HO
GENT

OOSDII

Lambda expressies Werkcollege

Table of Contents

1. Oefening 1

1.1. Gegeven Movie - Comparator interface - Anonymous inner class

Werk verder op de oefening uit het werkcollege rond polymorfisme en interfaces waarbij een Comparator interface werd geïmplementeerd (Oefening 3).

Implementeer het gedrag uit de klasse `RatingCompare` opnieuw gebruik makende van een 'anonymous inner class' binnen de klasse ComparatorApp.

```
Sorted by rating
8.8 Empire Strikes Back 1980
8.7 Star Wars 1977
8.4 Return of the Jedi 1983
8.3 Force Awakens 2015
```

2. Oefening 2

2.1. Gegeven Movie - Comparator interface - Lambda Expressie

Werk verder op de vorige oefening.

Implementeer het gedrag uit de 'anonymous inner class' opnieuw gebruik makende van een 'Lambda expressie' binnen de klasse ComparatorApp.

Zorg ervoor dat de geïmplementeerde lambda expressie slechts bestaat uit één statement. Een rating heeft het type Double. De klasse Double implementeert de interface 'Comparable'. Maak hiervan gebruik.

```
Sorted by rating
8.8 Empire Strikes Back 1980
8.7 Star Wars 1977
8.4 Return of the Jedi 1983
8.3 Force Awakens 2015
```

Implementeer ook een tweede comparator d.m.v. een lambda expressie dewelke de lijst van Movie objecten sorteert op basis van hun naam.

```
Sorted by name
Empire Strikes Back 8.8 1980
Force Awakens 8.3 2015
Return of the Jedi 8.4 1983
Star Wars 8.7 1977
```

3. Oefening 3

3.1. Gegeven Movie - Comparator interface - Method reference

Werk verder op de vorige oefening.

Implementeer in de klasse Movie volgende methode, het gedrag is identiek aan de implementatie in voorgaande oefeningen om twee Movie objecten met elkaar te vergelijken op basis van hun `rating`.

```
1 public static int compareRating(Movie m1, Movie m2)
```

Vervang nu de lambda expressie in de ComparatorApp door gebruik te maken van een methode referentie naar bovenstaande methode.

```
Sorted by rating
8.8 Empire Strikes Back 1980
8.7 Star Wars 1977
8.4 Return of the Jedi 1983
8.3 Force Awakens 2015
```

Wil je ook voor het sorteren op naam, gebruik maken van een method reference, dan schrijf je ofwel een (static) methode uit in Movie die je dan aanroept, ofwel maak je gebruik van de bestaande comparing()- methode uit Comparator (voorbeeld van een static interface methode) die op basis van de meegegeven parameter de collectie sorteert.

```
1 public static <T, U extends Comparable<? super U>> Comparator<T> comparing(
  Function<? super T, ? extends U> keyExtractor)
```

```
Sorted by name
Empire Strikes Back 8.8 1980
Force Awakens 8.3 2015
Return of the Jedi 8.4 1983
Star Wars 8.7 1977
```

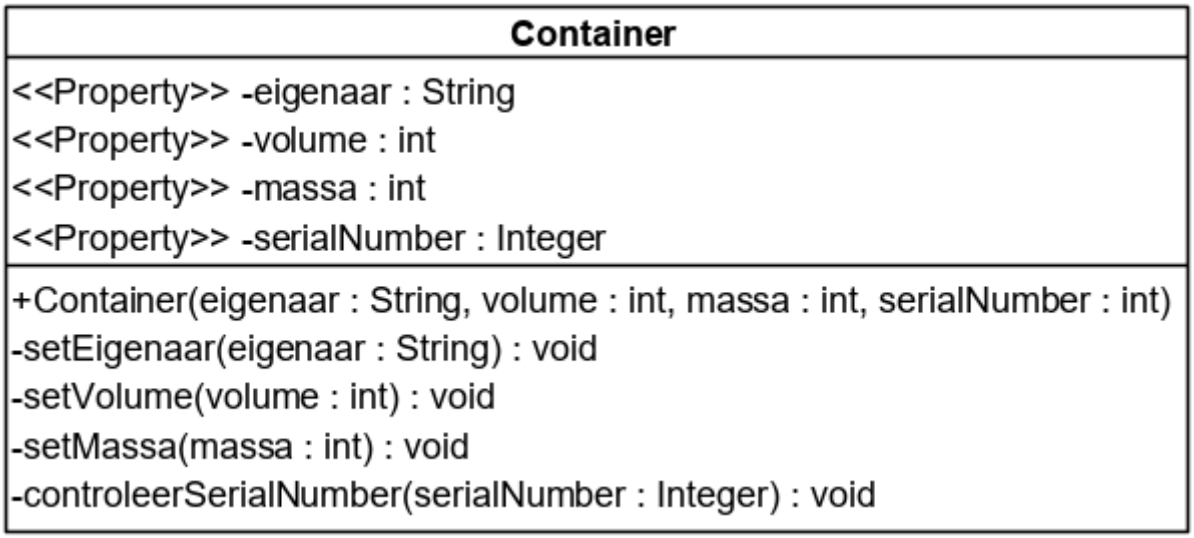
Vervang nu ook de sortering van de rating en laat hier het systeem een comparator opstellen op

