Objektno orjentisano programiranje:

Klase, objekti i njihovi elementi

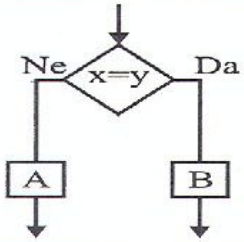
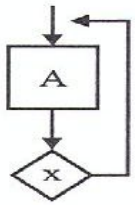
prof dr Gordana Đorđević

Metode programiranja

- Modularno programiranje
- Strukturno programiranje
- Objektno-orjentisano programiranje

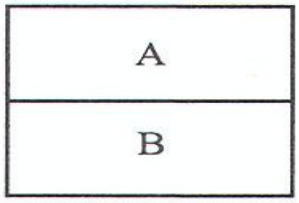
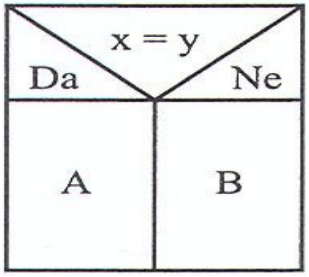
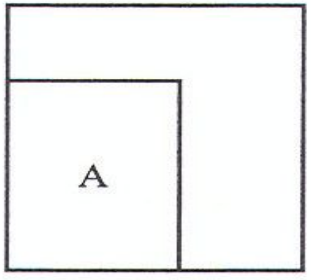
Modularno programiranje

- Metod modularnog programiranja podrazumeva dekompoziciju jednog složenog (kompleksnog) programa na module čime se obezbeđuje jednostavnost i efikasnost programa
- Dva osnovna koncepta o kojima treba voditi računa pri ovoj dekompoziciji su:
  - čvrstoća (kohezija) modula – mera funkcionalne povezanosti naredbi unutar jednog modula
    - Logička čvrstoća
    - Komunikacijska čvrstoća
    - Funkcionalna čvrstoća
  - povezanost modula
    - Bezuslovno povezivanje
    - Uslovno povezivanje
    - Iterativno povezivanje

Elemenat	Sekvencija	Selekcija	Iteracija
Značenje - pseudo kod	Redosled	IF x=y THEN A ELSE B	DO WHILE x URADI A (EOF)
Klasični blok dijagram			

Strukturno programiranje

- Strukturno programiranje zasniva se na standardizaciji programske logike, pa se kompleksne logičke strukture razbijaju na elementarne logičke celine (module) i to po principima koji baziraju na logici razmišljanja čoveka. Koncepti na bazi kojih se vrši dekompozicija na module i obeležja koja moduli moraju posedovati su identični kao i kod metode modularnog programiranja.
- Kako metoda strukturnog programiranja obezbeđuje preglednost, jasnoću i lakše održavanje programa, izbegavaju se instrukcije grananja koje će izvršavanje programa sa jednog mesta prebaciti na neko sasvim drugo (npr. naredba GO TO). Ovakve instrukciji inače bitno usporavaju izvršavanje programa i umanjuju njegovu preglednost. Zato ova metoda programiranja bazira na korišćenju nekih drugih programskih instrukcija (IF-THAN-ELSE, DO-WHILE) koje obezbeđuju standardizaciju procedura grananja. Za razliku od metode modularnog programiranja koje koriste klasične blok dijagrame, metoda strukturnog programiranja koriste Chapinove dijagrame (slika 4.9.).

Elemenat	Sekvencija	Selekcija	Iteracija
Značenje - pseudo kod	Redosled	IF x=y THEN A ELSE B	DO WHILE x URADI A (EOF)
Chapinov dijagram			

Objektno orjentisano programiranje

- Strukturno programiranje baziralo je na skupu tehnika koje su se koristile za izradu programa sa jasnom i lako razumljivom strukturom, uz korišćenje podataka sa jasno definisanim strukturama (nizovi, zapisi, skupovi, datoteke). Kako su vremenom programski paketi narasli toliko da tehnike strukturnog programiranja nisu više davale zadovoljavajuće rezultate, razvijena je nova metoda programiranja – objektno orjentisano programiranje.

