**Leerplandoelstellingen**:

***6 Efficiënt software ontwikkelen***

***6.2 Een goede programmeerattitude hanteren***

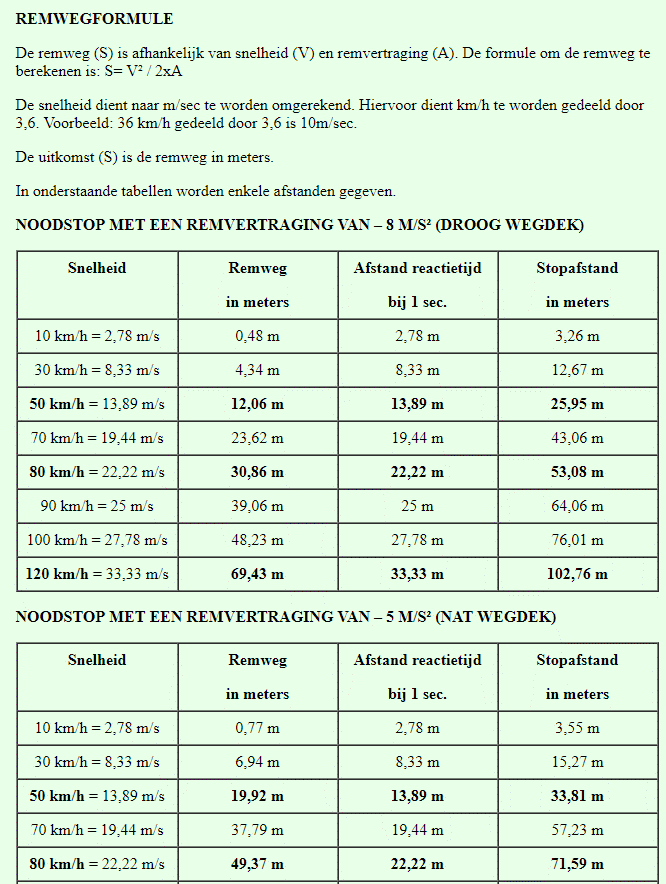
*6.2.3 Een consequente naamgeving en schikking hanteren bij het maken van programma’s.*

*6.2.5 Een programma steeds van de nodige commentaar voorzien om de leesbaarheid van de code te verhogen.*

***6.3 Technieken i.v.m. softwareontwikkeling op een efficiënte wijze toepassen***

*6.3.1 De basiselementen van het gestructureerd object georiënteerd programmeren kennen en toepassen, onder meer:(object)variabele, constante, gegevenstypes, scope; programmeerstructuren; arrays, lijsten; ingebouwde methodes en eigenschappen; zelf gedefinieerde functies, procedures, argumenten; klasse, object, eigenschap, gebeurtenis, methode, instantie.*

Op de volgende site vinden we een formule voor het berekenen van de stopafstand van een auto: <http://members.home.nl/johngrimbergen/remwegformule.htm>. Daarop lezen we:



We willen nu een toepassing maken die het berekenen van de stopafstand automatiseert.

# Werkwijze

Analyse

* Probleemstelling begrijpen
* Bedenken van de nodige klassen en bestanden
* Bedenken van testen

Uitvoering

* Code schrijven en regelmatig testen

# Analyse:

## Probleemstelling begrijpen

Uit bovenstaande informatie kunnen we heel wat afleiden:

De totale **stop**afstand bestaat uit twee grote delen **welke 2 delen** zijn dat?

* snelheid
* reactietijd

Welke factoren bepalen de **rem**afstand? Schrijf de naam op van die factoren.

* snelheid
* het weer

Schrijf een formule neer om de remafstand te bepalen rekening houdend met de naam van die factoren.

Afstand = v^2 / 2\*afstand

In welke **eenheid** wordt de **snelheid** uitgedrukt bij het berekenen van de **rem**afstand?

m/s

Hoe kan je een snelheid van km/u omzetten naar een snelheid in m/s? Geef een concreet voorbeeld.