basis van de rating. Enige probleem nu is dat de rating op 'natuurlijke wijze' zal gesorteerd worden, en de oorspronkelijke sortering was de omgekeerde natuurlijke wijze (van hoog naar laag). Ook deze comparator kan je door het systeem laten opstellen door de sortering om te keren m.b.v. de methode `.reversed()`.

4. Oefening - Container

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

4.1. Gegeven



4.2. 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 - 80m²
    100kg - Brugge - 70m²
    110kg - Rotterdam - 70m²
    150kg - Antwerpen - 60m²
    ```

- 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 - 60m² - 150kg
Brugge - 70m² - 100kg
Calais - 80m² - 90kg
Rotterdam - 70m² - 110kg
```

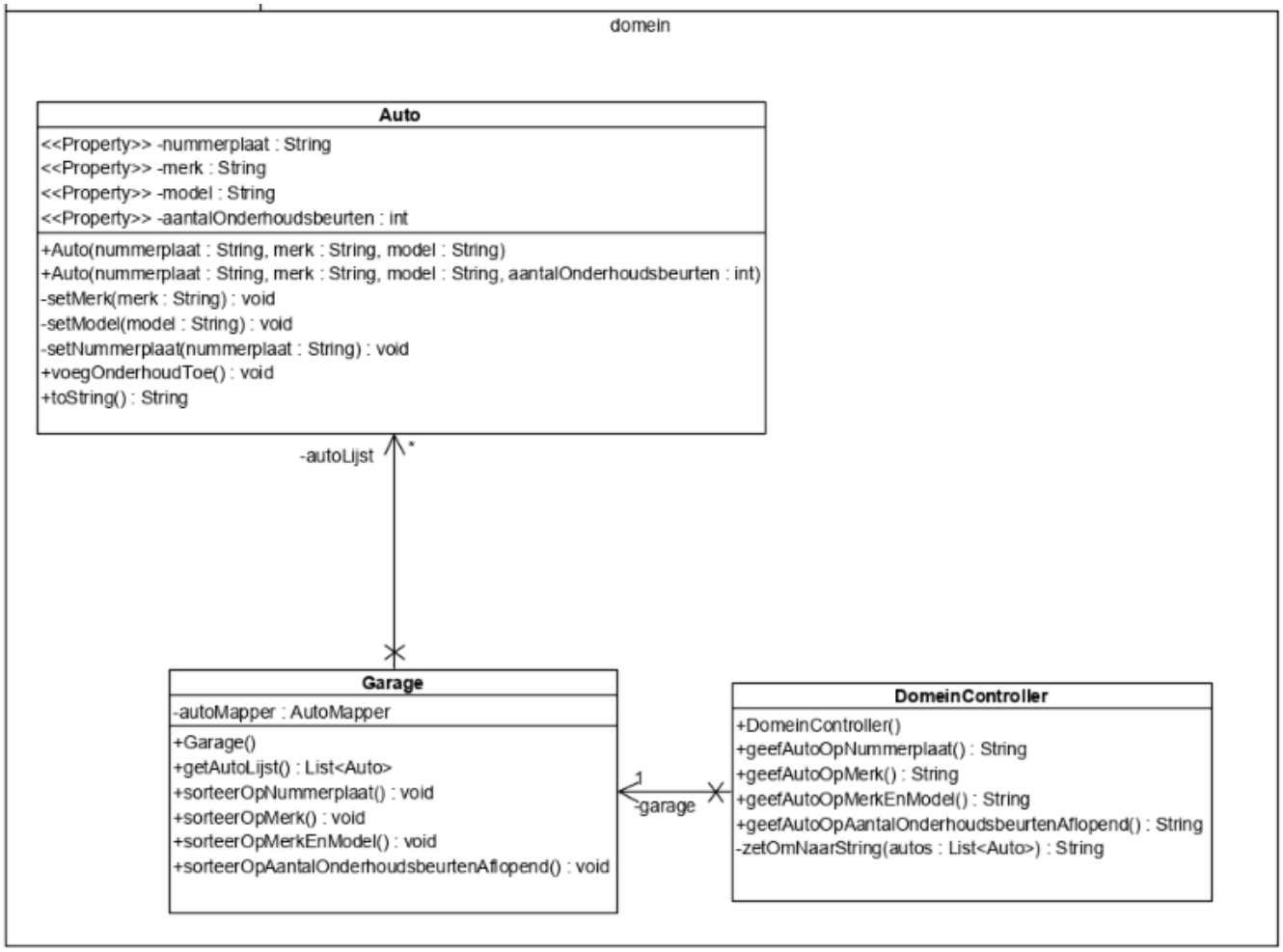
- Stap 3:
  - Breid de code uit zodat de applicatie ook sorteert op serienummer. Implementeer dit met een 'method reference' en laat het systeem de Comparator zelf opstellen.