- Dok je u centru strukturnog programiranja - struktura programa, u centru OOP je objekat.

- Objektno orjentisano programiranje omogućava verno preslikavanje realnog sistema u računarski sistem. Ova metoda programiranja omogućiće predstavljanje realnog sistem kao skup međusobno povezanih objekata. Na taj način se ceo sistem svodi na manje celine koje međusobno komuniciraju, a korisnik može lakše da sagleda i razume takav sistem i da njime lakše upravlja.

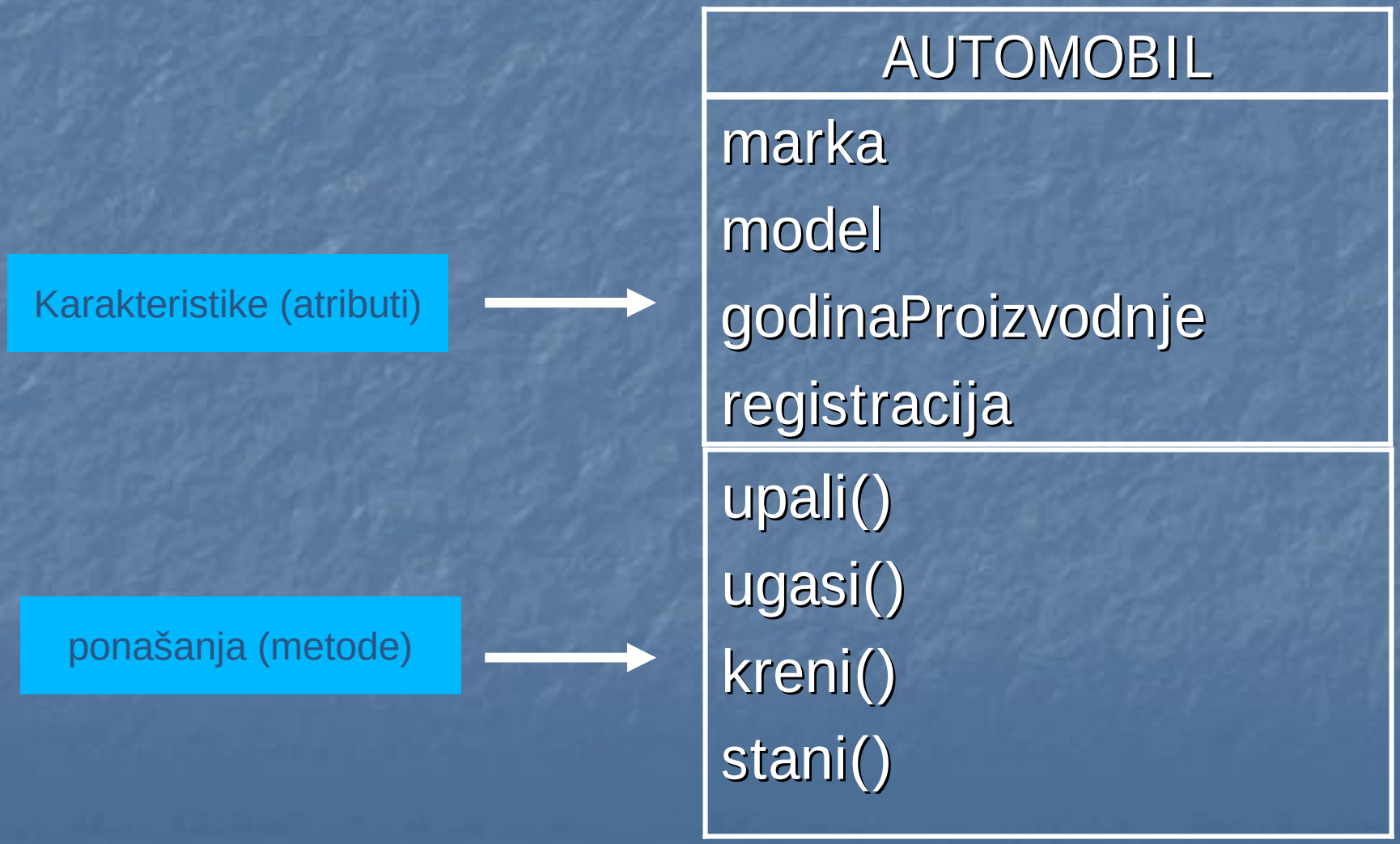
Osnovni principi OOP: klase, objekti i njihovi elementi