Delen door 3.6

Wat wordt er bedoeld met ‘Afstand reactietijd’ (in meter) ?

De remweg + de afstand na de reactietijd

Hoe bereken je die ‘Afstand reactietijd’ (in meter)?

Dat is de tijd in s \* de snelheid in m/s

Geef een concreet voorbeeld voor een auto die 50km/u rijdt.

50/3.6 = 13.89 m na 1 seconde

Wat wordt er bedoeld met de remvertraging?

De afstand die je remt per m

Welke mogelijke gevallen zijn er?

Droog weer of nat weer

## Bedenken van de nodige klassen en bestanden

Welke **attributen** en **methoden** zal zo’n object auto moeten hebben in ons model? Welke **parameters** hebben die methoden nodig? M.a.w. welke zaken willen we van zo’n auto-object berekenen? Denk hierbij aan de zaken die we met ons software model willen bereiken. Hou er ook rekening mee dat de **auto géén negatieve snelheid toegewezen mag krijgen**.

Schets nu hieronder een klassendiagram voor de **klasse Auto** met de attributen en de methoden.

|  |
| --- |
| Auto |
| Snelheid  Reactietijd  remvertraging |
| \_\_init\_\_()  Remweg() |

We willen **deze klasse nu gebruiken in een functie in een andere module** ‘optijdremmen.py’

We willen ook gebruik maken van vensters om de nodige informatie op te vragen.

Van welke klassen uit welke library willen we dan gebruik maken om met vensters te werken?

Simpledialogs uit simpleutils

Met welke instructie maak je die vensters dan ter beschikking in de module ‘optijdremmen.py’

From simpleutils.simpledialogs import SimpleDialog

We willen ook gebruik maken van de klasse ‘Auto’ uit de module ‘auto.py’

Hoeveel import instructies heb je dus nodig?

From mijnPackageh4.auto import Auto

## Bedenken van testen

* Omschrijf **op welke manier** je de toepassing gaat **testen** om uit te zoeken of alles werkt naar behoren**.**
* **Welke objecten** ga je creëeren? Met welke beginwaarden?
* **Welke methoden** ga je van die objecten uitvoeren?

Noteer de beginwaarden van de attributen van elk object en daaronder de methoden die je op dat object wil uitvoeren en het resultaat van het uitvoeren van die methode.  
  
objauto1

|  |  |  |
| --- | --- | --- |
| attribuutnaam | Parameters | Waarden van het object na het creëren van het object |
| Snelheid | 0 | 0 |
| reactietijd | 0 | 0 |
| remvertraging | true | 5m/s^2 |

objauto2 met negatieve parameter voor de snelheid bij het creëeren

|  |  |  |
| --- | --- | --- |
| attribuutnaam | Parameters | Waarden van het object na het creëren van het object |
| Snelheid | -10 | 0 |
| reactietijd | -10 | 0 |
| remvertraging | true | 5m/s^2 |

objauto3 met positieve parameter voor de snelheid bij het creëeren

|  |  |  |
| --- | --- | --- |
| attribuutnaam | Parameters | Waarden van het object na het creëren van het object |
| Snelheid | 11m/s | 11m/s |
| reactietijd | 1 | 1 |
| remvertraging | false | 8m/s^2 |

Welke resultaat verwacht je na uitvoeren van:

objauto3.snelheid\_in\_m\_per\_s() 11m/s

objauto3.remafstand(True) 8m/s^2

objauto3.afstand\_reactietijd(1) 11m/s

objauto3.stopafstand(True,1)

# Uitvoering

Maak nu de toepassing.

Welke objecten ga je creëren? Kies een naam voor het object.

…

Welke andere informatie moet je invoeren?

…

…

Wat moet je berekenen?

…

…

…

Zorg voor een passende uitvoer?

Maak een klasse Auto met de nodige attributen en methoden in een module auto.py

We willen een auto met een **constructor met drie parameters**.