```
Containers bij sorteren op serienummer:
1234 - Antwerpen - 60m² - 150kg
2568 - Rotterdam - 70m² - 110kg
8564 - Brugge - 70m² - 100kg
8569 - Calais - 80m² - 90kg
```

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

5. Oefening Garage

5.1. Gegeven



Maak gebruik van het startproject op Chamilo.

- Stap 1 :
  - Vul de klasse Auto aan met de volgende informatie:
    - Twee auto zijn gelijk als ze dezelfde nummerplaat hebben
    - Auto’s worden op natuurlijke wijze gesorteerd op nummerplaat (alfabetisch)
- Stap 2 :
  - Vervolledig de sorteermethodes in de klasse Garage. Maak gebruik van anonieme innerklassen en ook lambda expressies en/of method references.
- Stap 3 :
  - Vervolledig de methodes in de klasse DomeinController zodat de applicatie de auto’s op de juiste manier kan tonen.

```
Alle auto's oplopend op nummerplaat:
Toyota Yaris met nummerplaat 123xyz, 6 onderhoudsbeurt(en)
Opel Astra met nummerplaat 456abc, 4 onderhoudsbeurt(en)
Renault Fluence met nummerplaat 567xyz, 5 onderhoudsbeurt(en)
Mercedes C-klasse Berline met nummerplaat 789cde, 2 onderhoudsbeurt(en)
Opel Zafira met nummerplaat ab12ab, 7 onderhoudsbeurt(en)
BMW Berline met nummerplaat azerty, 0 onderhoudsbeurt(en)
Toyota Avensis met nummerplaat qwerty, 8 onderhoudsbeurt(en)
Peugeot 308 met nummerplaat xy12xy, 9 onderhoudsbeurt(en)

Alle auto's oplopend op merk:
BMW Berline met nummerplaat azerty, 0 onderhoudsbeurt(en)
Mercedes C-klasse Berline met nummerplaat 789cde, 2 onderhoudsbeurt(en)
Opel Astra met nummerplaat 456abc, 4 onderhoudsbeurt(en)
Opel Zafira met nummerplaat ab12ab, 7 onderhoudsbeurt(en)
Peugeot 308 met nummerplaat xy12xy, 9 onderhoudsbeurt(en)
Renault Fluence met nummerplaat 567xyz, 5 onderhoudsbeurt(en)
Toyota Yaris met nummerplaat 123xyz, 6 onderhoudsbeurt(en)
Toyota Avensis met nummerplaat qwerty, 8 onderhoudsbeurt(en)

Alle auto's oplopend op merk en model :
BMW Berline met nummerplaat azerty, 0 onderhoudsbeurt(en)
Mercedes C-klasse Berline met nummerplaat 789cde, 2 onderhoudsbeurt(en)
Opel Astra met nummerplaat 456abc, 4 onderhoudsbeurt(en)
Opel Zafira met nummerplaat ab12ab, 7 onderhoudsbeurt(en)
Peugeot 308 met nummerplaat xy12xy, 9 onderhoudsbeurt(en)
Renault Fluence met nummerplaat 567xyz, 5 onderhoudsbeurt(en)
Toyota Avensis met nummerplaat qwerty, 8 onderhoudsbeurt(en)
Toyota Yaris met nummerplaat 123xyz, 6 onderhoudsbeurt(en)

Alle auto's oplopend op aantal :
Peugeot 308 met nummerplaat xy12xy, 9 onderhoudsbeurt(en)
Toyota Avensis met nummerplaat qwerty, 8 onderhoudsbeurt(en)
Opel Zafira met nummerplaat ab12ab, 7 onderhoudsbeurt(en)
Toyota Yaris met nummerplaat 123xyz, 6 onderhoudsbeurt(en)
Renault Fluence met nummerplaat 567xyz, 5 onderhoudsbeurt(en)
Opel Astra met nummerplaat 456abc, 4 onderhoudsbeurt(en)
Mercedes C-klasse Berline met nummerplaat 789cde, 2 onderhoudsbeurt(en)
BMW Berline met nummerplaat azerty, 0 onderhoudsbeurt(en)
```