- Klasa je opšti predstavnik nekog skupa objekata (predmeta ili pojava) koji imaju istu strukturu i ponašanje (npr. klasa Osoba, klasa Automobil, ...). Klasa sadrži opis odnosno definiciju ovih realnih predmeta i pojava, predstavljajući tako njihovu uprošćenu sliku.

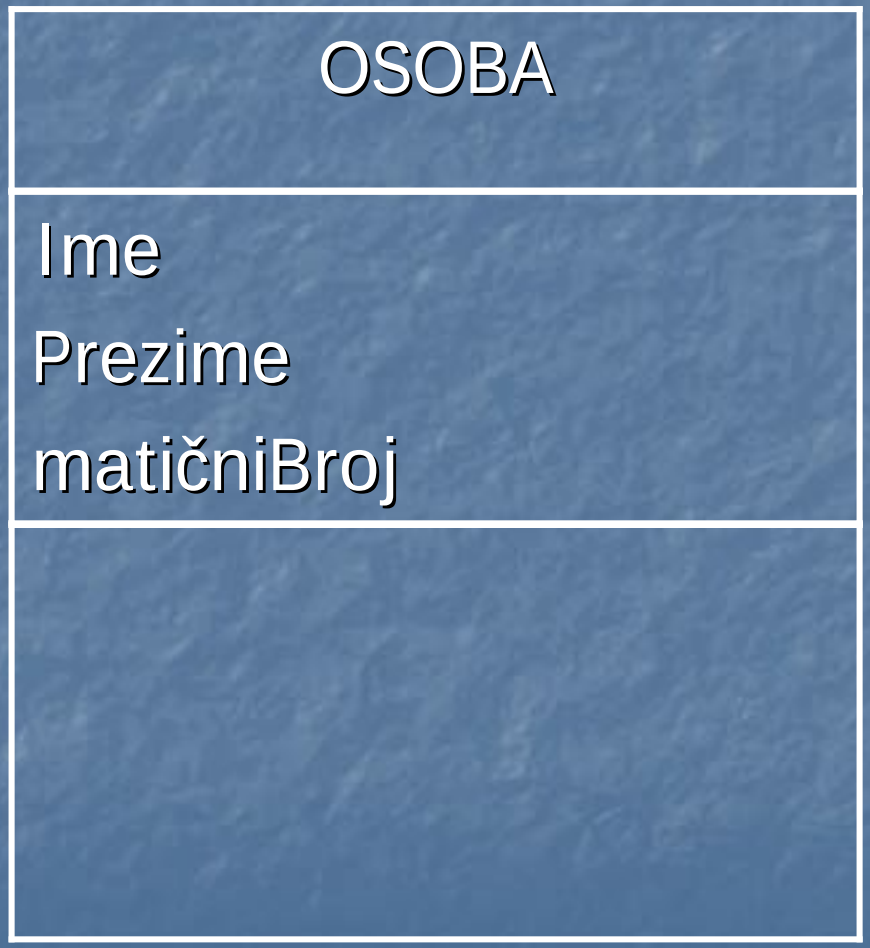
- Klasa obuhvata:
  - karakteristike (atribute)
  - ponašanje (metode)
  - odnose sa drugim klasama (realcije)

- Atributi, metode i relacije su elementi klase.

