šta je onda bolje?
  - Zavisi od konteksta...
  - ako pravimo IS fakulteta, bolje je sa agregacijom
  - ako pravimo nešto drugo, gde je informacija o studijama sporedna a ne primarna, onda može da bude bolje da koristimo asocijaciju
- Osnovni kriterijum je razumljivost
  - Konceptualni model služi za što tačnije i jasnije opisivanje domena!
  - Težimo da predstavimo što više informacija (agregacija je dodatna inf.).
  - Šta će neko ko čita dijagram lakše i tačnije razumeti?

Dopune UML-a

- UML sadrži standardizovane koncepte koji omogućavaju uvođenje novih načina označavanja:
  - stereotipovi
  - označene vrednosti
  - proširenja

Stereotipovi u UML-u

- Stereotip predstavlja vrstu *šablona*:
  - apstrakciju opštijeg slučaja nečega
  - neki predefinisani skup osobina
  - neko predefinsano ponašanje
- Koriste se za navođenje dodatnih deklaracija ili napomena, tj. proširenje
  - "šta konkretan element predstavlja u modelu"
- Mogu da se navode i uz klase i uz atribute

Označavanje stereotipova

- Podrazumevano označavanje stereotipova je navođenjem naziva između dvostrukih izlomljenih zagrada:
  - «abstract»
  - «persistent»
  - «entity»
  - «view»
  - «table»
  - «generated»
  - «auto»
  - «key»
- Obično se navodi iznad, ispred ili ispod naziva klase (ili objekta ili atributa)

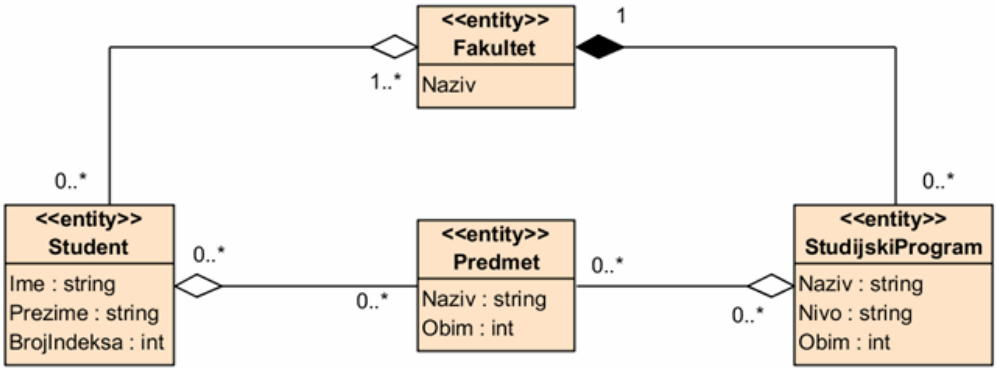
Stereotipovi klasa

- Uobičajeni stereotipovi klasa u dijagramima klasa podataka su:
  - «entity», «persistent», «set» - konceptualni ili logički model entiteta, sadržaj objekata se trajno zapisuje i predstavlja
  - «persistent» može da bude opštija oznaka da se sadržaj objekata trajno zapisuje ali bez prejudiciranja forme
  - «set» - nikada nema ponašanje
  - «type» - radi se o tipu, a ne o skupu entiteta
  - «table» - naglašava se da je u pitanju tabela u logičkom modelu
  - «view» - naglašava se da je u pitanju pogled u logičkom modelu
  - ...

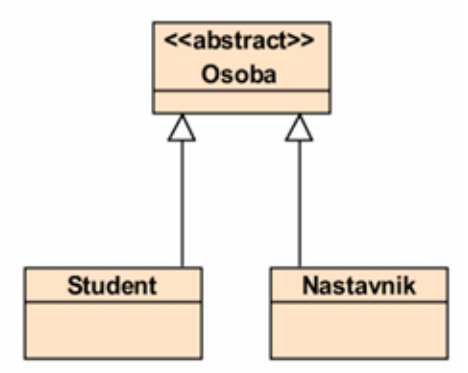
Stereotipovi atributa

- Uobičajeni stereotipovi atributa su:
  - «key» - atribut pripada primarnom ključu
    - može i «PK»
  - «generated» - atribut je izveden, tj. izračunava se automatski na osnovu drugih atributa
  - «auto», «autokey» - automatski generisan atribut, obično surogat ključ
  - «unique» - atribut ima jedinstvenu vrednost
  - «fk» - atribut je referenca (ili deo reference)
    - retko se koristi u konceptualnom modelu, češće u logičkom modelu
  - «null» - atribut može da ima nedefinisanu vrednost
  - «not null» - atribut ne sme da ima nedeifnisanu vrednost
  - ...

