

## Oefeningen Hoofdstuk 1

In IntelliJ heb je reeds een module aangemaakt met de naam “H1”.

Per opdracht en oefening maak je in deze module een aparte package. Voor oefening 1 geef je deze als naam “be.pxl.h1.oef1”.

### Oefening 1

Op de wikipedia pagina

[https://nl.wikipedia.org/wiki/Lijst\\_van\\_hoogste\\_gebouwen\\_van\\_Belgi%C3%AB](https://nl.wikipedia.org/wiki/Lijst_van_hoogste_gebouwen_van_Belgi%C3%AB) vind je de hoogste gebouwen van België. Maak nu een klasse Toren. Wat zijn de eigenschappen van de klasse Toren? Implementeer deze klasse en voorzie getters en setters.

Maak ook een klasse MijnToren met een hoofdprogramma. Kies 1 toren uit de lijst op wikipedia en maak een object voor deze toren. Vul alle eigenschappen in. Druk vervolgens alle eigenschappen af op het scherm.

### Oefening 2

Een handelaar in op maat gemaakte zwembaden wenst snel te kunnen uitrekenen hoeveel liter water en ontsmettingsmiddel in een zwembad van een bepaalde grootte kunnen. 100 liter zwembadwater moet 98 l water bevatten en 2 liter ontsmettingsmiddel. Maak een klasse Zwembad met 3 eigenschappen: lengte, breedte en diepte (uitgedrukt in meter, laat kommagetallen toe).

Voorzie de methode getVolume() die het totale volume (in liter) van het zwembad teruggeeft. Voorzie verder de methoden getVolumeWater() en getVolumeOntsmettingsmiddel().

In het hoofdprogramma laat je de gebruiker de afmetingen van het zwembad invoeren, waarna aantal liter water en aantal liter ontsmettingsmiddel getoond worden (zonder cijfers na de komma).

### Voorbeeld output:

Geef de breedte van het zwembad: 12,5

Geef de lengte van het zwembad: 25

Geef de diepte van het zwembad: 1,2

Liter water: 367500

Liter ontsmettingsmiddel: 7500

### Oefening 3

Een voetbalploeg uit eerste klasse wenst de prijs van zijn spelers in te schatten. Maak eerst een klasse Voetballer met volgende eigenschappen:

- naam speler
- leeftijd
- gemiddeld beoordelingscijfer van de sportjournalisten (cijfer tussen 0 en 10)
- type speler: "Aanvaller", "Middenvelder", "Verdediger", "Doelman"
- aantal doelpunten (= doelpunten gescoord, bij doelman: tegendoelpunten)

Op basis van deze gegevens wordt de prijs van een voetballer berekend.

#### *Berekening prijs:*

- Als basisprijs wordt voor een aanvaller €30000, voor een middenvelder €28000, voor een verdediger €29000 en voor een doelman €25000 gerekend.
- Als de speler jonger is dan 25 jaar, wordt deze basisprijs met 10% verhoogd. Als de speler ouder is dan 30 jaar, wordt deze basisprijs met 5% verlaagd. Bij spelers van 25 jaar t.e.m. 30 jaar oud wordt de basisprijs behouden.
- Voor aanvallers geldt: voor de eerste 5 doelen, wordt de prijs met 10 000 euro per doel verhoogd, vanaf het zesde doel, wordt de prijs met 20 000 euro per doel verhoogd.
- Voor middenvelders, verdedigers en doelmannen geldt: de prijs wordt verhoogd met 10 000 euro \* beoordelingscijfer (bv. iemand met beoordeling 8, bekommt een prijsverhoging van 80 000 euro).
- Voor doelmannen geldt bovendien: vanaf de 20ste goal die ze binnenlaten, daalt de prijs met 9 000 euro.

In het hoofdprogramma vraag je alle nodige gegevens van een voetballer en toon je vervolgens zijn prijs. Je kan voor meerdere voetballers de prijs berekenen, het programma stopt pas als er "stop" wordt ingegeven als naam van de voetballer.

```
Naam: Messi
Leeftijd: 34
Positie: Aanvaller
Beoordeling: 10
Doelpunten: 15
Prijs: €278500,00
Naam: Courtois
Leeftijd: 29
Positie: Doelman
Beoordeling: 8
Doelpunten: 22
Prijs: €96000,00
Naam: stop
```

## Oefening 4

Maak een klasse Hotel. Deze klasse heeft 3 eigenschappen:

- hotelcode (String)
- aantal sterren (getal van 1 (zeer simpel hotel) tot 5 (super-de-luxe hotel))
- kindercode ('A' t.e.m. 'Z' is toegelaten)

De methode getSterren() geeft het aantal sterren terug met \*-symbolen. Dus voor een 3-sterren hotel krijg je \*\*\*.