De standaardwaarde voor het merk en de nrplaat is de lege string “”, de standaard waarde voor de snelheid is 0.

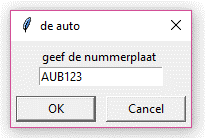
Hou ook rekening met het feit dat een auto geen negatieve snelheid toegewezen mag krijgen…

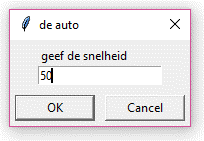
Maak een functie uitvoeren() in een module ‘optijdremmen.py’ waarin je

* een auto-object maakt – zonder parameters

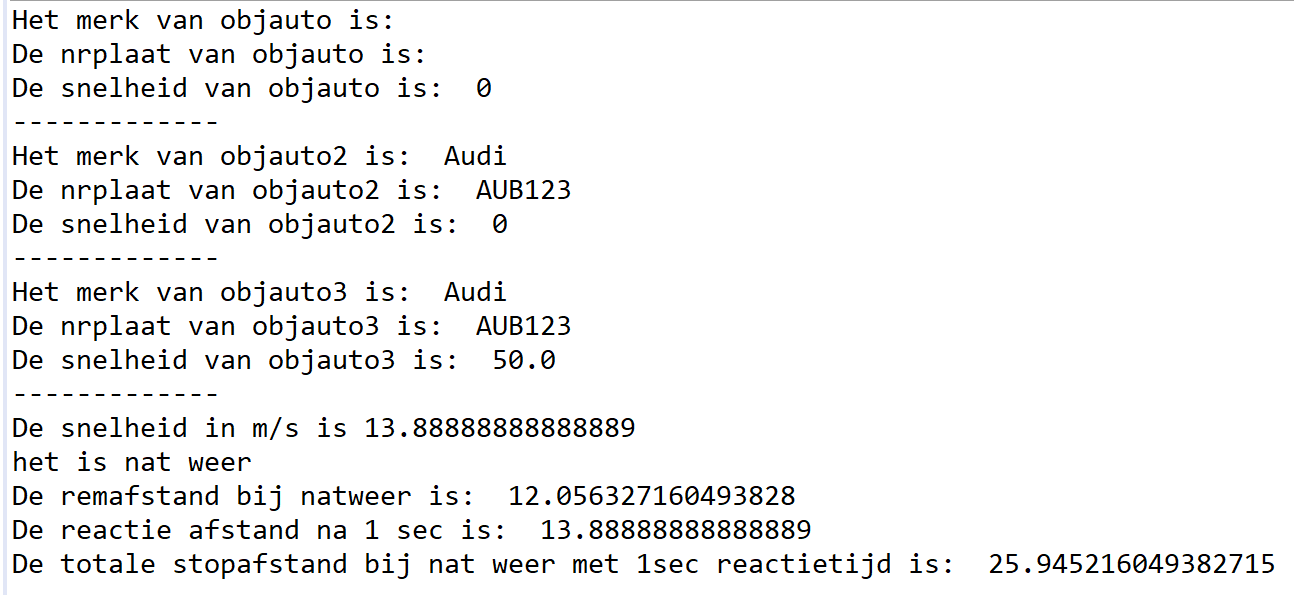
en vervolgens de waarden van de attributen toont in het console venster

* een auto-object maakt – met parameters én een negatieve snelheid   
  (in te voeren via de constructor) en vervolgens de waarden van de attributen toont in het console venster
* een auto-object maakt – met als parameters nrplaat= “AUB123”, merk=”Audi” en snelheid 50 (**in te voeren via vensters**)
  + en vervolgens de waarden van de attributen toont in het console venster
  + de remafstand berekent en toont in het console venster
  + de reactie afstand berekent en toont in het console venster
  + de totale stopafstand berekent en toont in het console venster





En een mogelijke uitvoer is dan:



De opdracht moet je uploaden op teams.