

1. Definiranje klasa. Konstruktori. Statičke metode i varijable

# Kategorizacija

▪ Kategorizacija: razvrstavanje stvari u grupe, tj. klase

▪ Program koji se razvija odnosi se na neku (poslovnu) domenu

▪ Svaka domena ima

* stvarne entitete koji se moraju kategorizirati (podijeliti u klase)
  + npr. fakultet: studenti, nastavnici, predmeti, ispiti, ...
  + entiteti imaju atribute: ime, prezime, naziv predmeta, ocjena, …
* postupke vezane uz pojedinu identificiranu kategoriju
  + npr. upis predmeta, prijava ispita, izračun ocjene, …
  + bit će modelirani kao metode pojedinih klasa
* i „poslovna” pravila
  + npr. „za ocjenu 5 treba imati minimalno 90% bodova”, …
  + bit će ugrađenu u postupke (metode)

# Apstrakcija u objektno orijentiranom programiranju

▪ Identificirani entiteti i akcije moraju se modelirati kao klase i metode

▪ Apstrakcija: modeliranje entiteta tako da se koriste samo bitne komponente stvarnog entiteta

▪ Što je bitno za npr. modeliranje studenta?

* ime, prezime i JMBAG su vjerojatno bitni
* datum rođenja, mentor, visina, težina, ... ? ovisno o namjeni sustava, tj. što modeliramo
  + radimo li sustav za organiziranje sportskih događaja?
  + ...ili možda sustav za evidenciju studija i izdavanje potvrda?

# Učahurivanje (enkapsulacija)

▪ Kako radi TV?

* sve dok radi i dok imamo sučelje za upravljanje TV-om nije bitno

▪ Hoćemo li studentove ocjene spremati u polju ili negdje drugdje? Je li nam bitno kako se zove varijabla koja sadrži ocjene?

* želimo li dopustiti bilo kome da mijenja podatke direktno ili isključivo putem ponuđenih metoda?
* potrebno je zaštititi implementacijske detalje od javnog pristupa kako bi se izbjegle slučajne ili namjerne pogreške

▪ Enkapsulacija (učahurivanje)

* veže podatke i metode koji rade nad tim podacima
* skriva implementacijske detalje i sprječava direktni pristup
* pridonosi slaboj povezanost objekata (engl. *loose coupling*)
  + slabom povezanošću objekti postaju neovisniji i interne promjene jednog objekta ne utječu na rad drug

# Konstruktori

▪ Posebna metoda koja služi za inicijalizaciju novog objekta (npr. pridruživanje vrijednosti članskim varijablama)

* ne može se direktno pozvati
  + izvršava se odmah nakon rezervacije memorije i stvaranja objekta
* naziv konstruktora isti kao i naziv klase
* može imati argumente, ali nema povratni tip (čak ni void)

▪ Klasa može imati nula ili više eksplicitno napisanih konstruktora

* ako nije napisan nijedan instruktor u Dartu, stvara se default-ni konstruktor
  + To ne znači da zadani konstruktor biti generiran u klasi. Umjesto toga, kada se kreira nova instanca klase, taj će se konstruktor pozvati. Neće imati argumente i pozvat će konstruktor super klase, također bez argumenata.

[*https://www.freecodecamp.org/news/constructors-in-dart/*](https://www.freecodecamp.org/news/constructors-in-dart/)

# Statičke varijable i metode

▪ riječ static koristi se za upravljanje memorijoma. Riječ static se može primijeniti na atribute i metode klase. Statičke varijable i metode dio su klase umjesto određene instance.

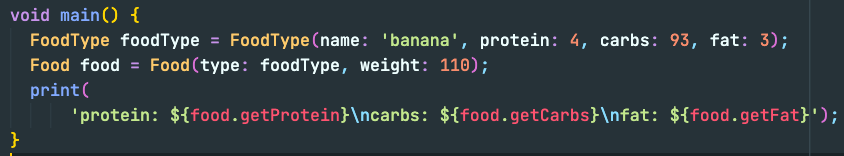
▪ riječ static koristi se za varijablu i metodu na razini klase koja je ista za svaku instancu klase, što znači da ako je član podataka statičan, može mu se pristupiti bez kreiranja objekta.

▪ riječ static omogućuje članovima podataka da traju vrijednosti između različitih instanci klase.

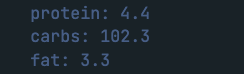
▪ nema potrebe stvarati objekt klase za pristup statičkoj varijabli ili pozivati statičku metodu: jednostavno se stavi naziv klase ispred statičke varijable ili naziva metode da bi se koristile.

<https://www.geeksforgeeks.org/dart-static-keyword/>

Zadaci za vježbu

1. Napravi klasu FoodType koja ima sljedeće javne atribute: name (String), protein (int), carbs (int), fat (int). Atributi protein, carbs i fat su postotci.
2. Dodaj klasi FoodType konstruktor koji prima kao parametre sve  
   vrijednosti za atribute. Napravite get metode za svaki atribut.
3. Dodaj u klasi FoodType javnu metodu toString koja nema argumenata, a vraća String koji predstavlja objekt. Ako atributi imaju vrijednost: name=banana, protein=4, carbs=93, fat=3 onda je String koji se vraća: "banana: p - 4%, c - 93%, f - 3%".
4. Napravi statičku varijablu counter koja sadrži broj kreiranih objekata iz klase FoodType. Varijabla counter prilikom deklaracije postavlja vrijednost 0, a u konstruktoru se povećava za jedan. Napravi statičku metodugetNumberOfCreatedInstances koja vraća counter.
5. Napravi klasu Food koja ima dva atributa: type (FoodType), weight(int). Težina je u gramima. Napravi konstruktor sa svim atributima te dodajte get metode i metodu toString koja nema argumenata i vraća String koji predstavlja objekt u sljedećem formatu: "banana: p - 4%, c - 93%, f - 3%, w – 110g".
6. Dodaj u klasu Food get metode za protein, carbs i fat, ali tako da vraćaju te vrijednosti u gramima (double), a ne u postocima kako je zapisano u klasi FoodType. Npr. sljedeći isječak:  
   

će ispisati:



1. Dodaj u klasu Food javnu metodu sa sljedećim potpisom String toStringInGrams() koja predstavlja objekt u formatu: "banana: p – 4.4g, c – 102.3g, f – 3.3g, w – 110g ". Brojevi s decimalnom točkom se trebaju ispisivati na jednu decimalu.