

Week 1 - oefening 2: BMI

Doelstelling

Introductie tot de programmeertaal Java
Creatie van eerste java-applicatie
OOP – herhaling
Gebruik van Enumeration

Algemene beschrijving

We maken een console Java-applicatie dat in staat is het BMI (Body-Mass-index) van een volwassen persoon te berekenen. Hiervoor gebruiken we volgende formule:

$$\text{BMI} = \text{gewicht} / (\text{lengte} * \text{lengte})$$

Hou rekening met:

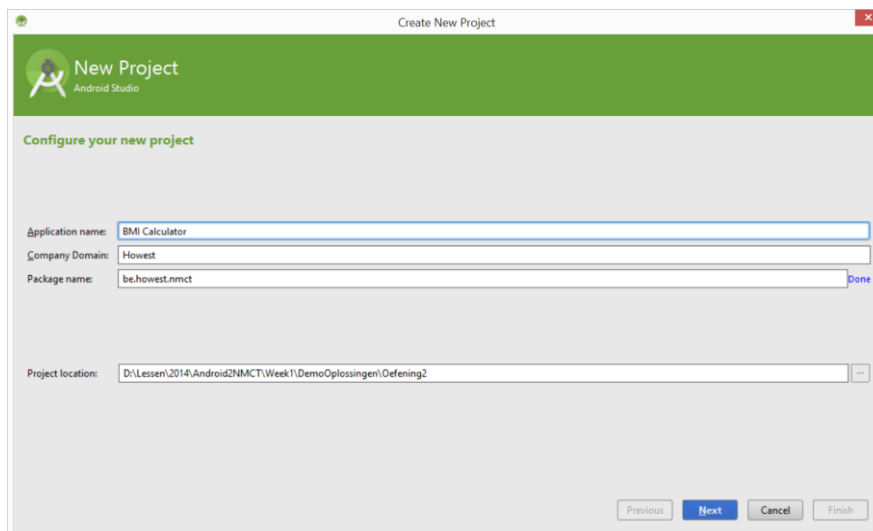
- het gewicht moet in kg uitgedrukt worden
- de lengte dient in meter uitgedrukt worden.

Betekenis:

- Groot ondergewicht: bmi is kleiner dan of gelijk aan 15
- Ernstig ondergewicht: bmi is groter dan 15, en is kleiner dan of gelijk aan 16
- Ondergewicht: bmi is groter dan 16 en is kleiner dan of gelijk aan 18.5
- Normaal: bmi is groter dan 18.5 en is kleiner dan of gelijk aan 25
- Overgewicht: bmi is groter dan 25 en is kleiner dan of gelijk aan 30
- Matig overgewicht: bmi is groter dan 30 en is kleiner dan of gelijk aan 35
- Ernstig overgewicht: bmi is groter dan 35 en is kleiner dan of gelijk aan 40
- Zeer groot overgewicht: bmi is groter dan 40.

Voorbereiding

Volg dezelfde stappen als opgave 1 een Android-Studio project 'BMI Calculator' aan.



Code

Maak een Klasse **BMIInfo** aan met volgende attributen:

- height: float met default waarde 1.70f
- mass: int met default waarde 70
- bmiIndex: float

Voorzie volgende methodes:

- een get- en set-methode voor height en mass; set-methode voor bmiIndex
- constructor(en)
- de methode recalculate() berekent bmiIndex
- een methode dat een categorie terug geeft. Hiervoor wordt een enumeration gebruikt (zie onder).

Voorzie in de klasse een public enumeration met de naam 'Category'. Een enumeration kan veel meer bevatten dan enkel een opsomming van constanten. Baseer u op het voorbeeld 'Planet' dat op onderstaande link te vinden is: <http://docs.oracle.com/javase/tutorial/java/javaOO/enum.html>

In de enumeration som je de verschillende categorie-namen op, samen met hun bijhorende buitengrenzen. Breid hiervoor de enumeration uit met:

- twee float-attributen: 'lowerBoundary' en 'upperBoundary'
- een constructor met twee float-parameters
- get-methodes om beide float-attributen terug te geven
- 'isInBoundary' methode met float-parameter dat controleert of parameter binnen de twee grenzen valt. Indien zo wordt true teruggegeven, false in het andere geval.
- static-methode 'getCategory' dat de juiste Category teruggeeft

```
public static Category getCategory(float index) {
    for(Category category : Category.values()) {
        if(category.isInBoundary(index)) return category;
    }
    return null;
}
```

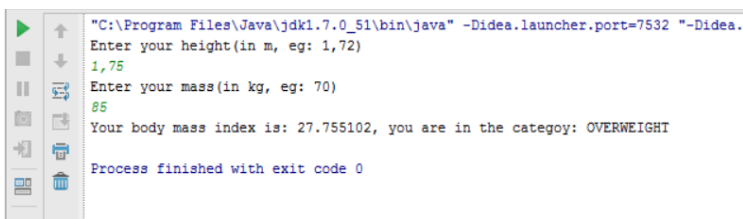
Maak een main-klasse 'RunBMI' aan waarin we de vorige klasse testen:

- maak via de constructor een aantal objecten van de klasse BMIInfo aan. Vraag de berekende BMI-waarde en de bijhorende categorienaam op
- vraag aan de gebruiker verschillende waarden aan: lengte en gewicht. Bereken vervolgens het BMI en geef feedback aan de gebruiker terug.

Het inlezen van waarden vanuit het console venster kan via de klasse Scanner. Meer info hieromtrent is te vinden op:

<http://docs.oracle.com/javase/7/docs/api/java/util/Scanner.html>

Mogelijke uitvoering:



```
"C:\Program Files\Java\jdk1.7.0_51\bin\java" -Didea.launcher.port=7532 "-Didea.
Enter your height(in m, eg: 1,72)
1,75
Enter your mass(in kg, eg: 70)
85
Your body mass index is: 27.755102, you are in the category: OVERWEIGHT
Process finished with exit code 0
```