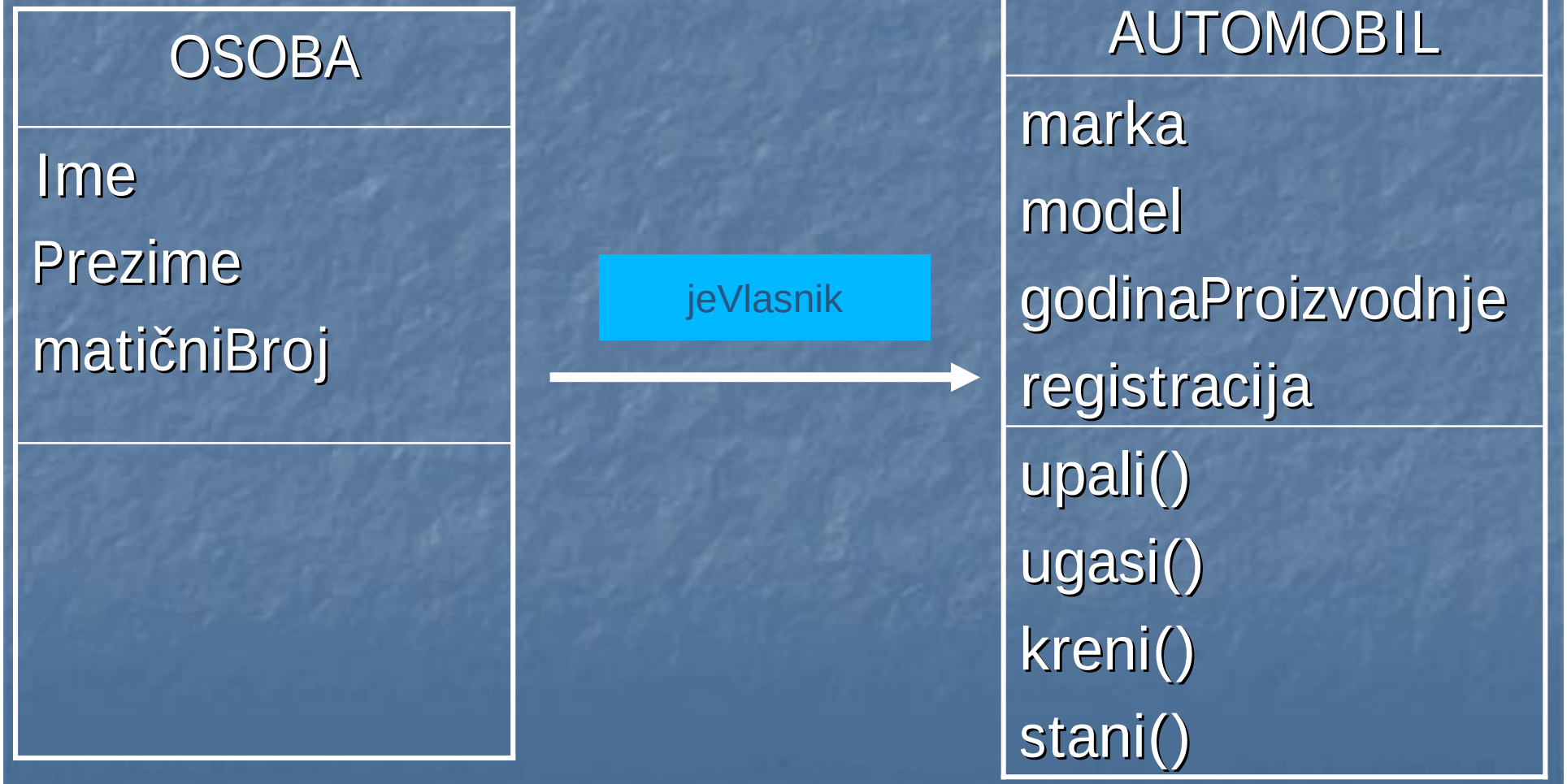
klasa Automobil



klasa Osoba



Relacije


OSOBA
Ime
Prezime
matičniBroj
jeVlasnik
AUTOMOBIL
marka
model
godinaProizvodnje
registracija
upali()
ugasi()
kreni()
stani()

Opšti oblik definicije klase u Javi

```
Class NazivKlase {
        //definicije atributa
        //definicije metoda
}
```

- Definicija klase rezervisanom reči `Class`, nakon čega ide naziv klase; ovaj deo je zaglavlje klase
- Otvorenom vitičastom zagradom započinje telo klase koje sadrži definiciju atributa i metoda
- Definicija klase se završava zatvorenom vitičastom zagradom

* napomena: Java je programski jezik koji pravi razliku između malih i velikih slova. To znači da će se nazivi klasa, atributa i metoda razlikovati u zavisnosti od veličine slova kojima su ispisani. Tako se promenavlasnika, promenaVlasnika i PROMENAVLSANIKA tretiraju kao potpuno različiti nazivi.

