

**H  
O  
GENT**

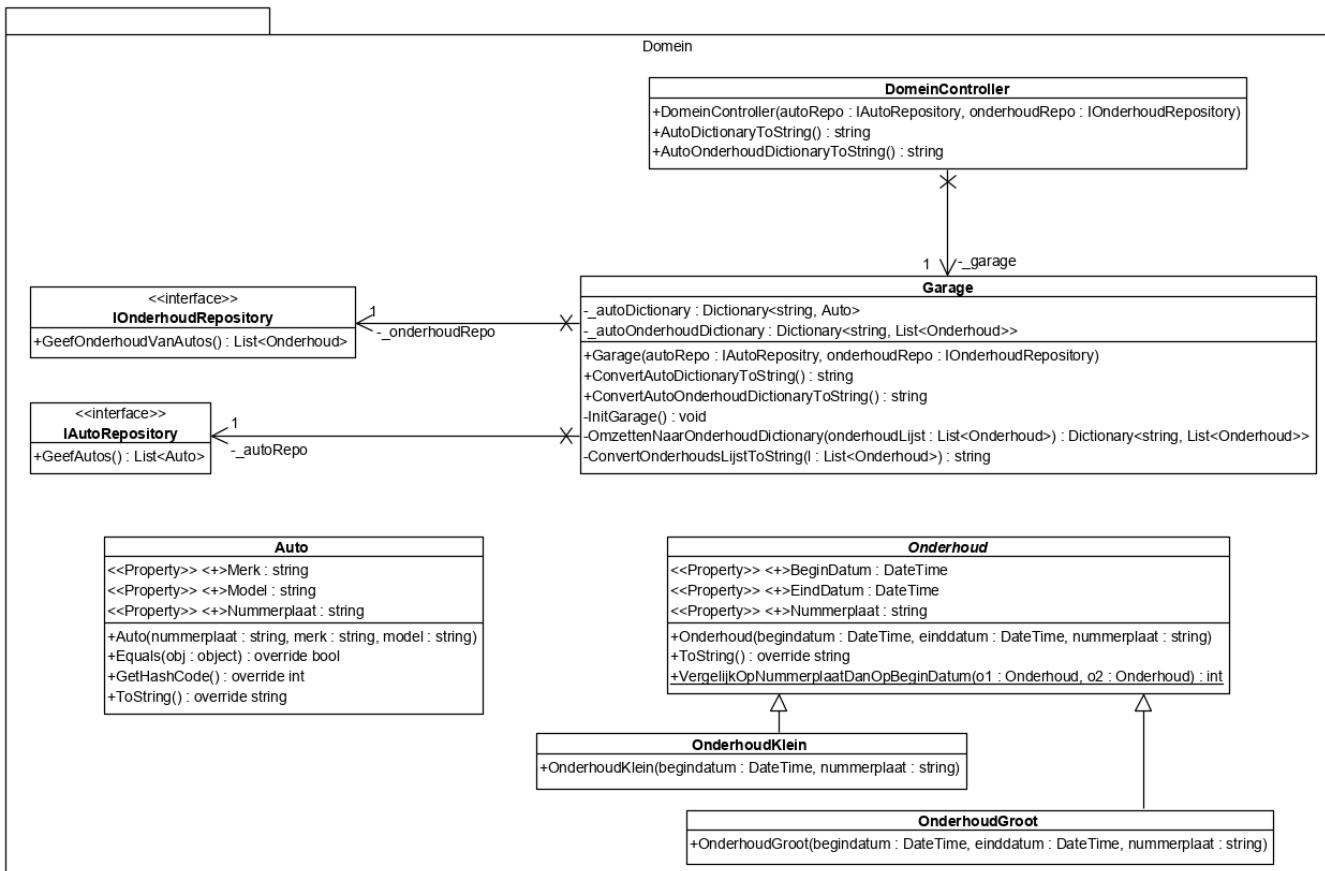
Delegates en Lambdas - Werkcollege

# Table of Contents

1. Oefening: Garage .....	1
1.1. Klasse Garage (domein).....	1
2. Oefening Clock .....	6
3. Oefening Movie.....	6
3.1. Gegeven Movie - Comparison<T> delegate - Lambda Expressie.....	7
4. Oefening 2 .....	7
4.1. Gegeven Movie - Named method .....	7

# 1. Oefening: Garage

Gegeven: Collections\_WC\_Garage\_StartBestanden.zip



Bekijk even de gegeven bestanden. Implementeer het domein en voer daarna in de domeinlaag de nodige stappen uit.

## 1.1. Klasse Garage (domein)

In de methode `InitGarage()` staan een aantal stappen beschreven die geïmplementeerd dienen te worden.