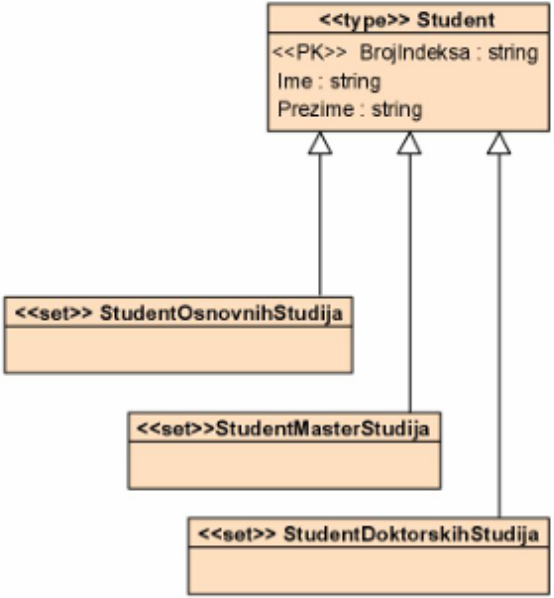
Primer dijagrama sa stereotipovima



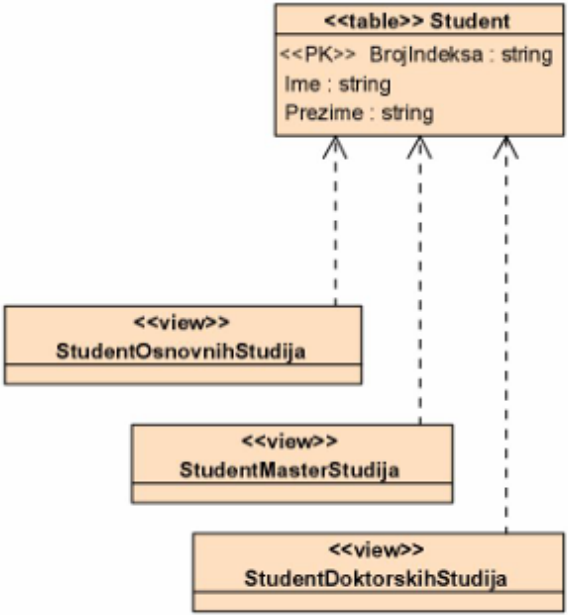
Primer dijagrama sa stereotipovima



Primer dijagrama sa stereotipovima



Primer dijagrama sa stereotipovima



Izmenjeno predstavljanje "klase"

- Neki alati omogućavaju da se stereotipovi koji odgovaraju skupovima podacima predstavljaju nalik na tabele u dijagramu tabela – sa zaobljenim uglovima
- Mogu da se koriste i boje

Razlike između nivoa modeliranja

- Razlike u načinu predstavljanja modela na konceptualnom / logičkom / fizičkom nivou slične su kao kod dijagrama tabela
- Konceptualni dijagram ne mora da sadrži (i obično ne sadrži):
  - dodatne atribute surogat ključeva
  - dodatne atribute stranih ključeva
  - oznake ključeva (osim možda prirodnog ključa)
  - precizne tipove atributa (mada je poželjno makar okvirno)
- Fizički dijagram sadrži i:
  - atribute surogat ključeva (ako su potrebne)
  - atribute vezanih tabela koji čine strane ključeve
  - oznake ključeva i vrsta ključeva
  - tačne tipove atribute

Razlike između nivoa modeliranja

- ...
- Logički dijagram u osnovi može i "ovako"i "onako"ali...
- Ako se dijagram klasa podataka koristi samo kao dijagramska tehnika za predstavljanje logičkog modela koji nije na OOM, onda mora da sadrži sve ono što se očekuje od logičkog modela za odgovarajući model podataka
- Na primer, ako se koristi za predstavljanje logičkog modela kome odgovara RM podataka, onda mora da sadrži sve ono što očekujemo da sadrži taj logički model

Odnosi na različitim nivoima modeliranja

- Videli smo da iste odnose često možemo da modeliramo *ispravno* na različite načine
- Izbor često zavisi od nivoa modela:
  - na konceptualnom nivou, cilj je da očuvamo semantiku odnosa iz domena problema
  - na logičkom nivou, cilj je da ostvarimo stabilnu i jednoznačnu strukturu
  - na fizičkom nivou, cilj je (pored prethodnog) da omogućimo dobre performanse

