

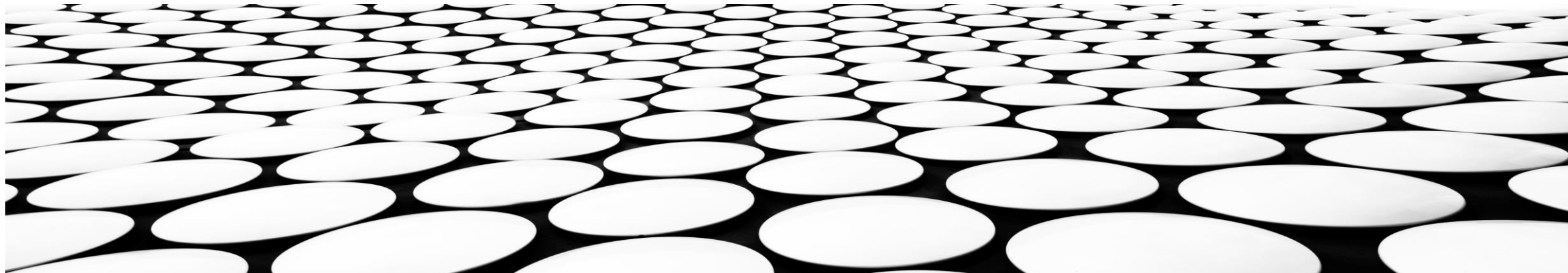


Универзитет у Нишу

ЕЛЕКТРОНСКИ ФАКУЛТЕТ

КАТЕДРА ЗА РАЧУНАРСТВО

Primena UML dijagrama klasa u modelovanju sistema



Sadržaj

- Kratak osvrt na UML
- UML dijagrami klasa
 - Klasa
 - Atributi klase
 - Operacije (metode) klase
 - Relacije
- Preslikavanje dijagrama klasa u C++ kod
 - Primeri ...
- Modelovanje sistema upotrebom UML dijagrama klasa
 - Primeri ...

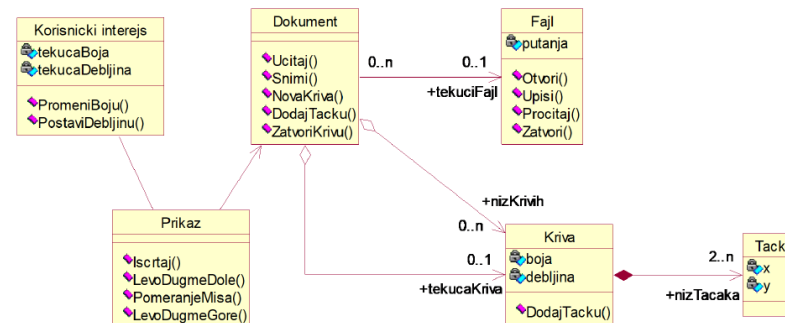
UML (Unified Modeling Language)



- **UML (Unified Modeling Language)**
 - Objedinjeni jezik za modelovanje
- UML predstavlja standardizovani jezik i grafičku notaciju za:
 - vizuelizaciju,
 - specifikaciju,
 - modelovanje i
 - dokumentovanje artefakata softverskog sistema koji se projektuje.
- UML predstavlja zajednički “rečnik” za sporazumevanje između osoba uključenih u projekovanje i razvoj nekog softverskog sistema.

Tipovi UML dijagrama

- **Dijagrami klasa** (*Class Diagram*)

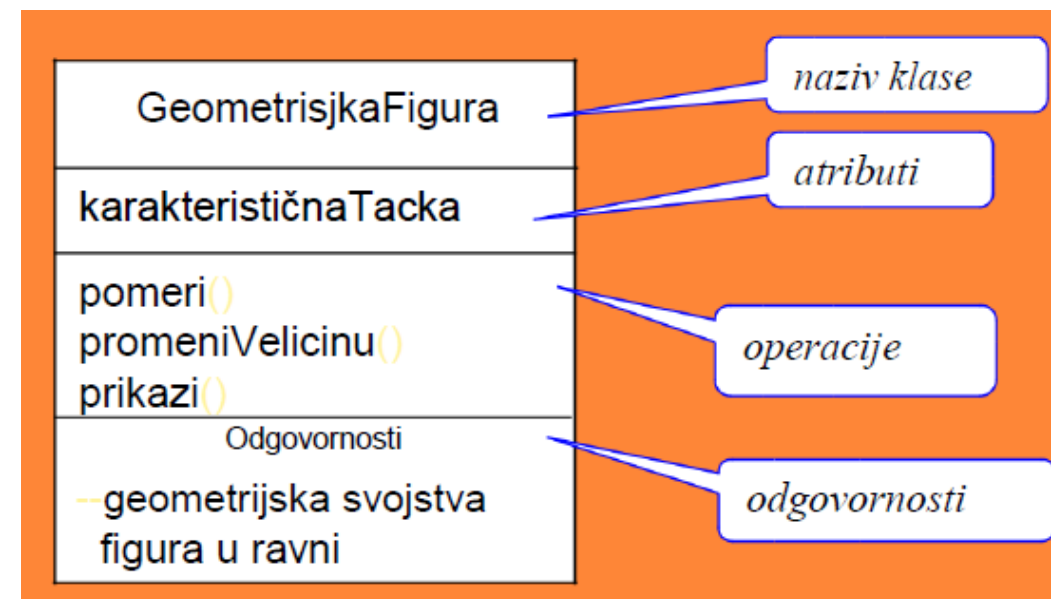
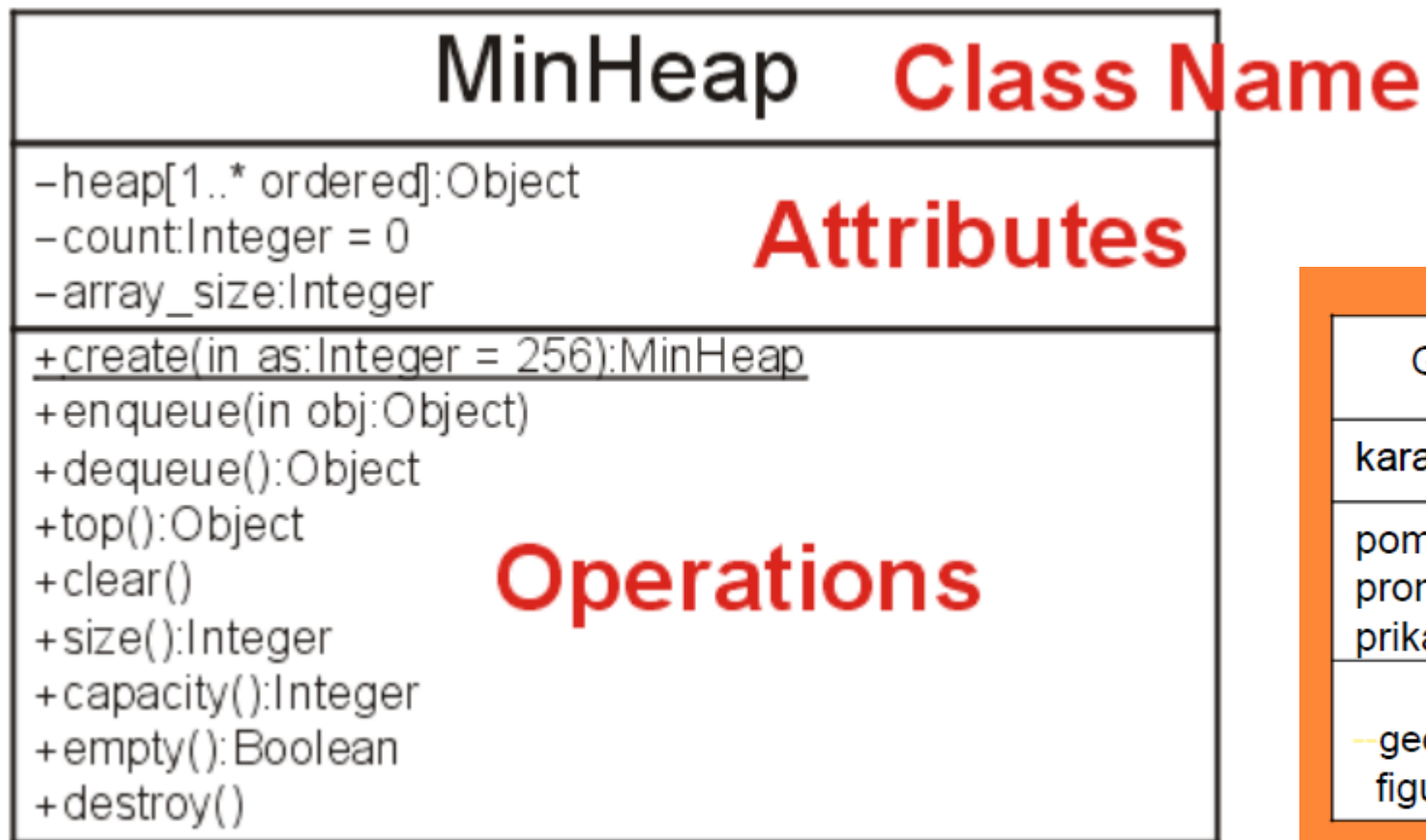