### 1.1.1. Stap 1

Haal de lijst van autos op uit de persistentielaaag.

**STAP 1 - Originele lijst met dubbels**

```
nummerplaat= 123xyz, merk= Toyota, model = Yaris
nummerplaat= 123xyz, merk= Toyota, model = Yaris
nummerplaat= 567xyz, merk= Renault, model = Fluence
nummerplaat= 456abc, merk= Opel, model = Astra
nummerplaat= azerty, merk= BMW, model = Berline
nummerplaat= qwerty, merk= Toyota, model = Avensis
nummerplaat= 789cde, merk= Mercedes, model = C-klasse Berline
nummerplaat= azerty, merk= BMW, model = Berline
nummerplaat= 567xyz, merk= Renault, model = Fluence
```

```
nummerplaat= ab12ab, merk= Opel, model = Zafira  
nummerplaat= xy12xy, merk= Peugeot, model = 308
```

Zet deze lijst om naar een set en merk op dat de dubbels verdwenen zijn.

STAP 1 - Set van autos

```
nummerplaat= 123xyz, merk= Toyota, model = Yaris  
nummerplaat= 567xyz, merk= Renault, model = Fluence  
nummerplaat= 456abc, merk= Opel, model = Astra  
nummerplaat= azerty, merk= BMW, model = Berline  
nummerplaat= qwerty, merk= Toyota, model = Avensis  
nummerplaat= 789cde, merk= Mercedes, model = C-klasse Berline  
nummerplaat= ab12ab, merk= Opel, model = Zafira  
nummerplaat= xy12xy, merk= Peugeot, model = 308
```

Verwijder nu uit de lijst zelf ook alle dubbels via een LINQ operatie.

STAP 1 - Lijst nu zonder dubbels

```
nummerplaat= 123xyz, merk= Toyota, model = Yaris  
nummerplaat= 567xyz, merk= Renault, model = Fluence  
nummerplaat= 456abc, merk= Opel, model = Astra  
nummerplaat= azerty, merk= BMW, model = Berline  
nummerplaat= qwerty, merk= Toyota, model = Avensis  
nummerplaat= 789cde, merk= Mercedes, model = C-klasse Berline  
nummerplaat= ab12ab, merk= Opel, model = Zafira  
nummerplaat= xy12xy, merk= Peugeot, model = 308
```

## 1.1.2. Stap 2

Zet de lijst van autos om naar een Dictionary. De sleutel is de nummerplaat en de waarde is de overeenkomstige auto. De dictionary wordt als attribuut bewaard voor later.

STAP 2 - Dictionary Nummerplaat -> Auto

```
123xyz -> nummerplaat= 123xyz, merk= Toyota, model = Yaris  
567xyz -> nummerplaat= 567xyz, merk= Renault, model = Fluence  
456abc -> nummerplaat= 456abc, merk= Opel, model = Astra  
azerty -> nummerplaat= azerty, merk= BMW, model = Berline  
qwerty -> nummerplaat= qwerty, merk= Toyota, model = Avensis  
789cde -> nummerplaat= 789cde, merk= Mercedes, model = C-klasse Berline  
ab12ab -> nummerplaat= ab12ab, merk= Opel, model = Zafira  
xy12xy -> nummerplaat= xy12xy, merk= Peugeot, model = 308
```

## 1.1.3. Stap 3

Een List<Onderhoud> aan de persistentie opvragen.

### STAP 3 : ONDERHOUDlijst

nummerplaat xy12xy van 12/12/2019 t.e.m. 15/12/2019  
nummerplaat ab12ab van 11/01/2020 t.e.m. 11/01/2020  
nummerplaat ab12ab van 15/12/2020 t.e.m. 15/12/2020  
nummerplaat 789cde van 11/01/2020 t.e.m. 12/01/2020  
nummerplaat xy12xy van 16/11/2019 t.e.m. 16/11/2019  
nummerplaat ab12ab van 15/01/2020 t.e.m. 15/01/2020  
nummerplaat 456abc van 17/12/2019 t.e.m. 19/12/2019  
nummerplaat 789cde van 10/01/2020 t.e.m. 10/01/2020  
nummerplaat xy12xy van 8/12/2019 t.e.m. 8/12/2019  
nummerplaat ab12ab van 17/01/2020 t.e.m. 17/01/2020  
nummerplaat 123xyz van 10/12/2019 t.e.m. 11/12/2019  
nummerplaat 123xyz van 15/12/2019 t.e.m. 16/12/2019  
nummerplaat azerty van 15/12/2019 t.e.m. 16/12/2019  
nummerplaat 567xyz van 15/02/2020 t.e.m. 15/02/2020  
nummerplaat 789cde van 5/01/2020 t.e.m. 7/01/2020  
nummerplaat 123xyz van 17/12/2019 t.e.m. 19/12/2019  
nummerplaat azerty van 18/12/2019 t.e.m. 18/12/2019  
nummerplaat azerty van 17/12/2019 t.e.m. 17/12/2019  
nummerplaat azerty van 20/12/2019 t.e.m. 22/12/2019

### 1.1.4. Stap 4

Sorteer de lijst van Onderhoud in stijgende volgorde volgens nummerplaat. Bij gelijke nummerplaten, volgens begindatum (stijgende volgorde).