Implementeer in de klasse Hotel de methode getPrijs(). Deze methode berekent de prijs voor een volwassene voor 1 nacht. De prijs voor een volwassene voor 1 nacht wordt als volgt berekend:

- 4- en 5- sterrenhotels kosten 60 €
- 3-sterrenhotels kosten 56 € (uitgezonderd: 3-sterrenhotel met code "BR" of "AN, deze kosten eveneens 60 €)

- 2- en 1-sterrenhotels kosten 48 €

LET OP: hotelcode "HI" is steeds 70€

Daarnaast implementeer je in de klasse Hotel ook de methode getPrijsKind(). De prijs per kind wordt als volgt berekend:

- indien kindercode "A" is geldt: kinderen in 1- en 2-sterrenhotels zijn gratis, evenals het hotel met hotelcode "HA"
- In alle andere gevallen betalen kinderen 50% van de prijs van een volwassene.

Schrijf nu een programma. Je vraagt via toetsenbord eerst het aantal volwassenen en het aantal kinderen. Daarna maak je Hotel-objecten aan. Hiervoor vraag je input aan de gebruiker via het toetsenbord. Je vraagt eerst de hotelcode, vervolgens het aantal sterren en tenslotte de kindcode (controleer de invoer!). Het programma toont vervolgens de hotelcode, het aantal sterren (weergegeven met \*-symbolen), de prijs voor een volwassene voor 1 nacht, de prijs voor een kind voor 1 nacht en de totale prijs voor het opgegeven aantal volwassenen en kinderen. (Voorbeeld: HI\*\*\*\*\* 70,00 35,00 210,00)

Het programma stopt als er voor hotelcode een "/" wordt ingegeven.

### Voorbeeld:

```
Geef het aantal volwassenen: 2
Geef het aantal kinderen: 3
Geef een hotelcode: HI
Geef het aantal sterren: 5
Geef een kindercode: A
Geef een hotelcode: BR
Geef het aantal sterren: 2
Geef een kindercode: X
Geef een hotelcode: /
HI***** 70,00 35,00 245,00
BR  ** 48,00 24,00 168,00
```

## Oefening 5

Implementeer de klasse Spaceship.

Voorzie de eigenschappen:

- name (String)
- hits (int): aantal keer dat het spaceship is geraakt
- shieldOn (boolean): indien true, kan het spaceship niet geraakt worden.

Voorzie de volgende methoden:

- setShieldOn(boolean): methode om het shield in of uit te schakelen. Toon ook een gepaste melding in de console:  
bijv. `<name> puts shield on` of `<name> puts shield on`
- fire(SpaceShip): methode om op een ander spaceship te vuren. Indien het shield van het andere spaceship niet aan staat, wordt het spaceship geraakt. Indien het shield wel aanstaat, wordt het spaceship dat vuurt geraakt. Toon ook een gepaste melding: bv. `<name> fires at <other_name>`
- isDestroyed(): een spaceship is vernietigd als het 5 keer werd geraakt.

In een hoofdprogramma maak je 2 Spaceship-objecten aan. Laat ze een aantal keer op elkaar vuren. Zet het shield van 1 van de spaceships aan en vuur nog een aantal keren.

Toon de status (isDestroyed()) van beide Spaceship-objecten.

Voorbeeld output:

```
BS Galactica fires at STS Shade
STS Shade fires at BS Galactica
BS Galactica puts shield on
STS Shade fires at BS Galactica
BS Galactica fires at STS Shade
BS Galactica fires at STS Shade
BS Galactica fires at STS Shade
BS Galactica fires at STS Shade
BS Galactica is destroyed? false
STS Shade is destroyed? true
```

## Oefening 6

Maak een klasse met de naam Phone om telefoon-objecten voor te stellen.

Elke telefoon heeft de volgende eigenschappen:

- number (String)
- provider (String)
- numberOfCalls (int)
- numberOfFreeCalls (int)

Voorzie getters en setters voor deze eigenschappen.