- **Dijagrami slučajeve korišćenja** (*Use-Case Diagram*)
- **Sekvencni dijagrami** (*Sequence Diagram*)
- **Dijagrami saradnje** (*Collaboration Diagram*)
- **Dijagrami stanja** (*Statechart Diagram*)
- **Dijagrami aktivnosti** (*Activity Diagram*)
- **Dijagrami komponenti** (*Component Diagram*)
- **Dijagrami razmeštaja** (*Deployment Diagram*)

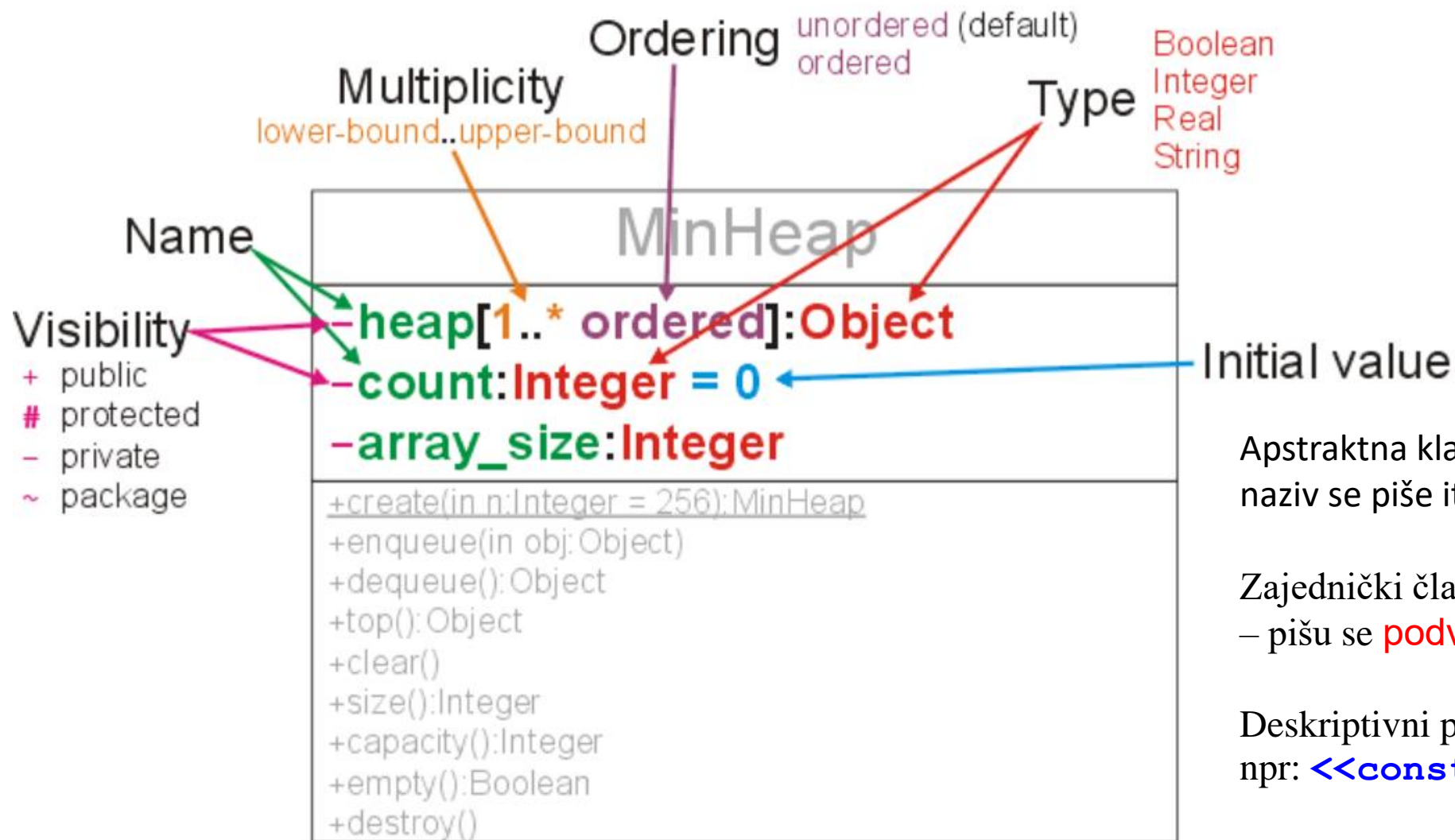
UML dijagrami klasa

- Dijagram klasa prikazuje skup klasa, interfejsa i kolaboracija i njihove relacije.
- Dijagrami klasa se koriste za modelovanje:
 - domena sistema
 - aplikacije
- Elementi dijagrama su:
 - **Stvari:** klase, interfejsi, kolaboracije, paketi, podsistemi, objekti
 - **Relacije:** zavisnosti, generalizacije, asocijacije, realizacije.