** Nazivi ne smeju da imaju nijedan blanko znak

*** U OOP postoji nepisano pravilo da se naziv klase piše velikim slovom, a ako se naziv sastoji od dve reči onda se svaka reč piše velikim slovom (npr. PoslovniCentar)

Komentari u Javi

- jednolinijski komentari

kratki komentari koji objašnjavaju ili pomažu razumevanj programskog koda;
označavaju se duplom kosom crtom “//”

- višelinijski komentari

duži komentari od nekoliko redova;
započinju kosom crtom i zvezdom “/*”, a završavaju zvezdom i kosom crtom “*/”

Atributi

- Atributi su karakteristike (osobine) klase koji se mogu izraziti putem nekog broja, slova ili niza slova

- Opšti oblik definicije atributa u Javi

```
tip_podatka nazivAtributa;
```

- Tip podatka predstavlja skup mogućih vrednosti atributa (ceo broj, realan broj, broj, slovo, niz slova ili nešto drugo)
- Naziv atributa se prema nepisanom pravilu piše malim slovom osim kada se sastoji iz dve reči kada se druga reč piše velikim slovom (npr. godinaProizvodnje, matičniBroj)
- Definicija atributa se završava tačka-zarezom “;”

Najčešće korišćeni tipovi podataka u Javi

Naziv tipa podatka	opis	primer
int	celi brojevi	1, -55, 10.000
double	realni brojevi	11.23, 0.12
char	znak (slovo, cifra, neki drugi znak)	‘a’, ‘5’, ‘!’
boolean	logička promenljiva	true, false
String	Niz znakova	“Milan”, “2456”
Calendar	Datum i vreme	2012-03-15 10:50

- Tipovi podataka
  - Prosti
  - Složeni (predstavljaju se korišćenjem klasa; String i Calendar su dve predefinisane Java klase)
- long, short, float

Primer 1: Napravi klasu automat novca. Ova klasa bi trebalo da ima samo atribut “stanje” koji predstavlja iznos novca koji se trenutno nalazi u automatu. Klasa nema metode i o tome treba dati komentar u jednoj liniji

```
Class AutomatNovca {
        double stanje;
        //ova klasa nema metode
}
```

Primer 1: Napravi klasu računar. Ova klasa bi trebalo da ima sledeće atribute: takt procesora (realan broj npr. 4.0 GhZ), radna memorija (realan broj npr. 2.0 Gb), hard disk (ceo broj npr. 120 Gb). Klasa nema metode i o tome treba dati komentar u više linija.

```
Class Racunar {
      double taktProcesora;
       double radnaMemorija;
       int hardDisk;
        /*ova klasa
           nema metode*/
}
```

Dodeljivanje početnih vrednosti atributa

- Atributima je moguće prilikom definisanja dodeliti podrazumevane, početne vrednosti

- Opšti oblik definicije atributa i dodeljivanja početnih vrednosti u Javi

```
tip_podatka nazivAtributa = vrednost;
```

Zadaci

- Napraviti klasu Televizor. Ova klasa bi trebalo da ima:
  - Atribut jačinaTona koji je ceo broj i označava trenutnu jačinu tona na televizoru. Početna vrednost ovog atributa je 0 (ton je utišan do kraja)
  - Atribut trenutniProgram koji označava broj programa koji je trenutno na televizoru (npr. uključen je program 5). Početna vrednost ovog atributa je 1.
  - Atribut uključen koji označava da li je televizor uključen ili nije (ako je uključen ima vrednost TRUE, a inače ima vrednost FALSE). Smatra se da je na početku televizor isključen.