Voorzie de methode call(String) met een telefoonnummer als parameter. Deze methode houdt het aantal telefoongesprekken bij (numberOfCalls en numberOfFreeCalls).

Telefoontjes naar de volgende nummers zijn gratis: "112", "102", "070245245". Bij provider Floximus betaal je 25 cent per telefoontje, maar krijg je 5% korting op je totale kost als je meer dan 20 (betalende) telefoontjes hebt gedaan. Bij andere providers betaal je 21 cent per telefoontje. Voorzie de functie getCost() om de telefoonkosten te berekenen.

Voorzie de methode reset() om de tellers (numberOfCalls and NumberOfFreeCalls terug op 0 te zetten).

In het hoofdprogramma test je de klasse Phone uit. Je belt achtereenvolgens van nummer 012/12.34.10 tot 012/12.34.30. Tenslotte bel je ook naar het anti-gifcentrum (070/245.245). Toon de telefoonkosten voor en na een reset.

Voorbeeld uitvoer:

```
Calling 012123410
Calling 012123411
Calling 012123412
Calling 012123413
Calling 012123414
Calling 012123415
Calling 012123416
...
```

```
Calling 012123429
Calling 012123430
Calling 070245245
Phone cost: €4,99
Phone cost after reset: €0,00
```

## Oefening 7

Maak een klasse Sword.



Een Sword-object heeft de eigenschappen material (String) en durability (int). Voorzie getters en setters. Maak een aantal Sword-objecten en plaats deze in een array. Doorloop de array om het zwaard met de hoogste duurzaamheid op te zoeken. Toon het materiaal van dit zwaard.

Most durable sword is of Diamond

Vervolgens print je van alle Sword-objecten in je array een tabel met materiaal en duurzaamheid zoals hier getoond.

Materiaal	Duurzaamheid
Wood	33
Stone	500
Diamond	1500

## Oefening 8

Implementeer een klasse Lift. Een lift heeft een eigenschap aantalVerdiepingen en ook een eigenschap die bijhoudt op welke verdieping de lift zich bevindt. Verder heeft de lift een aantal toegelaten personen en houdt de lift ook bij hoeveel personen er in de lift zitten.

De lift heeft een methode betreed() waarbij in een parameter wordt meegegeven hoeveel personen de lift betreden. De methode verlaat() zorgt dat het opgegeven aantal personen de lift verlaten (Let op: er kunnen nooit minder dan 0 personen in de lift staan en nooit meer dan het toegelaten aantal)

Maak tenslotte een methode gaNaar() waarbij je een verdieping als parameter doorgeeft. Afhankelijk van de huidige verdieping gaat de lift naar boven of beneden bewegen tot de opgegeven verdieping (controleer wel dat de opgegeven verdieping bestaat!). De lift zal niet verplaatsen indien er geen personen in de lift staan. Bij het verplaatsen toont de lift alle nummers van verdiepingen waar hij passeert (incl. startverdieping en stopverdieping).

In het hoofdprogramma maak je een lift aan voor max. 5 personen met 11 verdiepingen (verdieping 0 tot en met 10). Vervolgens toon je een menu zoals hieronder weergegeven. Het programma stopt pas als de gebruiker keuze 4 kiest.

Maak uw keuze:

1. Ga naar verdieping ...
2. Betreed de lift...
3. Verlaat de lift...
4. Stoppen

2

Aantal personen:

2

Maak uw keuze:

1. Ga naar verdieping ...
2. Betreed de lift...
3. Verlaat de lift...
4. Stoppen

1



Kies een verdieping:

4

---

|0|

---

---

|1|

---

---

|2|

---

---

|3|

---

---

|4|

---

Maak uw keuze:

1. Ga naar verdieping ...
2. Betreed de lift...
3. Verlaat de lift...
4. Stoppen

3

Aantal personen:

1

Maak uw keuze:

1. Ga naar verdieping ...
2. Betreed de lift...

3. Verlaat de lift...

4. Stoppen

1

Kies een verdieping:

1

---

|4|

---

---

|3|

---

---

|2|

---

---

|1|

---

Maak uw keuze:

1. Ga naar verdieping ...

2. Betreed de lift...

3. Verlaat de lift...

4. Stoppen

4

## Extra Oefeningen

### Oefening 1

Geef een oneven getal tussen 5 en 49 in via het toetsenbord. Doe hierop een invoercontrole.

