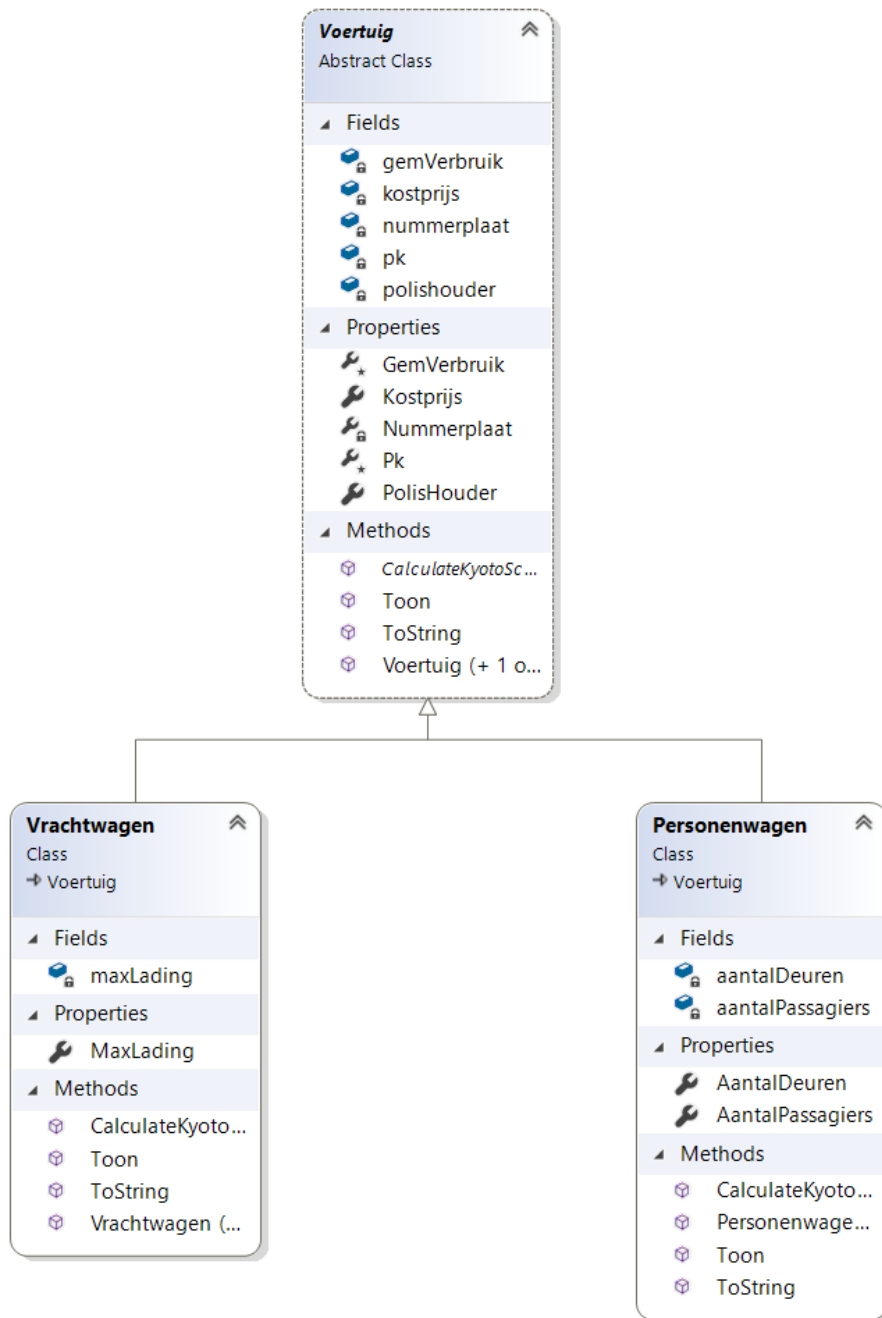


OEFENING ABSTRACT CLASS - VOERTUIG

Maak onderstaande klassen aan en let goed op wat er telkens gevraagd wordt



A. De class Voertuig

Variabelen en Properties:

Maak een class Voertuig met:

- een property *Polishouder* van het type String
- een property *Kostprijs* van het type float
- een property *Pk* van het type int
- een property *GemVerbruik* van het type float
- een property *Nummerplaat* van het type String

Extra info:

De default polishouder en nummerplaat zijn “onbepaald” en de kostprijs staat op 0, evenals het aantal pk en het gemiddeld verbruik.

Constructors:

Maak een constructor zonder parameters.

Maak ook een constructor die de polishouder, de kostprijs, het aantal pk, het gemiddeld verbruik en een nummerplaat aanvaardt.

ToString:

Return alle gegevens in één String in een ToString() method. Eerst de polishouder dan de kostprijs, het aantal pk, het verbruik en een nummerplaat. Voorzie ook een method Toon(), die uiteraard alle gegevens toont op het scherm.