UML dijagrami klasa - Klasa



UML dijagrami klasa – Atributi klase

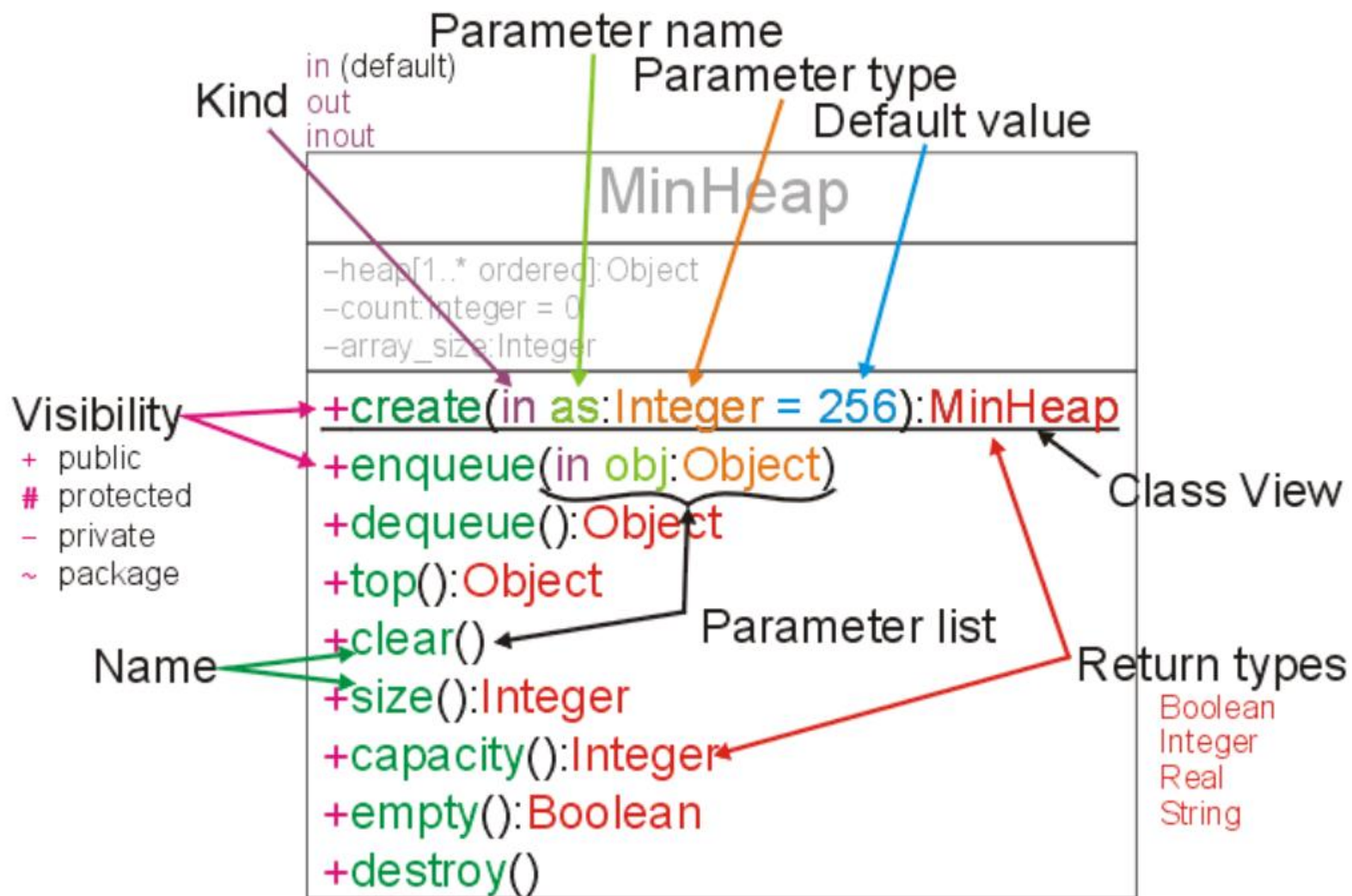


Apstraktna klasa i apstraktna operacija – naziv se piše italicom







Zajednički članovi (atributi i operacije) klase – pišu se **podvučeno**

Deskriptivni prefiks se piše kao stereotip, npr: **<<constructor>>**, **<<query>>**

UML dijagrami klasa – Operacije (metode) klase



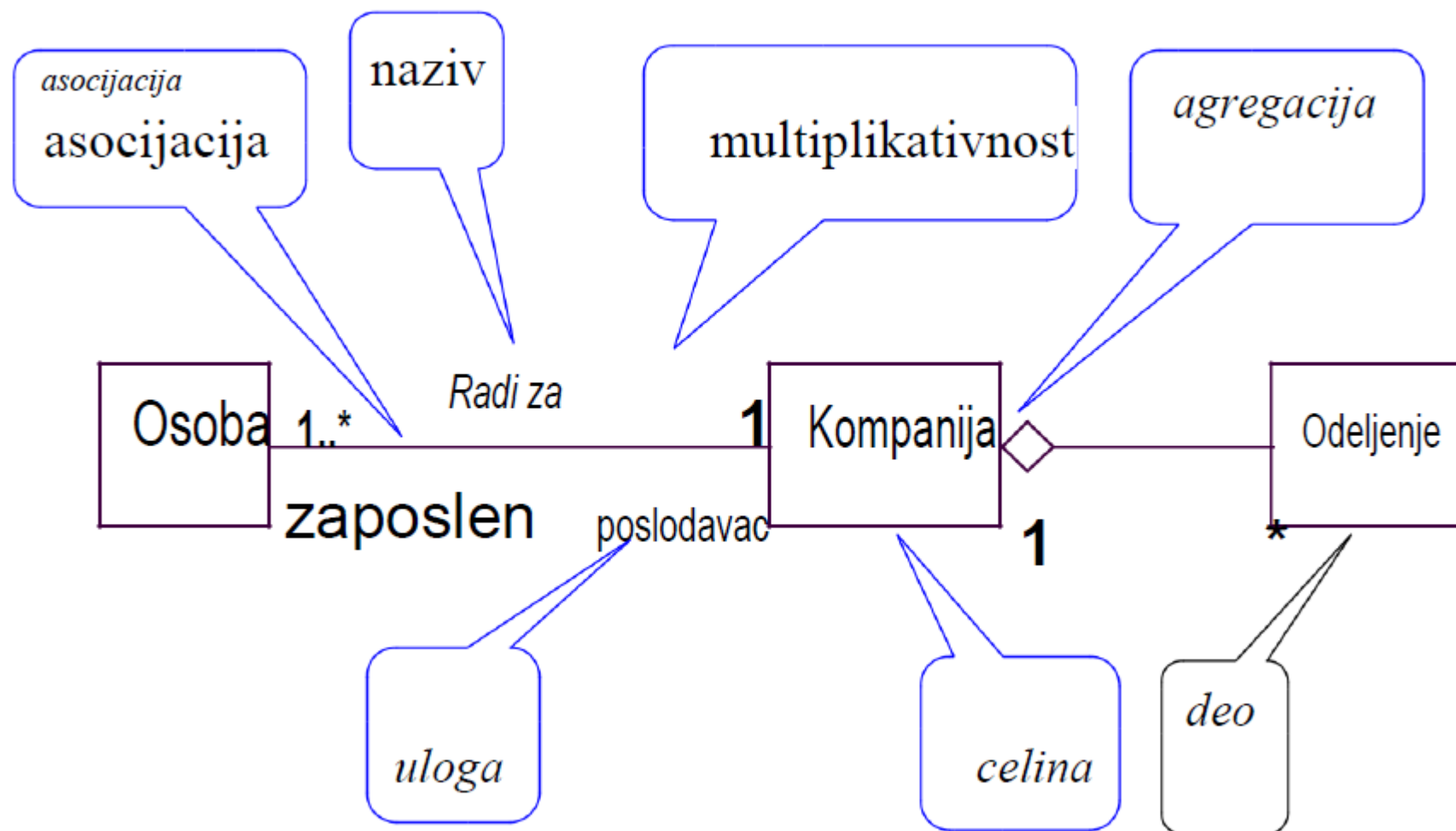
UML dijagrami klasa – Relacije

Relacija	Oznaka relacije na UML dijagramu
Asocijacija	
Generalizacija	
Implementacija/Realizacija	
Zavisnost	
Agregacija	
Kompozicija	

Primeri:

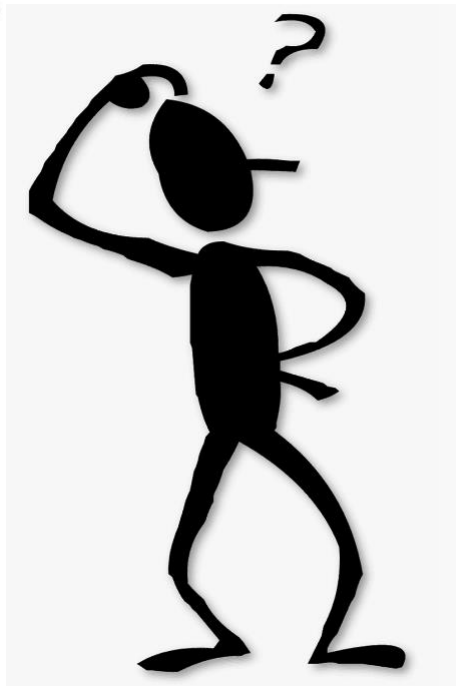
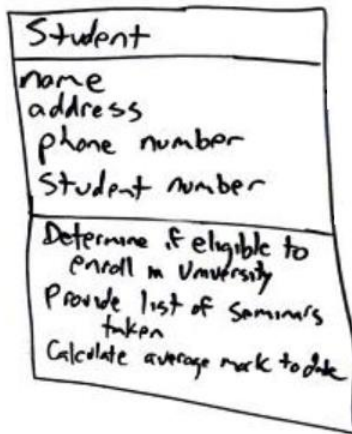
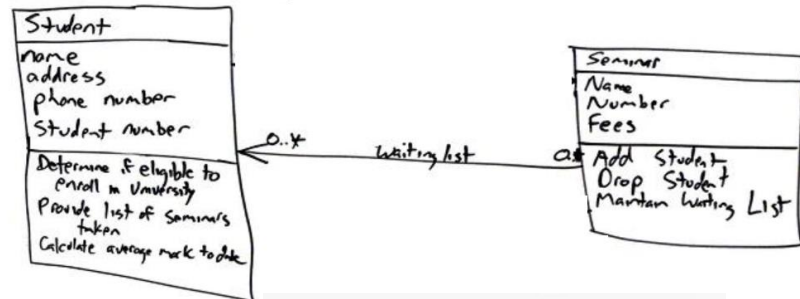