### STAP 4

nummerplaat 123xyz van 10/12/2019 t.e.m. 11/12/2019  
nummerplaat 123xyz van 15/12/2019 t.e.m. 16/12/2019  
nummerplaat 123xyz van 17/12/2019 t.e.m. 19/12/2019  
nummerplaat 456abc van 17/12/2019 t.e.m. 19/12/2019  
nummerplaat 567xyz van 15/02/2020 t.e.m. 15/02/2020  
nummerplaat 789cde van 5/01/2020 t.e.m. 7/01/2020  
nummerplaat 789cde van 10/01/2020 t.e.m. 10/01/2020  
nummerplaat 789cde van 11/01/2020 t.e.m. 12/01/2020  
nummerplaat ab12ab van 11/01/2020 t.e.m. 11/01/2020  
nummerplaat ab12ab van 15/01/2020 t.e.m. 15/01/2020  
nummerplaat ab12ab van 17/01/2020 t.e.m. 17/01/2020  
nummerplaat ab12ab van 15/12/2020 t.e.m. 15/12/2020  
nummerplaat azerty van 15/12/2019 t.e.m. 16/12/2019  
nummerplaat azerty van 17/12/2019 t.e.m. 17/12/2019  
nummerplaat azerty van 18/12/2019 t.e.m. 18/12/2019  
nummerplaat azerty van 20/12/2019 t.e.m. 22/12/2019  
nummerplaat xy12xy van 16/11/2019 t.e.m. 16/11/2019  
nummerplaat xy12xy van 8/12/2019 t.e.m. 8/12/2019  
nummerplaat xy12xy van 12/12/2019 t.e.m. 15/12/2019

## 1.1.5. Stap 5 (optioneel)

Per auto de aangrenzende perioden samenvoegen. Sommige onderhoudsbeurten zijn aangrenzend qua datum. Vervang aangrenzende onderhoudsbeurten door één onderhoud.

## 1.1.6. Stap 6

De lijst van Onderhoud omzetten naar een Dictionary. De sleutel is de nummerplaat en de waarde is een lijst van Onderhoud (List<Onderhoud>). De map wordt als attribuut bijgehouden voor later.

### STAP 6

123xyz

nummerplaat 123xyz van 10/12/2019 t.e.m. 11/12/2019

nummerplaat 123xyz van 15/12/2019 t.e.m. 16/12/2019

nummerplaat 123xyz van 17/12/2019 t.e.m. 19/12/2019

456abc

nummerplaat 456abc van 17/12/2019 t.e.m. 19/12/2019

567xyz

nummerplaat 567xyz van 15/02/2020 t.e.m. 15/02/2020

789cde

nummerplaat 789cde van 5/01/2020 t.e.m. 7/01/2020

nummerplaat 789cde van 10/01/2020 t.e.m. 10/01/2020

nummerplaat 789cde van 11/01/2020 t.e.m. 12/01/2020

ab12ab

nummerplaat ab12ab van 11/01/2020 t.e.m. 11/01/2020

nummerplaat ab12ab van 15/01/2020 t.e.m. 15/01/2020

nummerplaat ab12ab van 17/01/2020 t.e.m. 17/01/2020

nummerplaat ab12ab van 15/12/2020 t.e.m. 15/12/2020

azerty

nummerplaat azerty van 15/12/2019 t.e.m. 16/12/2019

nummerplaat azerty van 17/12/2019 t.e.m. 17/12/2019

nummerplaat azerty van 18/12/2019 t.e.m. 18/12/2019

nummerplaat azerty van 20/12/2019 t.e.m. 22/12/2019

xy12xy

nummerplaat xy12xy van 16/11/2019 t.e.m. 16/11/2019

nummerplaat xy12xy van 8/12/2019 t.e.m. 8/12/2019

nummerplaat xy12xy van 12/12/2019 t.e.m. 15/12/2019

autoDictionary

nummerplaat= 123xyz, merk= Toyota, model = Yaris

nummerplaat= 456abc, merk= Opel, model = Astra

nummerplaat= 567xyz, merk= Renault, model = Fluence

nummerplaat= 789cde, merk= Mercedes, model = C-klasse Berline

nummerplaat= ab12ab, merk= Opel, model = Zafira

```
nummerplaat= azerty, merk= BMW, model = Berline  
nummerplaat= qwerty, merk= Toyota, model = Avensis  
nummerplaat= xy12xy, merk= Peugeot, model = 308
```

