

# HO GENT

H3 Klasse en object -  
voorbeeldoefeningen

# Table of Contents

1. Doelstellingen .....	1
2. Oefeningen.....	1
2.1. Oefening 1.....	1
2.2. Oefening 2.....	1
2.3. Oefening 3.....	1
2.4. Oefening 4.....	2
2.4.1. Implementeer de domeinklasse Driehoek .....	2
2.4.2. Implementeer de domeinklasse Rechthoek .....	3
2.5. Oefening 5.....	3

# 1. Doelstellingen

- Oefening 1: Onderscheid klasse - object
- Oefening 2: Context bepaalt of woord een klasse of object is
- Oefening 3: Klasse tekenen in UML
- Oefening 4: Implementeren van een domeinklasse
- Oefening 5: Uitwerken van een kleine applicatie mét gebruik van een domeinklasse

## 2. Oefeningen

### 2.1. Oefening 1

Welke van onderstaande woorden zou je gebruiken als de naam van een klasse, welke woorden zou je gebruiken voor objecten?

Opgelet: om het antwoord niet te verklappen, is elk woord met een hoofdletter geschreven.

Hoe zat die regel met hoofdletters weer in elkaar?

Volvo	Beuk	Interpreter	Eik
Voertuig	Fiets	Heimwee	Mercedes
Verdriet	Automerk	Gevoel	Vliegtuig
Vreugde	Vrachtwagen	Liefde	Trein
Esdoorn	Editor	Autobus	Boom
Audi	Compiler	Personenwagen	Programmeergereedschap

### 2.2. Oefening 2

Enkel de naam zal niet voldoende zijn om te beslissen of we te maken hebben met een klasse of met een object. Neem bv. het woord ‘vis’ en beschrijf een situatie waarin dit zou kunnen duiden op een klasse en een situatie waarin dit zou kunnen duiden op een object.

### 2.3. Oefening 3

Ontwerp de klasse **Fiets** in UML

Zorg dat een fiets

- volgende **eigenschappen** heeft

kleur, merk, aantal versnellingen, herenfiets (ja/nee), snelheid

- volgende **methodes** heeft
  - een methode die toelaat de kleur van een fiets op te vragen.
  - een methode die toelaat het aantal versnellingen van een fiets te wijzigen.
  - een methode die toelaat dat een fiets kan versnellen tot een zekere snelheid, de snelheid wordt als extra informatie via de methode doorgegeven.
- volgende **constructor** heeft
  - een constructor die toelaat dat de je de kleur kan bepalen van een nieuw fiets object, deze kleur wordt als extra informatie doorgegeven aan de constructor.

## 2.4. Oefening 4

### 2.4.1. Implementeer de domeinklasse Driehoek

Driehoek
<<Property>> -a : int <<Property>> -b : int <<Property>> -c : int
+Driehoek(a : int, b : int, c : int) -setA(a : int) : void -setB(b : int) : void -setC(c : int) : void +isRechthoekigeDriehoek() : boolean

**Attributen** : a, b en c (stellen de zijden van een driehoek voor)

**Constructor** met 3 parameters

**Getters en setters** voor alle attributen (zijde moet >0, indien niet defaultwaarde 1).

Andere **methode** isRechthoekigeDriehoek: geeft true/false terug naargelang a, b en c de zijden van een rechthoekige driehoek kunnen voorstellen.



*Stelling van Pythagoras:* In een rechthoekige driehoek is de som van de kwadraten van de lengtes van de rechthoekszijden gelijk aan het kwadraat van de lengte van de schuine zijde.

Als we klaar zijn met het implementeren van deze domeinklasse, kunnen we die aan de hand van een gegeven testklasse testen.

1. Maak een package testen in je project.
2. Voeg de testklasse toe in deze package.
3. Voeg JUnit5 library toe aan je project (zie ook handleiding\_eclipse)
4. Klik rechts op je testklasse en kies voor Run As → JUnit Test

Package Explorer JUnit

Finished after 0,121 seconds

Runs: 9/9 Errors: 0 Failures: 0

▼ DriehoekTest [Runner: JUnit 5] (0,000 s)

- maakDriehoek\_zijdeCTeKlein\_creatieObjectDefaultVoorC() (0,000 s)
- maakDriehoek\_3normaleZijden\_creatieObject() (0,000 s)
- maakDriehoek\_zijdeBgrenswaardeNul\_creatieObjectDefaultVoorB() (0,000 s)
- maakDriehoek\_zijdeATeKlein\_creatieObjectDefaultVoorA() (0,000 s)
- isRechthoekigeDriehoek\_retourneertTrue() (0,000 s)
- isRechthoekigeDriehoek\_retourneertFalse() (0,000 s)
- maakDriehoek\_zijdeAgrenswaardeNul\_creatieObjectDefaultVoorA() (0,000 s)
- maakDriehoek\_zijdeBTeklein\_creatieObjectDefaultVoorB() (0,000 s)
- maakDriehoek\_zijdeCgrenswaardeNul\_creatieObjectDefaultVoorC() (0,000 s)

## 2.4.2. Implementeer de domeinklasse Rechthoek

Rechthoek
<<Property>> -lengte : double
<<Property>> -breedte : double
+Rechthoek(lengte : double, breedte : double)
+Rechthoek()
+berekenOmtrek() : double
+berekenOppervlakte() : double
-setLengte(lengte : double) : void
-setBreedte(breedte : double) : void

De klasse heeft als **attributen** lengte en breedte.

