

# HO GENT

OOSDII

*Lambda expressies - Oefeningen*

# Table of Contents

- 1. Oefening - Container ..... 1
  - 1.1. Gegeven..... 1
  - 1.2. Opdracht ..... 1

# 1. Oefening - Container



Werk verder op de oefening Container uit vorig hoofdstuk (Polymorfisme en Interfaces).

## 1.1. Gegeven

Container
<<Property>> -eigenaar : String <<Property>> -volume : int <<Property>> -massa : int <<Property>> -serialNumber : Integer
+Container(eigenaar : String, volume : int, massa : int, serialNumber : int) -setEigenaar(eigenaar : String) : void -setVolume(volume : int) : void -setMassa(massa : int) : void -controleerSerialNumber(serialNumber : Integer) : void

## 1.2. Opdracht

In de vorige oefening werden twee Java klassen aangemaakt die elk de Comparator interface implementeren: één om containers op **massa** te sorteren, één om containers op **eigenaar** te sorteren.

- Stap 1:
  - Pas de code aan zodat de klasse die sorteren op **massa** mogelijk maakt wordt vervangen door een 'anonymous inner class' in de applicatie.

Containers bij sorteren op massa:

90kg - Calais - 80m<sup>2</sup>

100kg - Brugge - 70m<sup>2</sup>

110kg - Rotterdam - 70m<sup>2</sup>

150kg - Antwerpen - 60m<sup>2</sup>

- Stap 2:
  - Pas de code aan zodat de klasse die sorteren op **eigenaar** mogelijk maakt wordt vervangen door een lambda expressie in de applicatie.

Containers bij sorteren op eigenaar:

Antwerpen - 60m<sup>2</sup> - 150kg

Brugge - 70m<sup>2</sup> - 100kg

Calais - 80m<sup>2</sup> - 90kg

Rotterdam - 70m<sup>2</sup> - 110kg

- Stap 3:

- Breid de code uit zodat de applicatie ook sorteert op serienummer. Implementeer dit met een 'method reference' en laat het systeem de Comparator zelf opstellen.

Containers bij sorteren op serienummer:

1234 - Antwerpen - 60m<sup>2</sup> - 150kg

2568 - Rotterdam - 70m<sup>2</sup> - 110kg

8564 - Brugge - 70m<sup>2</sup> - 100kg

8569 - Calais - 80m<sup>2</sup> - 90kg

Pas nu de code aan zodat de sortering in omgekeerde volgorde gebeurt.