

# HO GENT

H3 Klasse en object - Oefeningen

# Table of Contents

1. Doelstellingen .....	1
2. Oefeningen.....	1
2.1. Oefening - Onderscheid klasse - object .....	1
2.2. Oefening - Onderscheid klasse - object (2) .....	1
2.3. Oefening - Domeinklasse Fiets.....	1
2.4. Oefening - Domeinklasse Rekening .....	2
2.5. Oefening - Implementeer de domeinklasse Driehoek .....	2
2.6. Oefening - Implementeer de domeinklasse Rechthoek.....	4
2.7. Oefening - VormApplicatie .....	5
2.8. Oefening - Product .....	6
2.9. Oefening - Rekenmachine.....	7

# 1. Doelstellingen

- Inleidende oefeningen : Onderscheid klasse - object
- Oefeningen op het voorstellen van een domeinklasse in UML
- Oefening op het implementeren van een domeinklasse
- Oefeningen op het uitwerken van kleine applicaties mét gebruik van een domeinklasse

## 2. Oefeningen

### 2.1. Oefening - Onderscheid klasse - object

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 - Onderscheid klasse - object (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 - Domeinklasse Fiets

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 - Domeinklasse Rekening

Stel een klasse Rekening voor in UML.

**Eigenschappen** van een object van Rekening

- een rekeningnummer voorgesteld door een geheel getal (tot 15 cijfers lang)
- het saldo van de rekening
- naam van de houder van de rekening

**Constructoren**

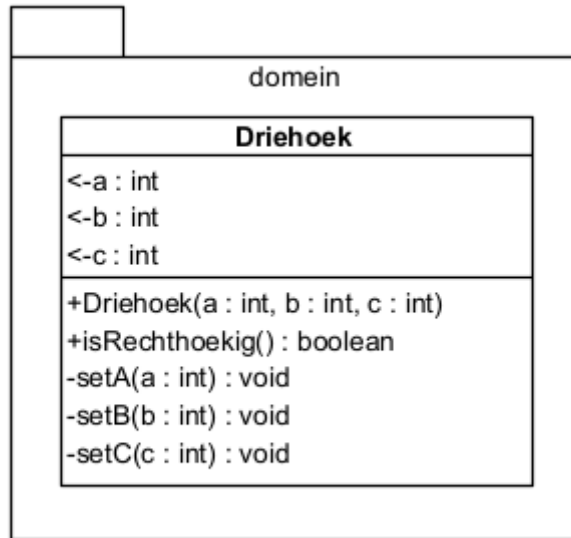
- Constructor met 2 parameters, één voor het rekeningnummer en één voor de naam van de houder.
- Constructor met 1 parameter: enkel het rekeningnummer wordt doorgegeven.
- Constructor zonder parameters

Bij creatie van een Rekening-object is het saldo altijd 0.0. Als het rekeningnummer niet meegegeven wordt, dan is de waarde 123456789, houder is “onbekend” als het niet meegegeven wordt.

**Methodes**

- De 3 eigenschappen hebben een getter.
- De naam van de houder van de rekening kan een nieuwe waarde krijgen via een setter, rekeningnummer ook (maar deze wijziging kan alleen maar van binnen in de klasse gebeuren, andere objecten kunnen dat niet doen). Voor saldo bestaat geen setter.
- Er is een methode om een bedrag te storten. De methode laat weten of het gelukt is of niet.
- Er is een methode om een bedrag af te halen. De methode laat weten of het gelukt is of niet.

## 2.5. Oefening - Implementeer de domeinklasse Driehoek



**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** isRechthoekig: 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

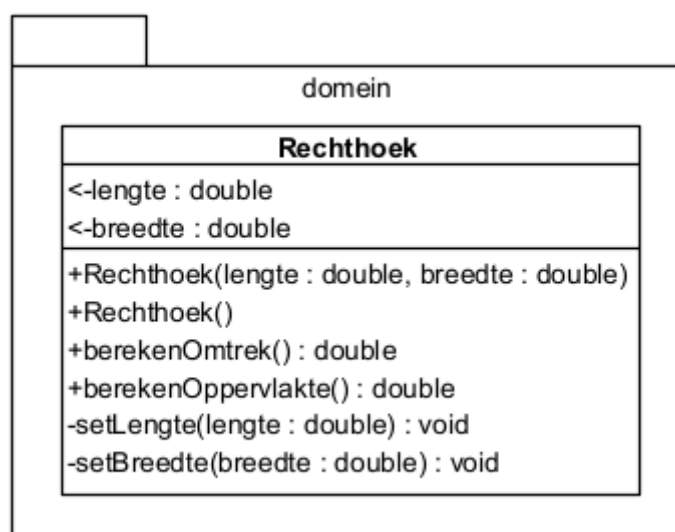
Finished after 0,121 seconds

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)

