

1. Nasljeđivanje. Polimorfizam

# Klasa Object

▪ osnovna klasa za sve dart objekte osim null

▪ korijen hijerarhije Dart klasa koje nisu null

* Svaka druga klasa koja nije null je podklasa klase Object i sadrži metode iz Object klase

# Zašto nasljeđivanje?

▪ Želimo napraviti novu klasu, a već postoji klasa koja uključuje nešto što trebamo (kôd, članske varijable, metode...)?

▪ Umjesto pisanja klase ispočetka, nasljeđivanjem imamo mogućnost korištenja već postojećih metoda

▪ Novu klasu „obogatimo” svojstvima koja nedostaju - specifičnim samo za tu klasu

# Nadjačavanje

▪ Nadjačavanje – engl. *override*

▪ Nadjačavanje definira novo ponašanje naslijeđene metode

* Metoda mora imati isti potpis (naziv, povratni tip, broj i tipovi argumenata)
* Napomena: novo ponašanje se primjenjuje na klasu u kojoj je došlo do nadjačavanja i na klase ispod u hijerarhiji

# 

# Bazna i izvedena klasa

▪ **Bazna** klasa (osnovna klasa, natklasa, klasa roditelj) je klasa iz koje se izvodi neka druga klasa

* engl. ***base class, superclass***

▪ Klasa koja nasljeđuje neku klasu naziva se **izvedena** klasa (potklasa, klasa dijete)  
▪ engl. ***derived class****,* ***subclass***

▪ Izvedena klasa je specijalizacija bazne klase  
▪ Bazna klasa je generalizacija svojih izvedenih klasa

# Nasljeđivanje

▪ Izvedena klasa se sastoji od vlastitih članova i članova bazne klase  
▪ ne može pristupiti privatnim članovima bazne klase

▪ Konstruktori se ne nasljeđuju, ali se mogu pozivati iz izvedene klase

▪ U Javi sve klase osim klase *Object* imaju jednu i samo jednu direktnu klasu roditelja.

▪ Ako to nije neka eksplicitno navedena klasa, onda je to klasa *Object.*

# Polimorfizam (engl. Polymorphism)

▪ „poly“ znači mnogo, „morph“ znači preoblikovanje u različite oblike

* zajedno znače stvatanje mnogih oblika ili konfiguracija.

▪ primjer u stvarnom svijetu

* recimo da kupujemo novu značajku automobila
* nova značajka je omogućavanje samostalne vožnje
* tada umjesto osnovne značajke vožnje koja zahtjeva ljudsku podršku, ažuriramo funkcionalnost da se automobil može voziti sam, bez ljudske podrške

▪ sažetak: Polimorfizam je ažuriranje ili modificiranje značajke,funkcije ili implementacije

koje već postoje u roditeljskoj klasi

Zadaci za vježbu

1. Napravi klasu Dessert koja ima sljedeće atribute: name (String), weight (double) i calories (int). Dodaj klasi konstruktor koji prima kao parametre vrijednosti za sva 3 atributa. Napravi get i set metode za svaki atribut, te nadjačaj metodu toString. Napišite i metodu getDessertType koja nema argumenata a vraća string „dessert“.
2. Napravi klase Cake i IceCream koje nasljeđuju Dessert. Kolač, uz sve atribute iz klase Dessert ima i atribute containsGluten (boolean) i cakeType (String, može biti „birthday“, „wedding“, „regular“ i sl.). Sladoled ima dodatne atribute flavour (String) i color (String). Napravi get i set metode za svaki atribut, kao i metodu toString koja vraća sve što vraća i metoda toString iz Klase Dessert, a dodatno još i atribute specifične za izvedenu klasu. Napiši metodu getDessertType u svakoj od izvedenih klasa, koja će za sladoled vratiti tekst „ice cream“, a za tortu vrijednost atributa cakeType i tekst „ cake“. Napiši glavni program s kojim ćete testirati sve zadane funkcionalnosti.