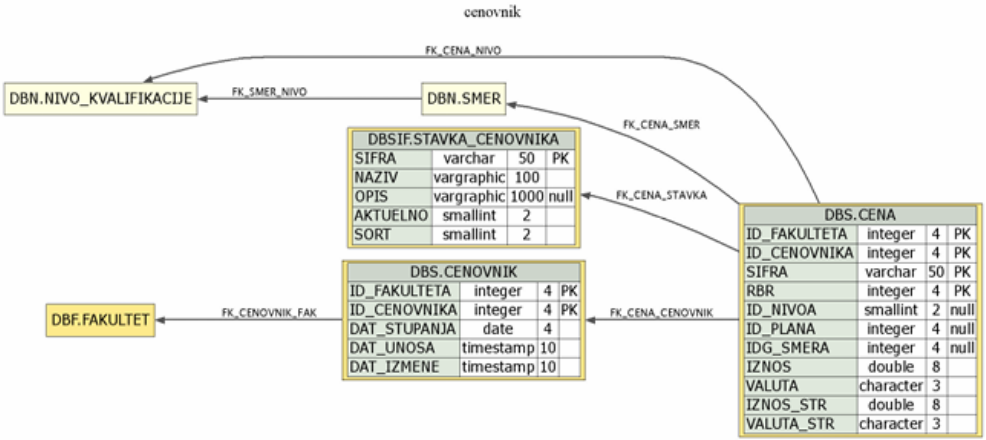
Grupisanje

- Na konceptualnom nivou
  - tabele se grupišu u scheme
  - klase se grupišu u pakete
- Na fizičkom nivou
  - tabele se
    - grupišu u prostore za tabele
    - rapoređuju na čvorove (servere)
  - (funkcionalne) klase se
    - grupišu u komponente
    - raspoređuju na čvorove (servere)

Grupisanje entiteta

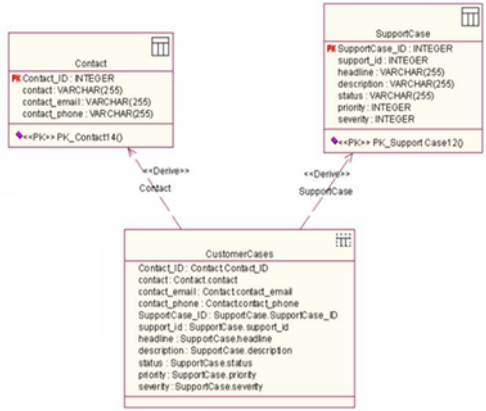
- Slično kao kod ER
  - skrivamo atribute
  - skrivamo posredno povezane elemente
  - koristimo različite boje za označavanje
  - možemo da grupu predstavimo paketom i da uz korišćenje paketa navodimo naziv klase koja se koristi

Grupisanje entiteta

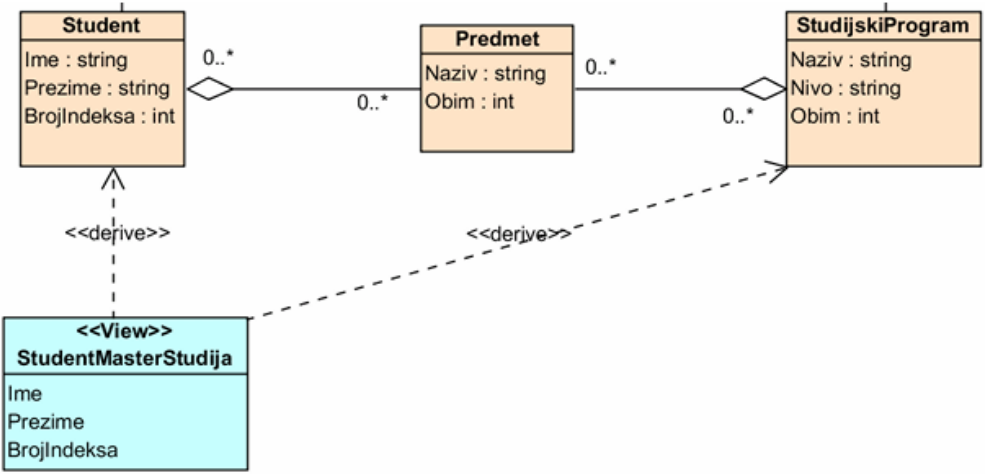


Predstavljanje pogleda - Primeri dodatnih oznaka

- Pogledi se predstavljaju sa oznakom stereotipa u uglu
  - može i simbol
  - ili druga boja
- uvodi se i odnos zavisnosti prema baznim klasama/tabelama