UML dijagrami klasa– Ukrasi asocijacije



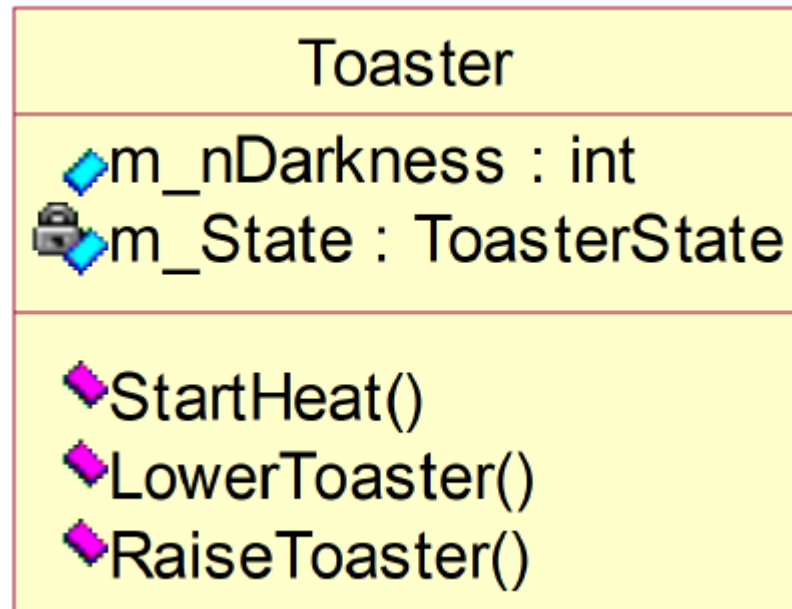
Preslikavanje dijagrama klasa u C++ kod

- Primeri ...

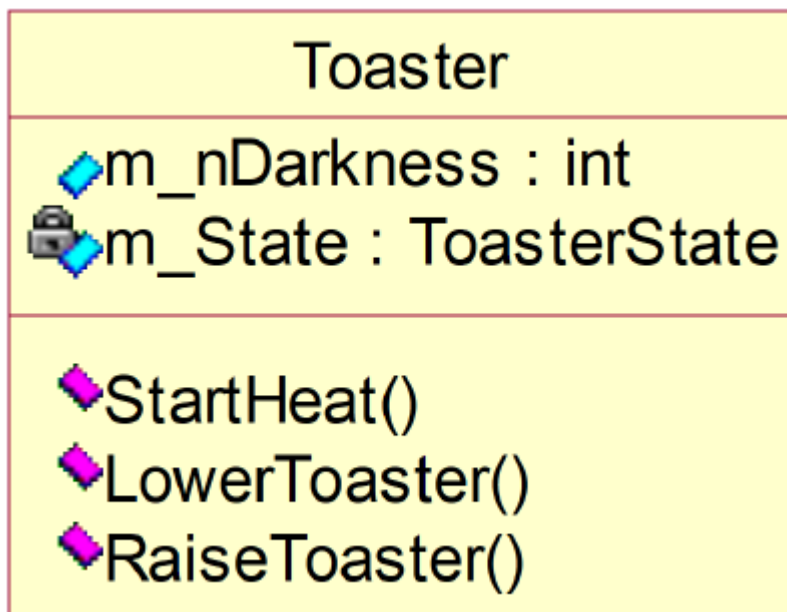


Primer 1.

Napisati C++ kod za klasu prikazanu na slici:



Primer 1. – Rešenje *Toaster.h*



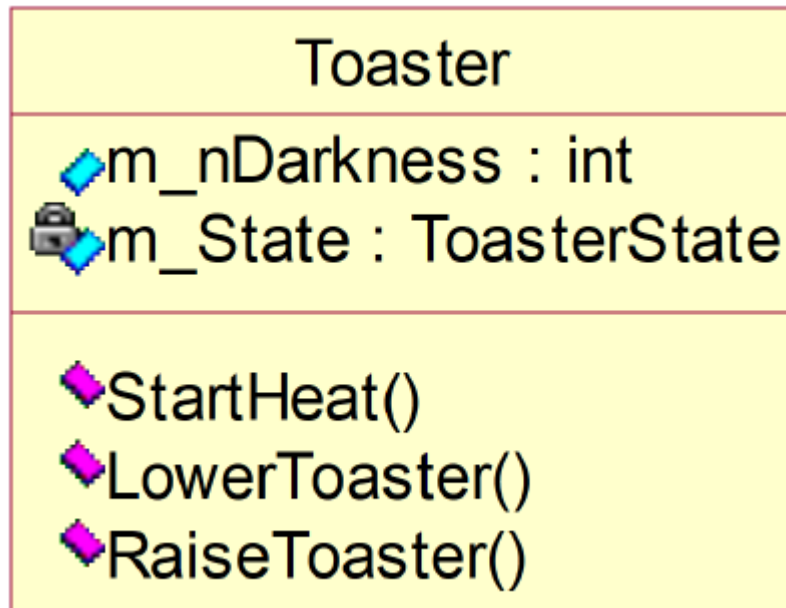
```
class Toaster
{
public:
    int m_nDarkness;

    void StartHeat();
    void LowerToast();
    void RaiseToast();

private:
    ToasterState m_State;
};
```

Primer 1. – Rešenje

Toaster.cpp



```
#include "Toaster.h"

void Toaster::StartHeat()
{
    //...
}

void Toaster::LowerToast()
{
    //...
}

void Toaster::RaiseToast()
{
    //...
}
```

Primer 2.

Napisati C++ kod za dijagram klasa prikazan na slici:



Primer 2. – Rešenje *LeftRight.h*

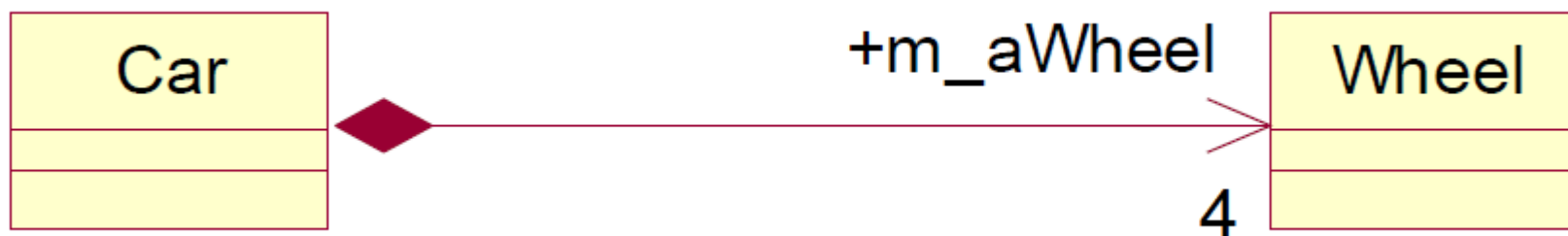


```
class Right;
class Left
{
public:
    Right* right;
};
```

```
class Right
{
public:
    Left* left;
};
```

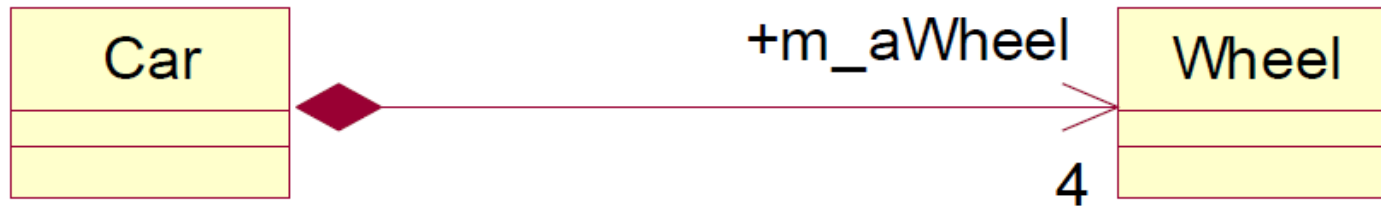
Primer 3.

