

# Practicum Overerving

## VAA

Naast overerving is polymorfisme een belangrijk onderdeel van een OO taal. Overerving laat ons toe functionaliteit van een klasse (de parent klasse) te hergebruiken en de nodige zaken toe te voegen of aan te passen in de child klasse. Polymorfisme (= letterlijke vertaling: meerdere vormen) laat ons toe om deze gespecialiseerde functionaliteiten in de child klasse aan te spreken via het parent object. Voorbeeld (zie les): Een zoogdier kan onder meerder vormen bestaan, namelijk: een hond / een vogelbekdier / een paard / ... . We kunnen objecten van al deze vormen onderbrengen in een lijst: `List<Zoogdier> zoogdieren = new List<Zoogdier>();` We kunnen nu de functie `KrijgKind` op al deze objecten oproepen. Wanneer de functie `KrijgKind` correct is ge “override” zal de functie van de child klasse worden uitgevoerd (en niet de functie `KrijgKind` in de parent klasse `Zoogdier`).

## Oefening 1

Maak een applicatie die het mogelijk maakt voor een bedrijf om het totaal VAA te berekenen van zijn wagenpark. In het wagenpark kunnen zowel diesel als benzine wagens aanwezig zijn. Elke wagen heeft steeds de volgende eigenschappen:

### Wagen

- Properties
  - Catalogus prijs (double): property die de catalogus prijs van de wagen bevat. Deze waarde mag niet negatief zijn (0 indien een negatieve waarde wordt ingegeven).
  - Nummerplaat (string): property die de nummerplaat van de wagen bevat
- Functies
  - ToString
    - Omschrijving: deze functie zal de omschrijving van de wagen teruggeven = de waarde van de 2 bovenstaande properties. Deze functie is een override van de ToString functie in de parent (Object)
    - Return
      - Type: string
      - Waarde: de Wagen omschrijving
    - Parameters: geen
  - Vaa
    - Omschrijving: geeft de VAA waarde (Voordelen Alle Aard) terug van de wagen. In deze klasse kan de correcte waarde nog niet worden berekend. Dit is enkel mogelijk in zijn child klassen. De functie zal dus enkel 0 teruggeven.
    - Return
      - Type: double
      - Waarde: VAA van de wagen
    - Parameters: geen

Een benzine en dieselwagen zijn beiden wagens waaraan specifieke kenmerken kunnen worden toegewezen:

### DieselWagen

- Properties
  - NOx (double): property die de NOx uitstoot waarde van de dieselwagen bevat.
- Functies
  - ToString
    - Omschrijving: deze functie zal de omschrijving van de dieselwagen teruggeven = de waarde van de 2 Wagen properties + NOx. Deze functie is een override van de ToString functie in de parent (Wagen)
    - Return
      - Type: string
      - Waarde: de DieselWagen omschrijving
    - Parameters: geen
  - Vaa
    - Omschrijving: geeft de VAA waarde (Voordelen Alle Aard) terug van de dieselwagen. De waarde zal als volgt worden berekend:  
$$\text{VAA} = \text{Cataloguswaarde} \times (\text{NOx waarde}) / 1000$$
    - Return

- Type: double
- Waarde: VAA van de dieselwagen
- Parameters: geen

## BenzineWagen

- Properties
  - CO2 (double): property die de CO2 uitstoot waarde van de benzinewagen bevat.
- Functies
  - ToString
    - Omschrijving: deze functie zal de omschrijving van de benzinewagen teruggeven = de waarde van de 2 Wagen properties + CO2. Deze functie is een override van de ToString functie in de parent (Wagen)
    - Return
      - Type: string
      - Waarde: de BenzineWagen omschrijving
    - Parameters: geen
  - Vaa
    - Omschrijving: geeft de VAA waarde (Voordelen Alle Aard) terug van de benzinewagen. De waarde zal als volgt worden berekend:  

$$\text{VAA} = \text{Cataloguswaarde} \times (\text{CO2 waarde}) / 1000$$
    - Return
      - Type: double
      - Waarde: VAA van de benzinewagen
    - Parameters: geen

Programme vereisten:

- Het moet mogelijk zijn om een ongelimiteerd aantal wagens aan de applicatie toe te voegen (lees: maak een List aan),
- Het moet mogelijk zijn een lijst op te vragen van alle toegevoegde wagens
- Het moet mogelijk zijn om het totaal aantal VAA op te vragen.

Controleer steeds dat de user input correct is (en vang numerieke fouten correct op dmv de TryParse functies)

Hieronder een voorbeeld van de mogelijke scherm / programma flow:

Menu:

```

** VAA berekening **
1. Voeg een wagen toe.
2. Overzicht wagens
3. Totaal VAA
4. Stop programma
>

```

1. Voeg een wagen toe

```
** VAA berekening **
1. Voeg een wagen toe.
2. Overzicht wagens
3. Totaal VAA
4. Stop programma
> 1
    Type
    1. Diesel
    2. Benzine
    > 1
    Nummerplaat: 1-DFS-234
    Catalogusprijs: 25.000
    NOx: 80
Druk op een toets om verder te gaan.
```

```
** VAA berekening **
1. Voeg een wagen toe.
2. Overzicht wagens
3. Totaal VAA
4. Stop programma
> 1
    Type
    1. Diesel
    2. Benzine
    > 2
    Nummerplaat: 1-OPI-654
    Catalogusprijs: 30.333
    CO2: 104
Druk op een toets om verder te gaan.
```

## 2. Overzicht wagens

```
** VAA berekening **
1. Voeg een wagen toe.
2. Overzicht wagens
3. Totaal VAA
4. Stop programma
> 2
Nummerplaat: 1-DFS-234 / Catalogusprijs: 25000 / NOx: 80
Nummerplaat: 1-OPI-654 / Catalogusprijs: 30333 / CO2: 104
Druk op een toets om verder te gaan.
```

## 3. Totaal VAA

C:\Program Files\dothnet\dothnet.exe

```
** VAA berekening **  
1. Voeg een wagen toe.  
2. Overzicht wagens  
3. Totaal VAA  
4. Stop programma  
> 3  
Het totaal te betalen VAA: 5154,632  
Druk op een toets om verder te gaan.
```