B. De class Vrachtwagen

Variabelen en Properties:

Maak een class Vrachtwagen, die is afgeleid van Voertuig en Properties heeft voor:

- *MaxLading* van het type float

Extra info:

De default maxLading is 10000 kg en mag niet negatief zijn. Als je toch een negatieve waarde meegeeft krijgt de vrachtwagen de default waarde mee.

Constructors:

Maak een constructor zonder parameters.

Voorzie een constructor waarbij je alle waardes via parameters meegeeft.

ToString:

Zorg ervoor dat de gegevens van de Vrachtwagen kunnen getoond worden op het scherm en override de ToString()-method.

C. De class Personenwagen

Variabelen en Properties:

Maak een class Personenwagen, die is afgeleid van Voertuig en Properties heeft voor:

- *AantalDeuren* van het type int

- *AantalPassagiers* van het type int

Extra info:

Het default aantal deuren is 4, het default aantal passagiers is 5. Deze properties mogen niet negatief zijn, als dit toch gebeurt geef dan gewoon de default waarde mee.

Constructors:

Maak een constructor zonder parameters.

Voorzie een constructor waarbij je alle waardes via parameters meegeeft.

ToString:

Zorg ervoor dat de gegevens van de personenwagen kunnen getoond worden op het scherm en override de ToString()-method.

D. De abstracte method CalculateKyotoScore ()

Maak de class Voertuig **abstract** en voeg de volgende abstracte method toe :

public double CalculateKyotoScore()

Deze method returnt een Kyoto-score. Voor een personenwagen is de Kyoto-score gelijk aan het verbruik vermenigvuldigd met het aantal pk, gedeeld door het aantal passagiers.

Voor een vrachtwagen is de Kyoto-score gelijk aan het verbruik vermenigvuldigd met het aantal pk, gedeeld door de lading in ton.

Werk CalculateKyotoScore() uit in de diverse classes.

E. Program

Gebruik volgende Main() code om alles uit te testen

```
class Program
{
    static void Main()
    {
        Voertuig opel1 = new Personenwagen();
        opel1.Toon();
        Console.WriteLine(opel1);
        Vrachtwagen volvo1 = new Vrachtwagen();
        volvo1.Toon();
        Console.WriteLine(volvo1);
        Personenwagen opel2 = new Personenwagen("Jan Klaasen", 14599.0F, 105,
6.8F, "KLM099", 5, 5);
        opel2.Toon();
        Console.WriteLine(opel2);

        opel2.Kostprijs = -15000;
        opel2.AantalDeuren = -7;
        opel2.AantalPassagiers = 0;
        Console.WriteLine(opel2);
        //opel2 is niet gewijzigd
        Vrachtwagen volvo2 = new Vrachtwagen("Michel Dewolf", 214599.0F, 440,
33.1F, "PRD441", 6000.0F);
        volvo2.Toon();
        Console.WriteLine(volvo2);
        Console.WriteLine();
        Console.WriteLine("Kyotoscore personenwagen 1: " +
        opel1.CalculateKyotoScore());
        Console.WriteLine("Kyotoscore personenwagen 2: " +
        opel2.CalculateKyotoScore());

        Console.WriteLine("Kyotoscore vrachtwagen 1: " +
        volvo1.CalculateKyotoScore());
        Console.WriteLine("Kyotoscore vrachtwagen 2: " +
        volvo2.CalculateKyotoScore());
        // Voertuig v = new Voertuig(); //Error message cannot create an instance
        of the abstract class or interface

        Console.ReadKey();
    }
}
```

```
C:\DATA\SYNTRA\2deREEKS\2deREEKS\LES...
PERSONENWAGEN
polishouder: onbepaald
kostprijs: 0
pk: 0
gemVerbruik: 0
nummerplaat: onbepaald
deuren: 4
passagiers: 5
onbepaald;0;0;0;onbepaald;4;5

VRACHTWAGEN
polishouder: onbepaald
kostprijs: 0
pk: 0
gemVerbruik: 0
nummerplaat: onbepaald
max. lading: 10000
onbepaald;0;0;0;onbepaald;10000

PERSONENWAGEN
polishouder: Jan Klaasen
kostprijs: 14599
pk: 105
gemVerbruik: 6.8
nummerplaat: KLM099
deuren: 5
passagiers: 5
Jan Klaasen;14599;105;6.8;KLM099;5;5
Jan Klaasen;-15000;105;6.8;KLM099;-7;0

VRACHTWAGEN
polishouder: Michel Dewolf
kostprijs: 214599
pk: 440
gemVerbruik: 33.1
nummerplaat: PRD441
max. lading: 6000
Michel Dewolf;214599;440;33.1;PRD441;6000

Kyotoscore personenwagen 1: 0
Kyotoscore personenwagen 2: ∞
Kyotoscore vrachtwagen 1: 0
Kyotoscore vrachtwagen 2: 2427.33322143555
```