Napisati C++ kod za dijagram klasa prikazan na slici:



Primer 3. – Rešenje

Car.h

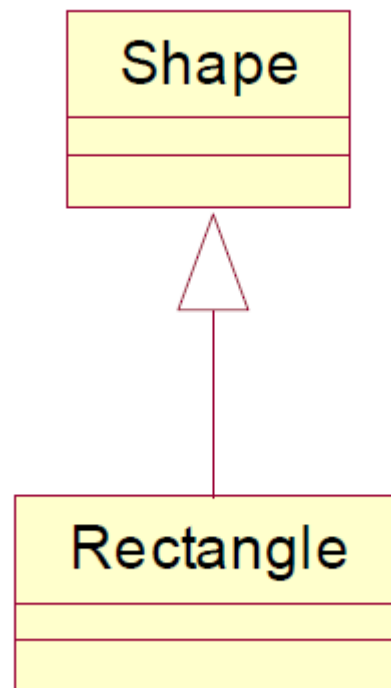


```
#include "Wheel.h"

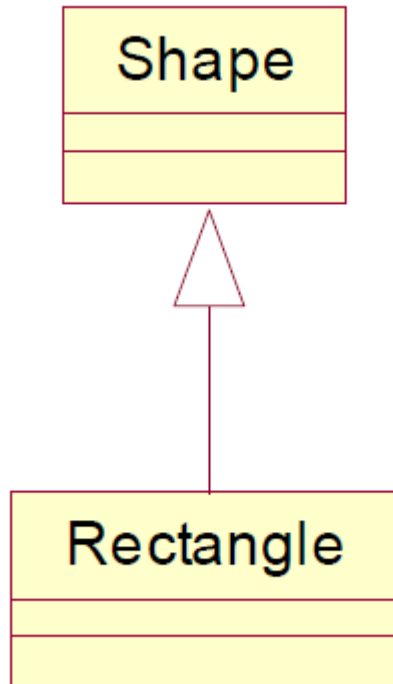
class Car
{
public:
    Wheel m_aWheel[4];
};
```

Primer 4.

Napisati C++ kod za dijagram klasa prikazan na slici:



Primer 4. – Rešenje *Rectangle.h*



```
#include "Shape.h"

class Rectangle : public Shape
{
    //...
};
```

Modelovanje sistema upotrebom UML dijagrama klasa

- Primeri ...



Primer 5.

Dijagramom klasa predstaviti model fakulteta. Svaki student upisuje studije na jednom i samo jednom odseku, a odsek pridapa jednom i samo jednom fakultetu.






Primer 5. - Rešenje

Dijagramom klasa predstaviti model **fakulteta**. Svaki **student** upisuje studije na jednom i samo jednom **odseku**, a odsek pripada jednom i samo jednom fakultetu.

