

## P1W3 Reflections - Opgave 2 RentCar

De bedoeling van deze opgave is reflection toepassen op de klasse **RentCar**.

Let op! Je mag de klasse **RentCar** nooit rechtstreeks gebruiken,  
bv niet `RentCar car = new RentCar(500);`

Je hoeft alleen de klasse **TestReflection** aan te vullen!

1. Druk de volledige naam van de package af.
2. Druk de 'header' van de constructor die de parameter `int` heeft af.
3. Druk de namen van alle methoden (ook die van de superklasse) af.
4. Druk de namen van alle attributen van de klasse **RentCar** af.
5. Maak een **Constructor** object (zonder parameter) en maak via dit object een nieuw object met de naam **car**. Haal nu de methode **computeRentalCost** op en voer ze uit (maak gebruik van `invoke`) met als parameterwaarde 4.
6. Haal het attribute **type** op en maak het toegankelijk voor reflection. Wijzig vervolgens de waarde van het type in het **car** object naar "extra large" en druk de nieuwe waarde af ter controle.

Als alles correct is zou je de volgende uitvoer moeten bekomen:

```
Package naam: package oefening.reflection
Constructor: public oefening.reflection.RentCar(int)
Methoden: getType setRate setType getRate computeRentalCost wait wait wait
equals toString hashCode getClass notify notifyAll
Attributen: rate type price
```

```
The cost of your rental car is €180
Waarde van type: extra large
```

### Extra

Vervang puntje 3 door het volgende: Druk alleen de namen van de methoden van de klasse zelf af.

Vervang puntje 5 door het volgende: Maak via een **Constructor** object met de **int** als parameter een nieuw object van de klasse **RentCar** en geef als waarde 500 mee.

De uitvoer zou nu het volgende moeten zijn:

```
Constructor: public oefening.reflection.RentCar(int)
Methoden: getType computeRentalCost getRate setRate setType
Attributen: rate type price
```

```
The cost of your rental car is €220
Waarde van type: extra large
```