Bij ingave van een fout getal, wordt een nieuw gevraagd.

Vervolgens wordt een “x” gedrukt in de opgegeven grootte.

Vb ingegeven getal = 5

Output:

```

X X
X X
X
X X
X X

```

### Oefening 2

Een van de meest voorkomende elektronische componenten is een weerstand. Maak een klasse Weerstand met 3 eigenschappen: band1, band2 en band3 (allemaal char).

Een weerstand wordt gekarakteriseerd door zijn weerstandswaarde R. Deze waarde valt af te lezen aan de hand van een kleurcode onder de vorm van drie bandjes. Volgende omzettingstabel is van toepassing:

<i>kleur</i>	<i>waarde</i>	<i>Letter</i>
zwart	0	z
bruin	1	b
rood	2	r
oranje	3	o
geel	4	g
groen	5	G
blauw	6	B
violet	7	V
grijs	8	L
Wit	9	W

Zijn a, b en c respectievelijk de waarden van de 1ste, 2de en 3de band, dan wordt de waarde van de weerstand R gevonden uit :  $R = (10 \times a + b) \times 10^c$

Voorzie de methode `getWeerstandswaarde()` waarin je bovenstaande berekening uitvoert.  
 Voorzie ook de methode `getKleur()` die de kleuren van de 3 banden toont (bijv. groen geel grijs).  
 Schrijf een programma dat 3 karakters inleest (zie tabel), en vervolgens de omzetting naar R uitvoert. Je mag ervan uitgaan dat enkel geldige karakters worden ingegeven.  
 Maak bij het oplossen van deze oefening op een zinvolle manier gebruik van arrays.

## Oefening 3

Je krijgt de opdracht van een tv-zender om een programma te schrijven om de punten te berekenen voor het programma “Komen eten”. Maak een klasse `Deelnemer`. Een deelnemer heeft een naam, een array met punten voor sfeer en een array met punten voor eten.

Het programma start met de ingave van het aantal deelnemers.

Vervolgens worden de namen van de deelnemers gevraagd.

Daarna kunnen de punten voor de deelnemers worden ingegeven. Verzamel de punten voor sfeer en eten voor iedere deelnemer in een array en hou ze bij in de `Deelnemer`-objecten.

Eerst worden de punten voor de eerste deelnemer ingegeven. Iedere andere deelnemer geeft een punt voor SFEER en een punt voor ETEN. Er wordt steeds een punt van 0 tot 10 (incl.) gegeven. Controleer de ingevoerde waarde. Indien een ingegeven waarde niet correct is, vraag je om de waarde opnieuw in te geven. Nadat alle deelnemers hun punten voor de eerste deelnemer hebben ingegeven, worden de punten voor de tweede deelnemer ingegeven, daarna voor de derde,...

Als alle punten zijn ingegeven wordt voor iedere deelnemer zijn totaal behaalde punten afgedrukt en wordt de naam van de winnaar afgedrukt. Voor het berekenen van het totaal aantal punten voorzie je een methode `getScore()` in de klasse `Deelnemer`.

### **Voorbeeld output:**

Geef het aantal deelnemers:

3

Geef naam van deelnemer 1:

Kwik

Geef naam van deelnemer 2:

Kwek

Geef naam van deelnemer 3:

Kwak

Punten voor Kwik

Kwek geef uw punten voor SFEER:

5

Kwek geef uw punten voor ETEN:

6

Kwak geef uw punten voor SFEER:

4

Kwak geef uw punten voor ETEN:

8

Punten voor Kwek

Kwik geef uw punten voor SFEER:

8

Kwik geef uw punten voor ETEN:

9

Kwak geef uw punten voor SFEER:

7

Kwak geef uw punten voor ETEN:

9

Punten voor Kwak

Kwik geef uw punten voor SFEER:

-1

Ongeldige punten!

Kwik geef uw punten voor SFEER:

11

Ongeldige punten!

Kwik geef uw punten voor SFEER:

8

Kwik geef uw punten voor ETEN:

5

Kwek geef uw punten voor SFEER:

6

Kwek geef uw punten voor ETEN:

7

Kwik = 23

Kwek = 33

Kwak = 26

Winnaar: Kwek