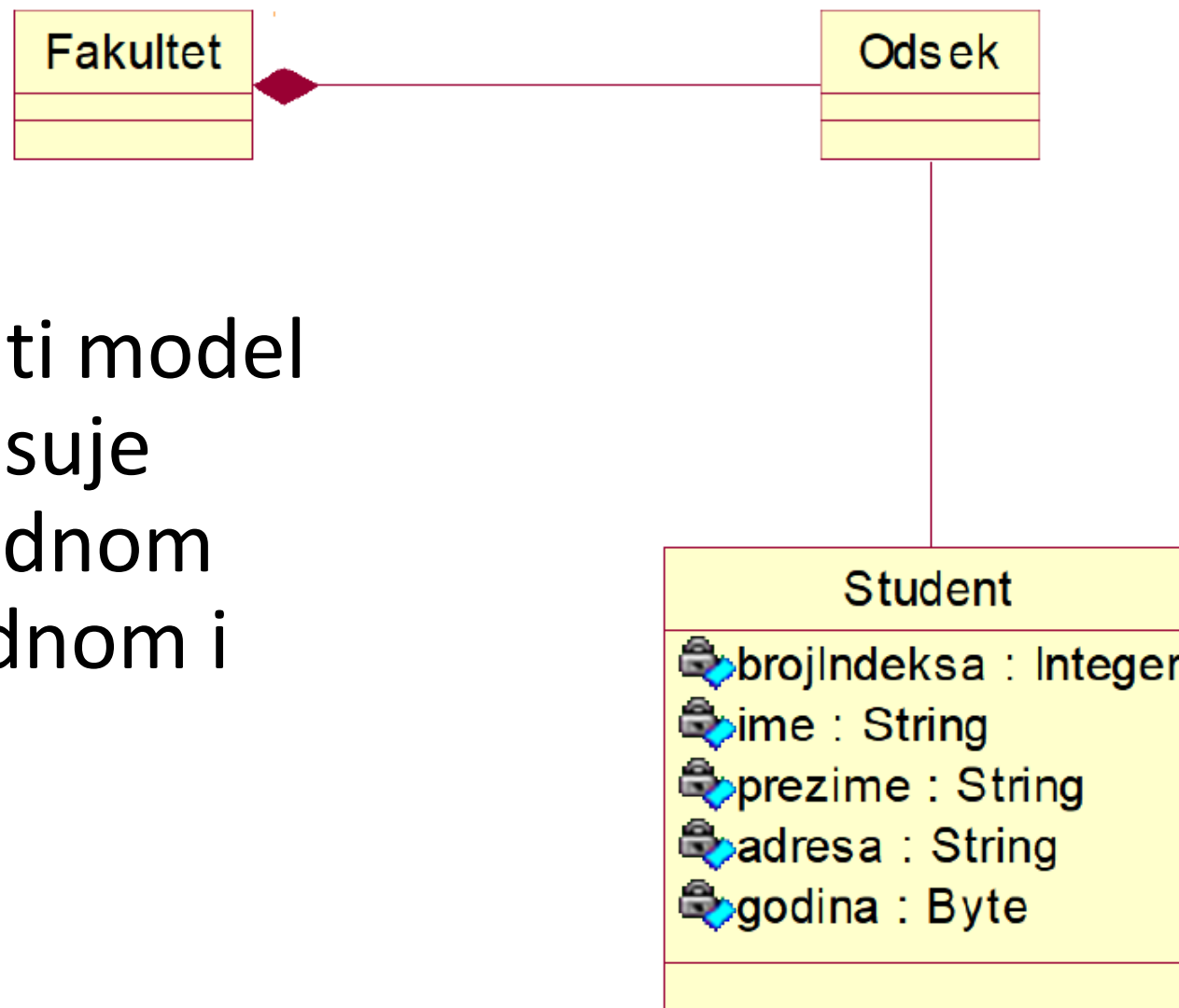
Primer 5. - Rešenje

Fakultet

Odsek

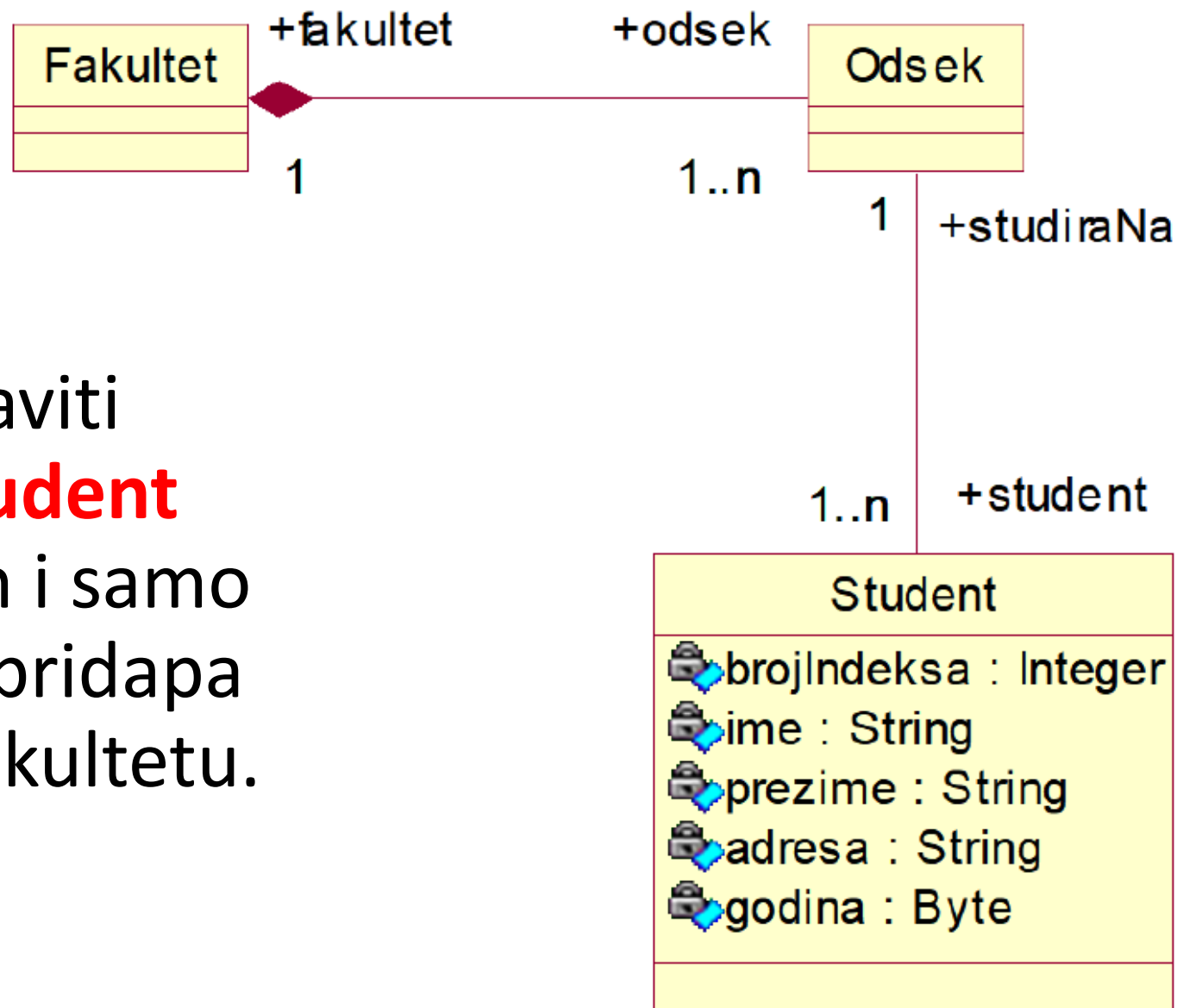
Student
 brojIndeksa : Integer
 ime : String
 prezime : String
 adresa : String
 godina : Byte

Primer 5. - Rešenje



Dijagramom klasa predstaviti model **fakulteta**. Svaki **student** upisuje studije na jednom i samo jednom **odseku**, a odsek pridapa jednom i samo jednom fakultetu.

Primer 5. - Rešenje



Dijagramom klasa predstaviti model **fakulteta**. Svaki **student** upisuje studije na jednom i samo jednom **odseku**, a odsek pridapa jednom i samo jednom fakultetu.

Smernice 1/3

- Identifikovati u tekstu pojedinačne iskaze (rečenice).
 - Svaki iskaz nosi određene informacije o sistemu koji se projektuje.
- Identifikovati ključne entitete u iskazima (npr. **Fakultet, Student, Ispit, Predavač, Profesor, Asistent**).
 - Svaki entitet će predstavljati jednu klasu u UML klasnom dijagramu.
- Za svaki entitet identifikovati ključne attribute koji se spominju u iskazima (npr. **Ime, Prezime**, itd ...).
- Za svaki entitet identifikovati ključne metode koje se spominju u iskazima.
 - (npr. za studenta može da bude: **prijaviIspit()**).
- Pristupiti identifikovanju veza između entiteta.

Smernice 2/3

- Poželjno je identifikovati generalizaciju izmedju pojedinih entiteta (npr. **Prodavač**, **Menadžer** i **Sekretarica** su izvedeni iz **Zaposlen**).
 - Razmotriti i mogućnost dodavanja novog entiteta (koji nije eksplicitno naveden u iskazima) kako bi se predstavila generalizacija.
- Identifikovati postojanje odnosa **deo/celina** između pojedinih entiteta (npr. **Fakultet** je celina a **Katedra** je deo).
 - Razmisliti kako životni vek celine utiče na životni vek dela.
 - Ukoliko ne postoji zavisnot životnog veka celine i dela onda se odnos ovih entiteta predstavlja agregacijom (prazan romb).
 - Ukoliko životni vek celine utiče na životni vek dela onda se odnos ovih entiteta predstavlja kompozicijom (popunjeni romb).

Smernice 3/3

- Pristupiti skiciranju UML dijagrama.
- Voditi računa da postoji što manje presecanja linija (realacija) kako bi UML klasni dijagram bio pregledan.
- Voditi računa da UML klasni dijagram rasporedom entita što jasnije predstavlja hijerarhiju/strukturu sistema.
- Za svaku definisanu vezu između entiteta naznačiti **multiplikativnost**.

Primer 6. Dijagramom klasa prikazati sledeću organizaciju

- Fakultet se sastoji od katedri.
- Katedra pripada samo jednom fakultetu.
- Ne postoji fakultet bez katedri (bar jedna).
- Više kursa se može držati na jednoj katedri.
- Jedan isti kurs se može držati na više katedri (isti predmet).
- Na katedri mora se držati bar jedan ispit (kurs).
- Kurs se drži ukoliko postoji bar jedan predavač koji može da drži nastavu iz tog predmeta.
- Jedan predavač može držati više kurseva stim što da bi bio predavač mora da predaje na jednom predmetu.
- Svaka katedra okuplja nastavnike.
- Jedan nastavnik pripada samo jednoj katedri
- Predavač može biti šef jedne katedre.
- Samo jedan predavač može biti u isto vreme šef katedre.
- Student može da pohađa bar jedan fakultet.
- Fakultet mora da ima bar jednog upisanog studenta.
- Studenti mogu da pohađaju kurseve na fakultetu.