## 2.6. Oefening - Implementeer de domeinklasse Rechthoek



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.

## 2.7. Oefening - VormApplicatie

Maak nu gebruik van de 2 domeinklassen uit voorgaande oefeningen 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
```



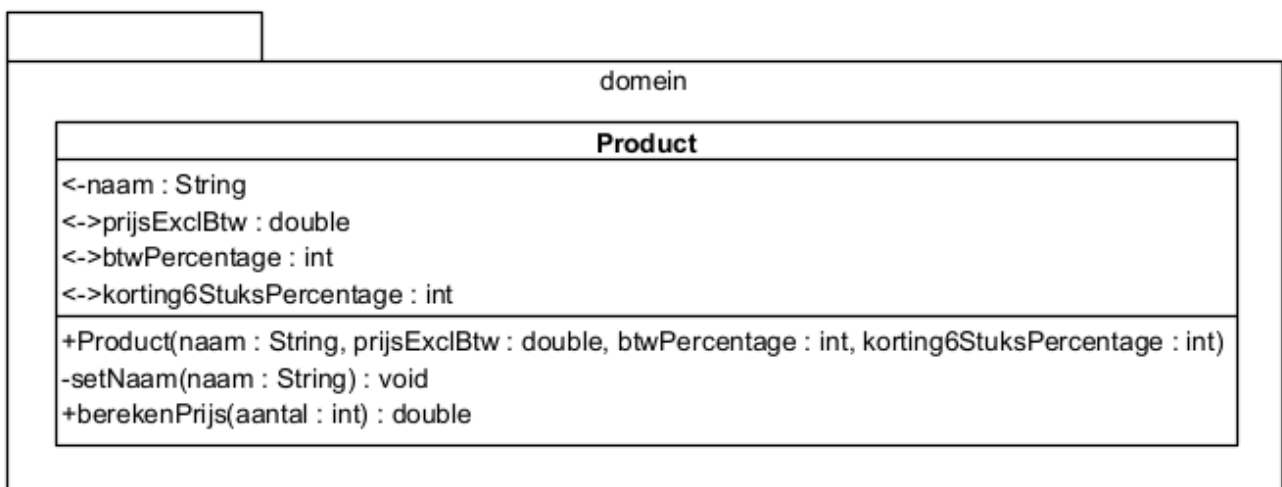
Om te weten of de oppervlakte van de rechthoek groter is dan 50 maak je gebruik van de voorziene methode uit de domeinklasse. Het bepalen of een driehoek rechthoekig is, hebben we ook al uitgewerkt in de domeinklasse. Communiceer dus met het object dat je aanmaakt om deze informatie te krijgen.



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.

## 2.8. Oefening - Product

### 2.8.1. Implementeer de domeinklasse Product.



Implementeer de nodige **attributen, getters en setters**.

- naam: geen controle



- btwPercentage: ligt tussen 6 en 21 (grenzen inbegrepen), defaultwaarde is 21
- korting6StuksPercentage: positief én maximaal 50, defaultwaarde is 0
- prijsExclBtw: geen controle

Maak een constructor voor deze klasse die onmiddellijk de 4 eigenschappen correct instelt.

De methode berekenPrijs berekent de prijs met btw voor het meegegeven aantal en houdt hierbij ook rekening met de eventuele korting vanaf 6 stuks.



Tekst kan zeker hier niet afgedrukt worden op het scherm, want we zitten in een domeinklasse. Deze bevat NOOIT uitvoer.

Ben je klaar? Vergeet dan niet je domeinklasse te testen m.b.v. de gegeven testklasse!

## 2.8.2. ProductApplicatie

Maak een applicatie (ProductApplicatie) die een marktkraam simuleert waar ze 3 verschillende producten verkopen:

Product	Prijs zonder btw	Btw	Korting vanaf 6 stuks
Pannenkoek	€0,125	21%	5%
Pannenkoek met suiker	€2,00	6%	10%
Pannenkoek met choco	€2,20	6%	15%

De klasse ProductApplicatie bevat alleen een main-methode.

Vraag aan een klant hoeveel stuks van elk product hij wilt en schrijf dan op de console uit hoeveel de klant zal moeten betalen (voorbeeld zie hieronder).