Rešenje

```
Class Televizor {
      int jacinaTona = 0;
       int trenutniProgram = 1;
       boolean iskljucen = false;
}
```

Zadaci

- Napraviti klasu Radio. Ova klasa bi trebalo da ima:
  - Atribut fmFrekvencija koji označava trenutnu FM radio frekvenciju koju radio pušta (npr. 102.2 ili 87.5). Početna vrednost ovog atributa je 87.5.
  - Atribut amFrekvencija koji označava trenutnu AM radio frekvenciju koju radio pušta (npr. 567 ili 1500). Početna vrednost ovog atributa je 567.
  - Atribut band koji ima vrednost A ako radio pušta AM frekvenciju ili F ako radio pušta FM frekvenciju. Početna vrednost ovog atributa je F.

Rešenje

```
Class Radio {
      double fmFrekvencija = 87.5;
      int amFrekvencija = 567;
      char band = ‘F’;
}
```

Zadaci

- Napraviti klasu `Student`. Ova klasa bi trebalo da ima:
  - Atribut `ime`. Početna vrednost ovog atributa je “nepoznato”.
  - Atribut `prezime`. Početna vrednost ovog atributa je “nepoznato”.
  - Atribut `pol` koji može imati vrednost `M` ili `Z`.
  - Atribut `brojIndeksa` (niz slova).
  - Atribut `prosecnaOcena`.

Rešenje

```
Class Student {
      String ime = “nepoznato”;
      String prezime = “nepoznato”;
      char pol;
      String brojIndeksa;
      double prosecnaOcena;
}
```

Objekti

- Osnovni pojam u objekno orjentisanim programima je objekat koji predstavlja entitet iz realnog sveta. Objekati mogu biti konkretni entiteti kao na primer ljudi, organizacije, predmeti (studenti, knjige, fakulteti) ili događaji (ispiti).
- Objekti predstavljaju konkretan primerak odnosno pojavu neke klase, pa se klasa može definisati kao skup objekata koji imaju iste osobine

Odnos klase i objekata

- Automobil predstavlja klasu jer je to opšti nacrt nekih karakteristika i ponašanja koje svaki automobil ima.
- Aston Martin DB9 sa godinom proizvodnje 2008 i registracijom BG 123-456 predstavlja jedan konkretan primerak automobila tj. objekat klase automobila

Odnos klase i objekata

KLASA

AUTOMOBIL

marka
model
godinaProizvodnje
registracija

upali()
ugasi()
kreni()
stani()

OBJEKTI

a1:Automobil

marka = „Aston Martin“
model = „DB9“
godinaProizvodnje = 2008
registracija = „BG 123-456“

upali()
ugasi()
kreni()
stani()

a2:Automobil

marka = „Jugo“
model = „Koral 55“
godinaProizvodnje = 2000
registracija = „BG 333-444“

upali()
ugasi()
kreni()
stani()

a3:Automobil

marka = „Ford“
model = „Focus 1.6TDI“
godinaProizvodnje = 2008
registracija = „BG 120-406“

upali()
ugasi()
kreni()
stani()

Deklaracija objekata u Javi

- U Javi se objekti deklarišu na sličan način kao i atributi klase
- Prvo se navodi naziv klase pa onda naziv konkretnog objekta, tako da se stvara promenljiva koja će da referencira konkretan objekat

```
NazivKlase nazivobjekta;
```

Inicijalizacija objekta

- Da bi objekat mogao da se koristi (da se pozivaju njegove metode, menjaju vrednosti atributa itd.) potrebno ga je inicijalizovati.
- Ako se objektu pokuša pristupiti bez inicijalizacije, Java javlja grešku.
- Inicijalizacija se vrši korišćenjem naredbe `new`

```
nazivobjekta = new NazivKlase();
```

Primer 1

- Napraviti klasu Motocikl koja ima:
  - Atribut markaModel
  - Atribut kubikaza (ceo broj)