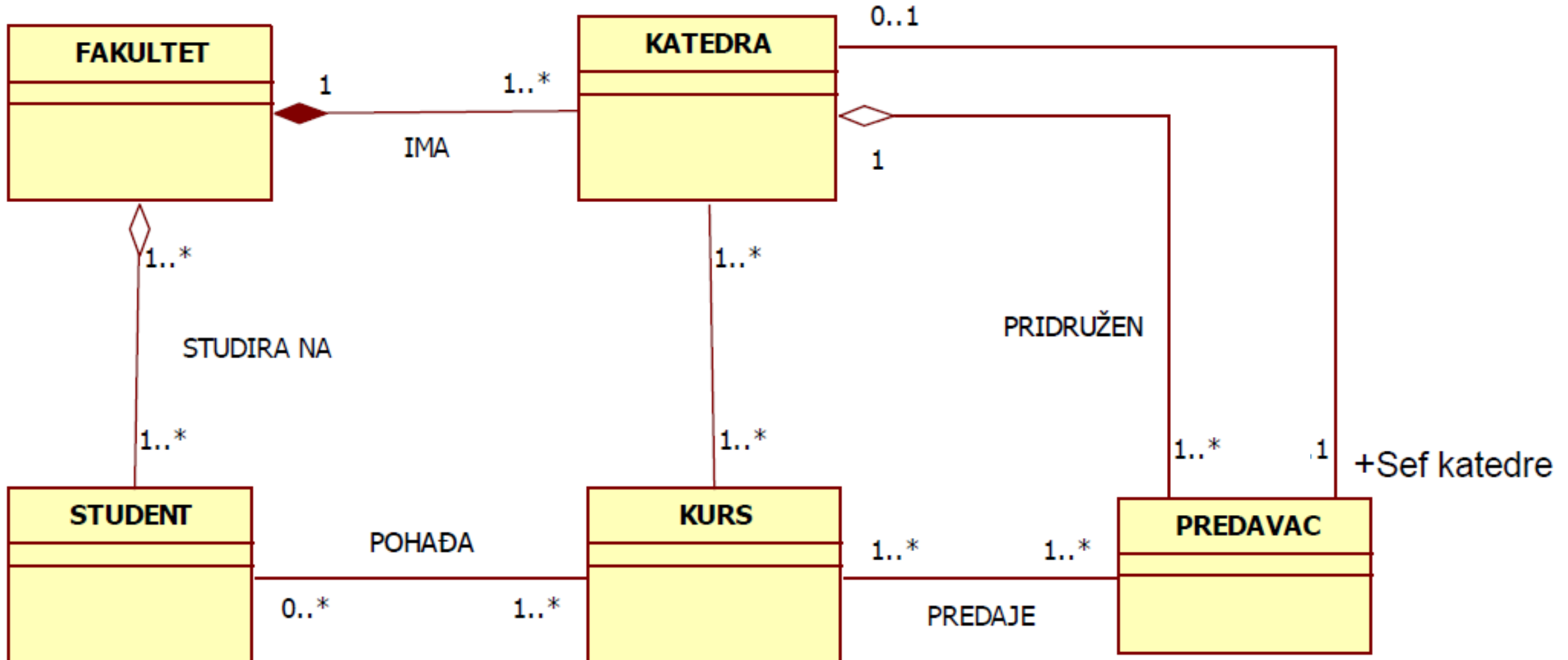
Primer 6. – Rešenje (identifikovanje entiteta)

- **Fakultet** se sastoji od **katedri**.
- **Katedra** pripada samo jednom **fakultetu**.
- Ne postoji **fakultet** bez **katedri** (bar jedna)
- Više **kursa** se može držati na jednoj **katedri**.
- Jedan isti **kurs** se može držati na više **katedri** (isti predmet)
- Na **katedri** mora se držati bar jedan ispit (**kurs**)
- **Kurs** se drži ukoliko postoji bar jedan **predavač** koji može da drži nastavu iz tog predmeta.
- Jedan **predavač** može držati više **kurseva** stim što da bi bio **predavač** mora da predaje na jednom predmetu.
- Svaka **katedra** okuplja **nastavnike**.
- Jedan **nastavnik** pripada samo jednoj **katedri**
- **Predavač** može biti šef jedne katedre.
- Samo jedan **predavač** može biti u isto vreme šef katedre.
- **Student** može da pohađa bar jedan **fakultet**.
- Fakultet mora da ima bar jednog upisanog studenta.
- **Studenti** mogu da pohađaju kurseve na **fakultetu**.



- **Fakultet**
- **Katedra**
- **Kurs**
- **Predavač**
- **Student**

Primer 6. – Rešenje (UML klasni dijagram)



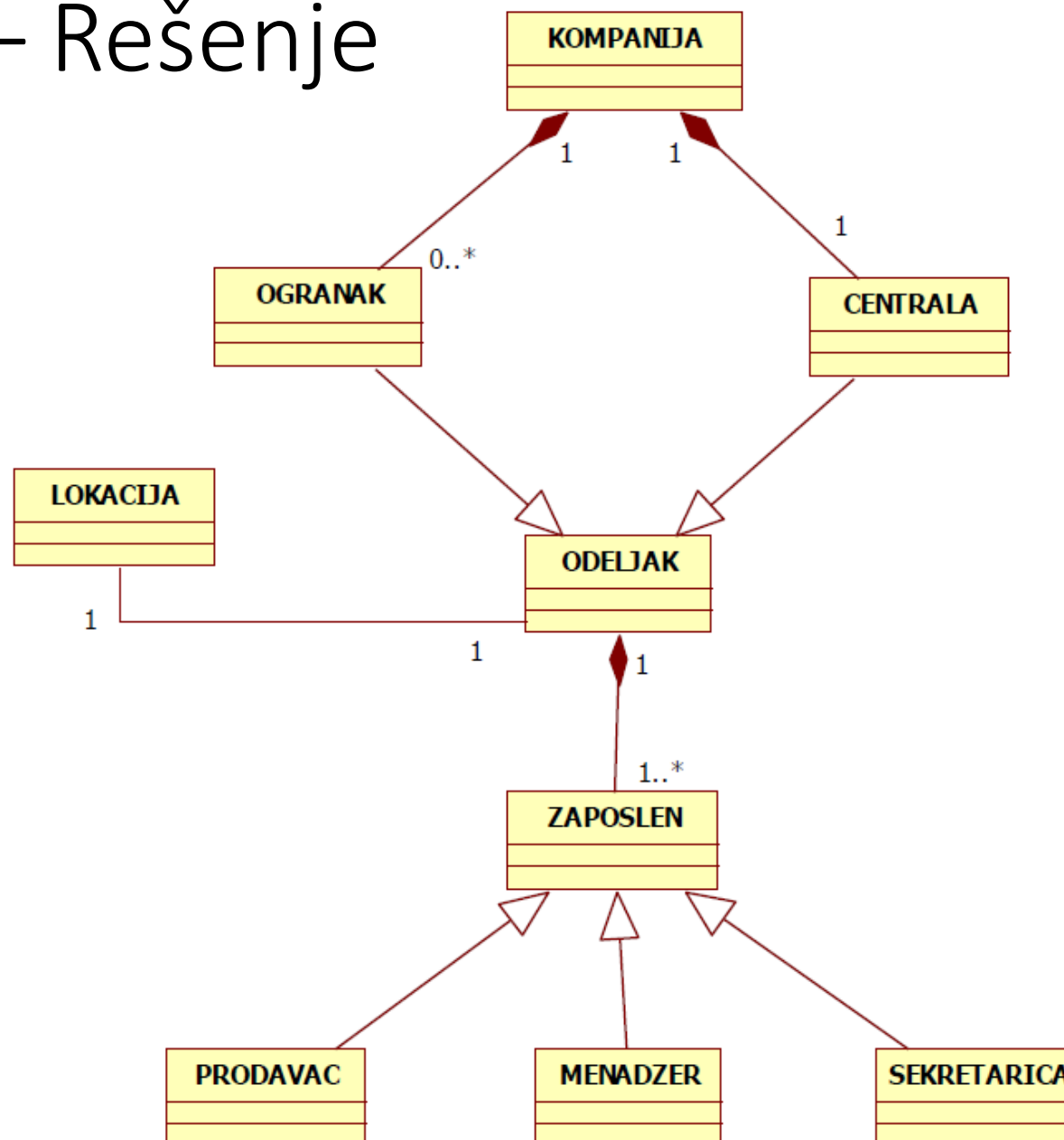
Primer 7. Dijagramom klasa prikazati sledeću organizaciju

- Kompanija se sastoji od centrale i nekoliko ogranaka koji se nalaze na različitim lokacijama.
- Na jednoj lokaciji nalazi se jedan ogranak ili centrala.
- Svaki odeljak ima različit broj zaposlenih (menadžeri, sekretarice, prodavci).
- Svaki zaposleni se karakteriše svojim id-jem, imenom, pozicijom i platom.
- Pored ovih zajedničkih karakteristika, menadžeri se još karakterišu datumom početka rada i bonusom, prodavci područjem koje pokrivaju, a sekretarice brzinom kucanja.
- Svaki zaposleni radnik vezan je ili za ogranak ili centralu.
- Kada se ogranak zatvori, zaposleni radnici se otpuštaju.