```
Hoeveel stuks wilt u van 'pannenkoek'? 2
Hoeveel stuk wilt u van 'pannenkoek met suiker'? 3
Hoeveel stuk wilt u van 'pannenkoek met choco'? 6
Aan de kassa zal u 18,56 EURO moeten betalen.
```

## 2.9. Oefening - Rekenmachine

### 2.9.1. Implementeer de domeinklasse Rekenmachine

Rekenmachine
<-resultaat : double
+telOp(getal1 : double, getal2 : double) : void +trekAf(getal1 : double, getal2 : double) : void +deel(getal1 : double, getal2 : double) : void +vermenigvuldig(getal1 : double, getal2 : double) : void

### Attributen, getters en setters

We voorzien 1 attribuut resultaat en een aantal methodes ( diensten om bewerkingen uit te voeren). Het attribuut resultaat bevat steeds het resultaat van de laatste rekenkundige bewerking.

### Methodes

De methodes voeren een rekenkundige bewerking uit: getal1 *bewerking* getal2. Respecteer de volgorde!

Klaar met implementeren? Testen maar!

## 2.9.2. RekenmachineApplicatie

RekenmachineApp
+main(args : String[]) : void -start() : void -leesGetal(boodschap : String) : double -toonResultaat(getal1 : double, getal2 : double, c : char, resultaat : double) : void -maakMenuKeuze() : void

Bouw een applicatie die volgende uitvoer heeft:

Voorbeeld 1:

```
<terminated> RekenmachineApp [Java Application]
Welke actie wil je uitvoeren?
 1. Som
 2. Verschil
 3. Vermenigvuldiging
 4. Deling
 5. Stoppen
Geef je keuze in: 5
|
```

Voorbeeld 2 :

```
<terminated> RekenmachineApp [Java Application]
Welke actie wil je uitvoeren?
  1. Som
  2. Verschil
  3. Vermenigvuldiging
  4. Deling
  5. Stoppen
Geef je keuze in: -9
Welke actie wil je uitvoeren?
  1. Som
  2. Verschil
  3. Vermenigvuldiging
  4. Deling
  5. Stoppen
Geef je keuze in: 6
Welke actie wil je uitvoeren?
  1. Som
  2. Verschil
  3. Vermenigvuldiging
  4. Deling
  5. Stoppen
Geef je keuze in: 5
```

Voorbeeld 3 :

```
<terminated> RekenmachineApp [Java Application]
```

```
Welke actie wil je uitvoeren?
```

1. Som
2. Verschil
3. Vermenigvuldiging
4. Deling
5. Stoppen

```
Geef je keuze in: 1
```

```
Geef het eerste getal: 12
```

```
Geef een tweede getal: 12
```

```
12,00 + 12,00 = 24,00
```

```
Welke actie wil je uitvoeren?
```

1. Som
2. Verschil
3. Vermenigvuldiging
4. Deling
5. Stoppen

```
Geef je keuze in: 2
```

```
Geef het eerste getal: 45
```

```
Geef een tweede getal: 8
```

```
45,00 - 8,00 = 37,00
```

```
Welke actie wil je uitvoeren?
```

1. Som
2. Verschil
3. Vermenigvuldiging
4. Deling
5. Stoppen

```
Geef je keuze in: 5
```

```
|
```

De klasse RekenmachineApplicatie heeft de gekende main-methode, start-methode en enkele hulpmethodes.

1. Schrijf de menutekst op het scherm
2. Laat de gebruiker een keuze ingeven (wordt gecontroleerd en bij foute invoer krijgt de gebruiker een herkansing) .
3. Maak gebruik van een Rekenmachine-object om de gekozen rekenkundige bewerking uit te voeren. Iedere nieuwe actie start met een rekenmachine met als resultaat 0.