

Zadaci za vježbu iz teme 3 (Definiranje klasa i prava pristupa. Konstruktori. Varijabilni broj argumenata. Statičke metode i varijable.)

U bilo kojem zadatku možete napisati dodatne privatne metode po potrebi.

1. Napravite klasu `FoodType` koja ima sljedeće javne atribute: `name (String)`, `protein (int)`, `carbs (int)`, `fat (int)`. Atributi `protein`, `carbs` i `fat` su postoci.
2. Promijenite klasu `FoodType` tako da su atributi privatni. Dodajte klasi konstruktor koji prima kao parametre sve vrijednosti za atribute. Napravite `get` metode za svaki atribut.
3. Dodajte 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. Napravite 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. Napravite statičku metodu `getNumberOfCreatedInstances` koja vraća `counter`.
5. Napravite klasu `Food` koja ima dva atributa: `type (FoodType)`, `weight(int)`. Težina je u gramima. Napravite 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. Dodajte 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:

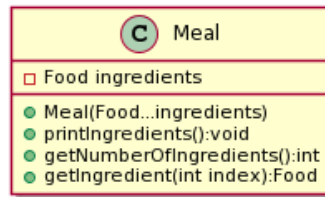
```
FoodType foodType = new FoodType("banana", 4, 93, 3);
Food food = new Food(foodType, 110);
System.out.println("protein: " + food.getProtein() + "\ncarbs: " +
    food.getCarbs() + "\nfat: " + food.getFat());
```

će ispisati:

```
protein: 4.4
carbs: 102.3
fat: 3.3
```

7. Dodajte 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.

8. Napravite klasu Meal kako je definirana u UML dijagramu klasa. Meal se sastoji od sastojaka (klasa Food). Konstruktor prima varijabilni broj elemenata klase Food. Ne želimo da nam nitko izvan klase može mijenjati sastojke. Kod vraćanja sastojka s indeksom koji ne postoji potrebno je vratiti vrijednost null.



Legenda:

Character	Icon for field	Icon for method	Visibility
-	□	■	private
#	◇	◆	protected
~	△	▲	package private
+	○	●	public

Za sljedeći isječak:

```
FoodType banana = new FoodType("banana", 4, 93, 3);
Food oneBanana = new Food(banana, 110);

FoodType grapes = new FoodType("grapes", 4, 94, 2);
Food grapeCluster = new Food(grapes, 151);

Meal fruitSalad = new Meal(oneBanana, grapeCluster);
fruitSalad.printIngredients();

System.out.println("*** printing ingredients");
for (int i = -1; i <= fruitSalad.getNumberOfIngredients(); i++) {
    System.out.println("ingredient[" + i + "] = " + fruitSalad.getIngredient(i));
}
```

će se ispisati:

```
Ingredient[0] = banana: p - 4.4g, c - 102.3g, f - 3.3g, w - 110g
Ingredient[1] = grapes: p - 6.0g, c - 141.9g, f - 3.0g, w - 151g
*** printing ingredients
ingredient[-1] = null
ingredient[0] = banana: p - 4%, c - 93%, f - 3%, w - 110g
ingredient[1] = grapes: p - 4%, c - 94%, f - 2%, w - 151g
ingredient[2] = null
```

9. Dodajte u klasu Meal konstruktor koji prima objekt Meal i weight u gramima. weight predstavlja novi obrok koji ima količinu (suma težine svih sastojaka), a omjeri sastojaka su jednaki kao u objektu Meal. Za sljedeći isječak koda:

```
FoodType banana = new FoodType("banana", 4, 93, 3);
Food oneBanana = new Food(banana, 110);

FoodType grapes = new FoodType("grapes", 4, 94, 2);
Food grapeCluster = new Food(grapes, 151);

Meal fruitSalad = new Meal(oneBanana, grapeCluster);
System.out.println("* original meal:");
fruitSalad.printIngredients();

System.out.println("* new meal:");
Meal doubleFruitSalad = new Meal(fruitSalad, 522);
doubleFruitSalad.printIngredients();
```

će se ispisati:

```
* original meal:
Ingredient[0] = banana: p - 4.4g, c - 102.3g, f - 3.3g, w - 110g
Ingredient[1] = grapes: p - 6.0g, c - 141.9g, f - 3.0g, w - 151g
* new meal:
Ingredient[0] = banana: p - 8.8g, c - 204.6g, f - 6.6g, w - 220g
Ingredient[1] = grapes: p - 12.1g, c - 283.9g, f - 6.0g, w - 302g
```

10. Dodajte u klasu Meal javnu metodu compare koja prima drugi Meal i vraća double koji predstavlja sličnost (u postocima) između ta dva obroka. Dva obroka su potpuno jednaka ako imaju iste sastojke i u istim omjerima bez obzira na ukupnu težinu. Npr. prvi obrok koji ima 500 g mlijeka i 200 g palente uspoređujemo s drugim obrokom koji ima 200 g mlijeka i 100 g kukuruznih pahuljica.

$$\text{prvi obrok} \quad \frac{500g}{700g} \text{ mlijeka} = 71,4\% \text{ mlijeka} \quad \frac{200g}{700g} \text{ palente} = 28,6\% \text{ palente}$$

$$\text{drugi obrok} \quad \frac{200g}{300g} \text{ mlijeka} = 66,6\% \text{ mlijeka} \quad \frac{100g}{300g} \text{ pahuljica} = 33,3\% \text{ pahuljica}$$

Ta dva obroka imaju sličnost od 66,6% jer je jedini zajednički sastojak jedino mlijeko u postotku 66,6%. Pretpostavite da u jednom obroku nema više puta istog sastojka.

Rješenja zadataka dostupna su na sljedećim poveznicama:

1. https://github.com/FER-OOP/Lectures/blob/master/Exercises/Homework-03/src/main/java/hr/fer/oop/homework_03/e01/FoodType.java
2. https://github.com/FER-OOP/Lectures/blob/master/Exercises/Homework-03/src/main/java/hr/fer/oop/homework_03/e02/FoodType.java
3. https://github.com/FER-OOP/Lectures/blob/master/Exercises/Homework-03/src/main/java/hr/fer/oop/homework_03/e03/FoodType.java
4. https://github.com/FER-OOP/Lectures/blob/master/Exercises/Homework-03/src/main/java/hr/fer/oop/homework_03/e04/FoodType.java
5. https://github.com/FER-OOP/Lectures/blob/master/Exercises/Homework-03/src/main/java/hr/fer/oop/homework_03/e05/Food.java
6. https://github.com/FER-OOP/Lectures/blob/master/Exercises/Homework-03/src/main/java/hr/fer/oop/homework_03/e06/Food.java
7. https://github.com/FER-OOP/Lectures/blob/master/Exercises/Homework-03/src/main/java/hr/fer/oop/homework_03/e07/Food.java
8. https://github.com/FER-OOP/Lectures/blob/master/Exercises/Homework-03/src/main/java/hr/fer/oop/homework_03/e08/Meal.java
9. https://github.com/FER-OOP/Lectures/blob/master/Exercises/Homework-03/src/main/java/hr/fer/oop/homework_03/e09/Meal.java
10. https://github.com/FER-OOP/Lectures/blob/master/Exercises/Homework-03/src/main/java/hr/fer/oop/homework_03/e10/Meal.java

Kod svakog rješenja u istom paketu obično postoji i klasa Main s kojom možete isprobati korištenje zadane funkcionalnosti.