Primer 7. – Rešenje

- **Kompanija** se sastoji od **centrale** i nekoliko **ogranaka** koji se nalaze na različitim **lokacijama**.
- Na jednoj **lokaciji** nalazi se jedan **ogranak** ili **centrala**.
- Svaki **odeljak** ima različit broj **zaposlenih** (**menadžeri**, **sekretarice**, **prodavci**).
- Svaki **zaposleni** se karakteriše svojim **id-jem**, **imenom**, **pozicijom** i **platom**.
- Pored ovih zajedničkih karakteristika, **menadžeri** se još karakterišu **datumom početka rada** i **bonusom**, **prodavci** **područjem koje pokrivaju**, a **sekretarice** **brzinom kucanja**.
- Svaki **zaposleni** radnik vezan je ili za **ogranak** ili **centralu**.
- Kada se **ogranak** zatvori, **zaposleni** radnici se otpuštaju.

Primer 7. – Rešenje



Primer 8. 1/2

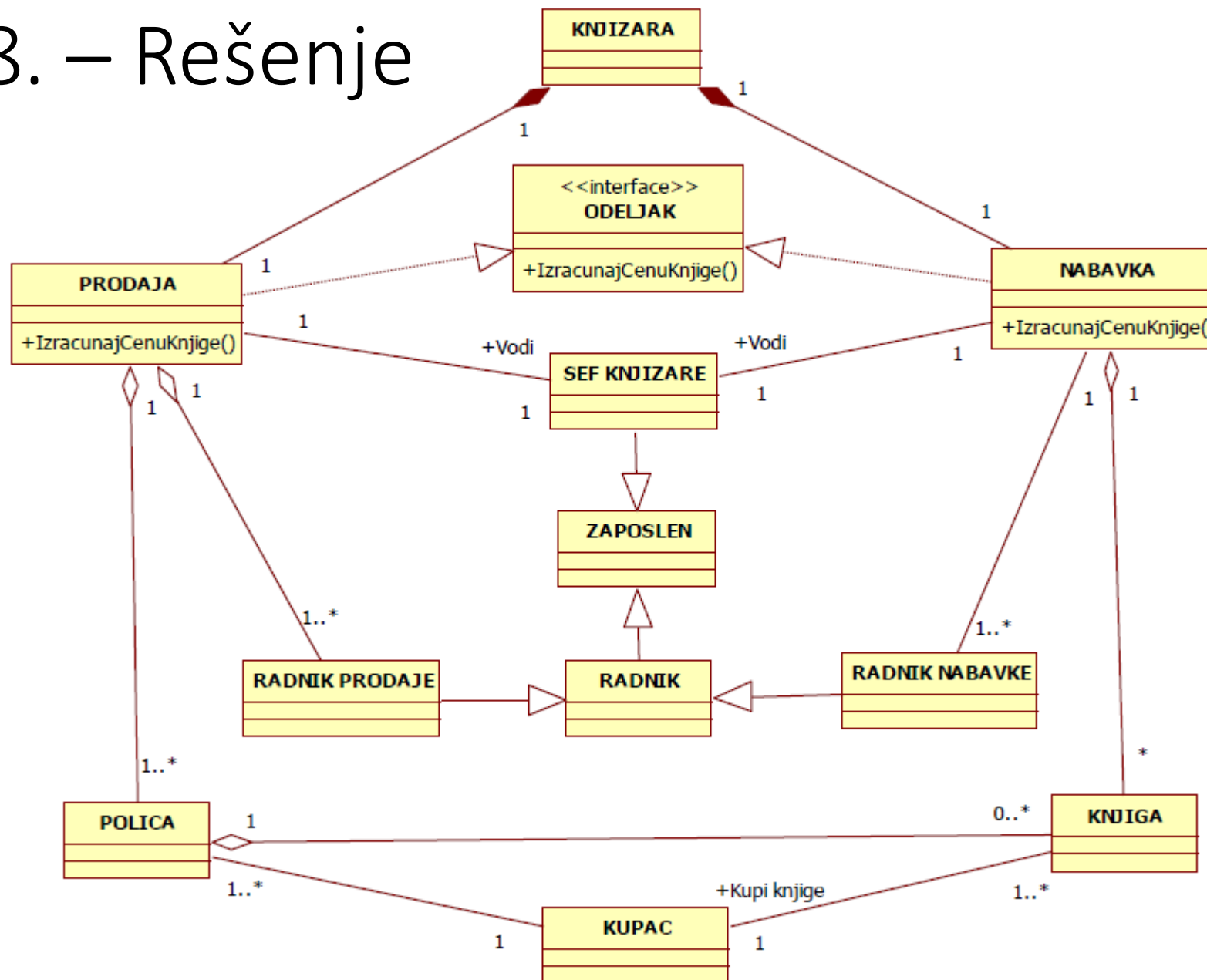
Dijagramom klasa prikazati sledeću organizaciju:

- **Knjižara** se sastoji od dva **odeljka**: **Prodaja** i **Nabavka**.
- **Knjižara** ima dva tipa **zaposlenih**: **Šef knjižare** i **Radnik**, s tim da uvek postoji samo jedan šef i samo jedan radnik.
- **Šef** vodi prodaju i nabavku.
- Jedinice **prodaje** i **nabavke** imaju vlastite **zaposlene**.
- **Radnici** koji rade u **prodaji** ne rade u **nabavci** i obrnuto.
- Ukoliko se **nabavka** ili **prodaja** zatvori, **radnici** neće biti otpušteni iz **knjižare**.
- Svaki **zaposleni** opisan je atributima: **JBMG**, **ime**, **prezime**.
- Glavni artikal sa kojim radi **nabavka** i **prodaja** je **knjiga**.
- Svaka **knjiga** ima **ISBN**, **naziv**, **izdavač** i **datum izdavanja**.

Primer 8. 2/2

- U **prodaji** nalazi se jedna ili više **polica** na kojima se nalazi više **knjiga**.
- U jednom trenutku jedna **knjiga** se može naći samo na jednoj **polici**.
- Ukoliko se neka **polica** ukloni, **knjige** se premeštaju na drugu **policu**.
- Ukoliko se **jedinica prodaje** ugasi, sa njom ne nestaju **police**.
- **Jedinica nabavke** odjednom dobija grupe od po 100 **knjiga**.
- U **knjižaru** dolaze **kupci**.
- **Kupci** mogu uzeti **knjige** sa **polica**, vratiti **knjige** na **police**, konsultovati se sa jednim radnikom koji radi u prodaji, stati u red, izaći iz reda i platiti **knjige**.
- **Radnik** može savetovati **kupca** i naplatiti **knjige** ako radi u prodaji.
- **Radnik** u **prodaji** ili **nabavci** u svakom trenutku se može konsultovati sa **šefom**.
- Oba odeljka imaju istu funkciju **IzracunajCenuKnjige()** koju implementiraju na različite načine.

Primer 8. – Rešenje



Domaći zadatak

- Za sve dijagrame klasa (urađene primere) napisati kod u programskom jeziku C++.
- Da li se neki od primera može drugačije uraditi (UML dijagram)?
- Za izabrane primere napisati kod u nekom drugom programskom jeziku npr. C#, JAVA, Python, ...
- Za primere identifikovati potencijalna ograničenja.

Hvala na pažnji 😊 !