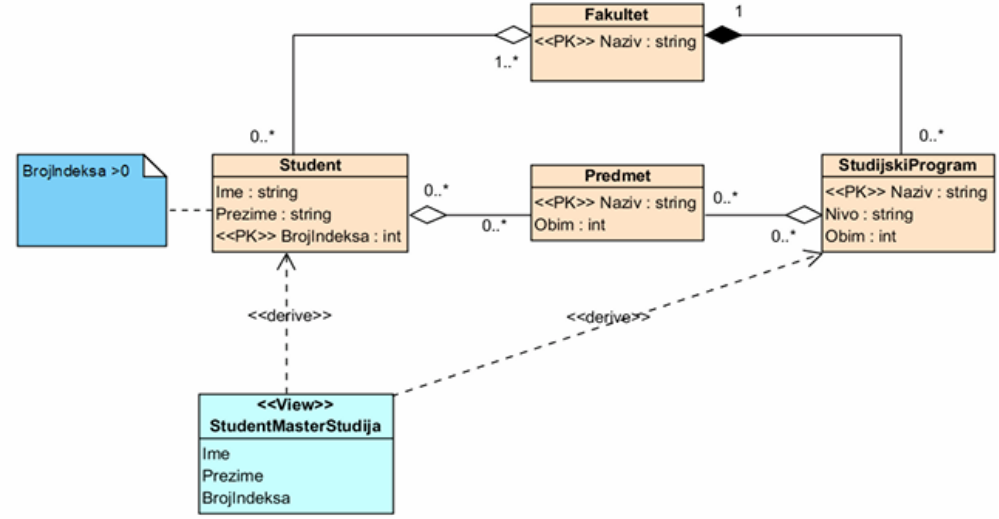


Predstavljanje pogleda - Primeri dodatnih oznaka



Označavanje ključeva

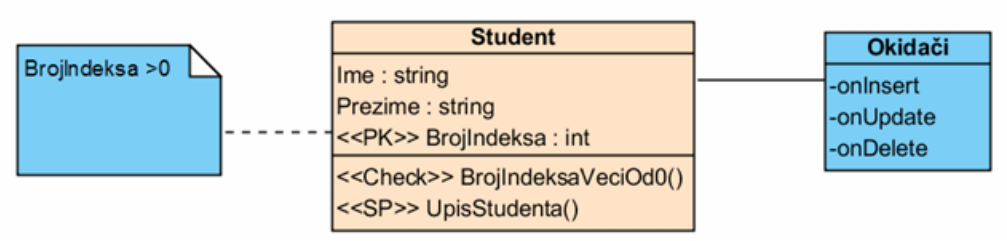
- Simbolima ili tekstom



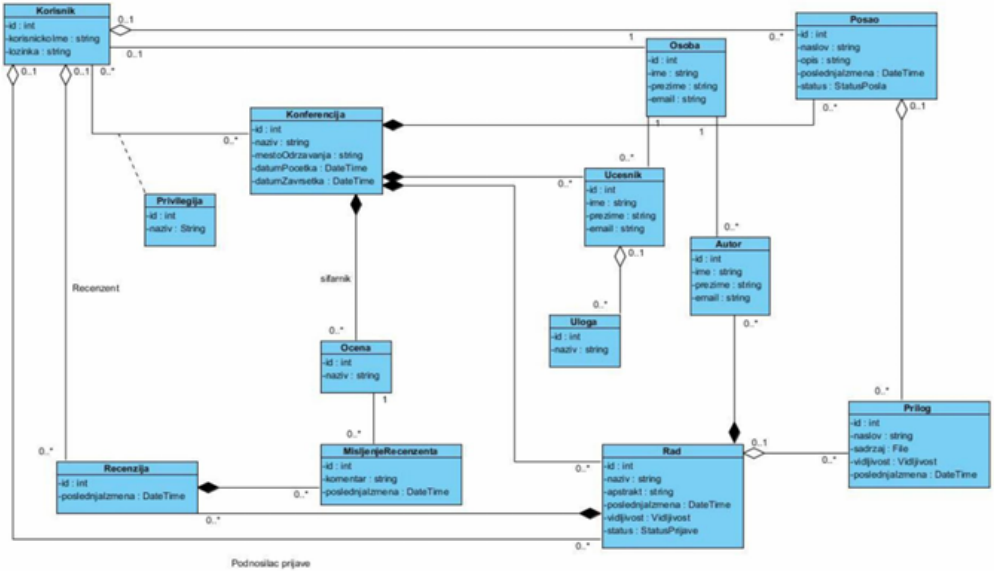
Uslovi integriteta, indeksi,...

- Razne druge stvari mogu da se navode u okviru klasa u odeljku ponašanja
  - ali neki alati ne pružaju takvu slobodu
- Alternativno se navode kao povezane "klase"ili komentari
  - uslovi integriteta
  - indeksi
  - okidači
  - serverske procedure
  - ...

Uslovi integriteta, indeksi,...



Primer – Sistem za podršku organizovanja naučnih skupova



Literatura za temu

- Molina, Database Systems, 2.ed, Pearson Prentice Hall (2009)
- Teorey et al., Database Design, Morgan Kaufmann (2009)
- Muller, Database Design for Smarties – Using UML for Data modeling, Academinc Press (1999)
- Alvaro Monge, Database design with UML and SQL, 4.ed.
  - https://web.csulb.edu/colleges/coe/cecs/dbdesign/dbdesign.php