Napravi klasu Test koja sadrži main metodu i u okviru nje pravi dva objekta klase Motocikl

```
Class Motocikl {
        String markaModel;
         int kubikaza;
}
```

main metoda

- Da bi neki Java program mogao da se pokrene, potrebno je da ima tzv. main metodu
- Zaglavlje ove metode je uvek isto, a njena definicija se piše u okviru tela klase

```
public static void main (String [ ] args) {
        // naredbe ....
}
```

Primer 1

```
Class Motocikl {
        String markaModel;
        int kubikaza;
}
Class Test {
        public static void main (String [ ] args) {
        Motocikl m1;
        Motocikl m2;

        m1 = new Motocikl ();
        m2 = new Motocikl ();
        }
}
```

Vrednosti atributa objekta

- Objekat je pojavljivanje klase koje ima konkretnu vrednost atributa
- Da bi se vrednosti atributa promenile ili pročitale potrebno im je pristupiti na određeni način
- Pristup atributima objekta preko naziva objekta i naziva atributa

```
nazivobjekta.nazivAtributa
```

- Prepravite klasu Test koja sadrži main metodu tako da pravi dva objekata klase Motocikl. Prvi bi trebalo da bude Suzuki GS od 500 kubika (dodeli atributima prvog objekta ove vrednosti), a drugi Yamaha RS od 600 kubika (dodeli atributima drugog objekta ove vrednosti)

```
Class Motocikl {
      String markaModel;
      int kubikaza;
}
Class Test {
        public static void main (String [ ] args) {
        Motocikl m1;
        Motocikl m2;

        m1 = new Motocikl ();
        m2 = new Motocikl ();

        m1.markaModel = “Suzuki”;
        m1.kubikaza = 500;

        m2.markaModel = “Yamaha RS “;
        m2.kubikaza = 600;
        }
}
```

Standardni izlaz
komanda za ispisivanje na ekranu

`System.out.println (...neki tekst i/ili vrednost...)`

- Rezultat izvršavanje ove komande je ispis sadržaja koji se nazali između zagrada i prelazak u novi red
- Jedna od varijanti komande za ispisitvanje na ekranu je `print` naredbe; jedina razlika je što će sve vrednosti pisati u istom redu
- Sadržaj u zagradama može da bude neki tekst, vrednost nekog atributa ili kombinacija ova dva
- pr. 1

```
System.out.println (“Lep je dan”);
      // Ispisaće na ekranu:
     // Lep je dan
```

- pr. 2

```
int broj;
broj = 12;
System.out.println (broj);
      // Ispisaće na ekranu:
     // 12
```

- pr. 3

```
int broj;
broj = 12;
System.out.println (“Vrednost broja je: “+broj);
      // Ispisaće na ekranu:
     // Vrednost broja je: 12
```

Zadatak

- Napravite klasu Grad. Ova klasa bi trebalo da ima:
  - Atribut naziv. Početna vrednost ovog atributa je “nepoznat”.
  - Atribut brojStanovnika. Početna vrednost ovog atributa je 0.

Napravite klasu testGrad koja ima main metodu i u okviru nje kreira tri objekta klase grad: Beograd (2.000.000 stanovnika), Njujork (20.000.000 stanovnika) i Vankuver (nepoznat broj stanovnika – ne dodeljivati nikakvu vrednost). Potrebno je ispisati vrednosti atributa svih objekata na ekranu

```
Class Grad {
        String naziv = “nepoznat”;
      int brojStanovnika = 0;
}
Class testGrad {
        public static void main (String [ ] args) {
        Grad g1;
        Grad g2;
        Grad g3;

        g1 = new Grad ();
        g2 = new Grad ();
        g3 = new Grad ();

        g1.naziv = “Beograd”;
        g1.brojStanovnika = 2000000;

        g2.naziv = “Njujork“;
        g2.brojStanovnika = 20000000;

        g3.naziv = “Vankuver“;

        System.out.println (g1.naziv);
        System.out.println (g1.brojStanovnika);

        System.out.println (g2.naziv);
        System.out.println (g2.brojStanovnika);

        System.out.println (g3.naziv);
        System.out.println (g3.brojStanovnika);
        }
}
```