De **constructor** is zo gedefinieerd dat bij creatie van een nieuw Rechthoek-object de attribuutwaarden worden meegegeven. Er is ook een **default constructor** gedefinieerd. Die maakt een rechthoek aan waarvan de lengte op 10.0 wordt ingesteld en de breedte op 7.0.

Voorzie voor elk attribuut een **setter- en een getter-methode**. Lengte en breedte moeten strikt positief zijn. Anders wordens ze ingesteld op de defaultwaarde 1.0.

Verder beschikt de klasse over 2 **extra methodes** berekenOmtrek en berekenOppervlakte, die respectievelijk de omtrek en de oppervlakte van de rechthoek berekenen.

Ben je klaar met implementeren, vergeet dan niet te testen via de testklasse.

## 2.5. Oefening 5

Maak nu gebruik van de 2 domeinklassen uit oefening 4 om volgende applicatie uit te werken.

Allereerst vragen we aan de gebruiker of hij een vorm wil ingeven.

- Bij keuze 0: applicatie stopt met het uitschrijven van de nodig info.
- Bij keuze 1: applicatie vraagt om lengte en breedte van een rechthoek in te geven.
- Bij keuze 2: applicatie vraagt om 3 zijden van een driehoek in te geven.

Na het ingeven van een rechthoek of driehoek krijg je terug de kans om te stoppen of om een nieuwe vorm in te geven.

Wat je ook koos, op het eind wordt volgende informatie uitgeschreven:

Overzicht vormen:

Totaal aantal vormen:

Aantal rechthoeken met opp > 50:

Aantal rechthoekige driehoeken:

Hieronder zie je 3 keer een verschillende uitvoer. Respecteer dit.

#### Voorbeeld uitvoer 1

```
Rechthoeken en driehoeken
```

```
-----
```

```
Wil je graag nog een vorm ingeven (1=een rechthoek, 2=een driehoek, 0=nee)? 0
```

```
Overzicht vormen:
```

```
Totaal aantal vormen: 0
```

```
Aantal rechthoeken met opp > 50: 0
```

```
Aantal rechthoekige driehoeken: 0
```

#### Voorbeeld uitvoer 2

```
Rechthoeken en driehoeken
```

```
-----
```

```
Wil je graag nog een vorm ingeven (1=een rechthoek, 2=een driehoek, 0=nee)? 1
```

```
Geef de lengte van de rechthoek: 8
```

```
Geef de breedte van de rechthoek: 6,5
```

```
Wil je graag nog een vorm ingeven (1=een rechthoek, 2=een driehoek, 0=nee)? 1
```

```
Geef de lengte van de rechthoek: 2
```

```
Geef de breedte van de rechthoek: 0,5
```

```
Wil je graag nog een vorm ingeven (1=een rechthoek, 2=een driehoek, 0=nee)? 0
```

```
Overzicht vormen:
```

```
Totaal aantal vormen: 2
```

```
Aantal rechthoeken met opp > 50: 1
```

```
Aantal rechthoekige driehoeken: 0
```

#### Voorbeeld uitvoer 3

## Rechthoeken en driehoeken

```
-----
Wil je graag nog een vorm ingeven (1=een rechthoek, 2=een driehoek, 0=nee)? 1
Geef de lengte van de rechthoek: 2,5
Geef de breedte van de rechthoek: 2
Wil je graag nog een vorm ingeven (1=een rechthoek, 2=een driehoek, 0=nee)? 2
Geef de lengte van zijde A: 3
Geef de lengte van zijde B: 4
Geef de lengte van zijde C: 5
Wil je graag nog een vorm ingeven (1=een rechthoek, 2=een driehoek, 0=nee)? 0
Overzicht vormen:
Totaal aantal vormen: 2
Aantal rechthoeken met opp > 50: 0
Aantal rechthoekige driehoeken: 1
```



Om te weten of de oppervlakte van de rechthoek groter is dan 50 maak je gebruik van de voorziene methode uit de domeinklasse. Het bepalen of een driehoek rechthoekig is, hebben we ook al uitgewerkt in de domeinklasse. Communiceer dus met het object dat je aanmaakt om deze informatie te krijgen.



Probeer niet alles in 1 keer uit te werken. Doe het stap per stap. Vraag bv eerst gewoon de gegevens van een rechthoek op en schrijf de uitvoer uit. Als dat werkt kan je de gegevens van 1 rechthoek én 1 driehoek opvragen, daarna de keuze geven en op het eind de herhaling. Probeer telkens stukje per stukje te laten werken. Zo kom je snelst tot een werkende, goede oplossing.