

HO GENT

OOSDII

Lambda expressies Werkcollege

Table of Contents

1. Oefening 1	1
1.1. Gegeven Movie - Comparator interface - Anonymous inner class	1
2. Oefening 2	1
2.1. Gegeven Movie - Comparator interface - Lambda Expressie	1
3. Oefening 3	2
3.1. Gegeven Movie - Comparator interface - Method reference	2

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

Ook de tweede comparator (die sorteert op basis van de naam) gaan we vervangen. We laten nu het systeem zelf de implementatie opstellen en het Comparator object instantiëren. Maak gebruik van volgende klasse methode uit de Comparator klasse.

```
1 public static <T, U extends Comparable<? super U>> Comparator<T> comparing(  
2     Function<? super T, ? extends U> keyExtractor)  
3     {  
4         Objects.requireNonNull(keyExtractor);  
5         return (Comparator<T> & Serializable)  
6             (c1, c2) -> keyExtractor.apply(c1).compareTo(keyExtractor.apply(c2));  
7     }
```

```
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()`.