### 1.1.7. Extra converteer methodes

In de klasse Garage voorzie je ook ConverteerToString methoden:

1) `public string ConvertAutoDictionaryToString()`

De dictionary met autos wordt als één string teruggegeven (zie uitvoer). De nummerplaten zijn in stijgende volgorde.

```
autoDictionary  
nummerplaat= 123xyz, merk= Toyota, model = Yaris  
nummerplaat= 456abc, merk= Opel, model = Astra  
nummerplaat= 567xyz, merk= Renault, model = Fluence  
nummerplaat= 789cde, merk= Mercedes, model = C-klasse Berline  
nummerplaat= ab12ab, merk= Opel, model = Zafira  
nummerplaat= azerty, merk= BMW, model = Berline  
nummerplaat= qwerty, merk= Toyota, model = Avensis  
nummerplaat= xy12xy, merk= Peugeot, model = 308
```

2) `public String autoOnderhoudMap_toStringprivate string ConvertOnderhoudsLijstToString(List<Onderhoud> l)` De dictionary van onderhoudslijsten wordt als één string teruggegeven (zie uitvoer). De nummerplaten zijn in stijgende volgorde.

```
autoOnderhoudDictionary  
123xyz:  
nummerplaat 123xyz van 10/12/2019 t.e.m. 11/12/2019  
nummerplaat 123xyz van 15/12/2019 t.e.m. 16/12/2019  
nummerplaat 123xyz van 17/12/2019 t.e.m. 19/12/2019
```

```
456abc:  
nummerplaat 456abc van 17/12/2019 t.e.m. 19/12/2019
```

```
567xyz:  
nummerplaat 567xyz van 15/02/2020 t.e.m. 15/02/2020
```

```
789cde:  
nummerplaat 789cde van 5/01/2020 t.e.m. 7/01/2020  
nummerplaat 789cde van 10/01/2020 t.e.m. 10/01/2020  
nummerplaat 789cde van 11/01/2020 t.e.m. 12/01/2020
```

```
ab12ab:  
nummerplaat ab12ab van 11/01/2020 t.e.m. 11/01/2020  
nummerplaat ab12ab van 15/01/2020 t.e.m. 15/01/2020  
nummerplaat ab12ab van 17/01/2020 t.e.m. 17/01/2020
```

nummerplaat ab12ab van **15/12/2020** t.e.m. **15/12/2020**

azerty:

nummerplaat azerty van **15/12/2019** t.e.m. **16/12/2019**

nummerplaat azerty van **17/12/2019** t.e.m. **17/12/2019**

nummerplaat azerty van **18/12/2019** t.e.m. **18/12/2019**

nummerplaat azerty van **20/12/2019** t.e.m. **22/12/2019**

xy12xy:

nummerplaat xy12xy van **16/11/2019** t.e.m. **16/11/2019**

nummerplaat xy12xy van **8/12/2019** t.e.m. **8/12/2019**

nummerplaat xy12xy van **12/12/2019** t.e.m. **15/12/2019**

## 2. Oefening Clock

### Gegeven

De klasse **Clock.cs**

### Gevraagd

Verwerk en herwerk deze **Clock** klasse in een drielagenmodel:

- Vanuit de Cui kunnen zich een aantal observers aanmelden, die geïnteresseerd zijn in het verstrijken van een seconde.
  - Registreer een UI klasse **DisplayClock** (als implementatie schrijf je bij elke gebeurtenis iets weg op het scherm)
  - Registreer een UI klasse **LogCurrentTime** (als implementatie schrijf je bij elke gebeurtenis iets weg op het scherm)
  - Registreer een lambda expressie (als implementatie schrijf je bij elke gebeurtenis iets weg op het scherm)

Uitvoer voorbeeld:

```
DisplayClock - Current Time: 10:34:16
Logging to file: 10:34:16
Lambda expressie - Current Time: 10:34:16
DisplayClock - Current Time: 10:34:17
Logging to file: 10:34:17
Lambda expressie - Current Time: 10:34:17
```

## 3. Oefening Movie

## 3.1. Gegeven Movie - Comparison<T> delegate - Lambda Expressie

Werk verder op de oefening uit het werkcollege rond polymorfisme en interfaces waarbij een **IComparer** interface werd geïmplementeerd.

De **Sort** methode van een lijst is ook overloaded met een delegate **Comparison<T>**.

Implementeer het gedrag uit de klasse **RatingCompare** 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 **IComparable**. 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 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
```

## 4. Oefening 2

### 4.1. Gegeven Movie - Named method

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**