

## Oefeningen Hoofdstuk 9

Maak een Java Project aan met naam "H9".

Per oefening maak je een aparte package. Voor oefening 1 geef je deze als naam "be.pxl.h9.oef1".

### Oefening1

In deze oefening worden een aantal klassen aangemaakt om persoonsgegevens, datums en adressen bij te houden zodat we uiteindelijk 2 personen kunnen laten huwen en oa met 1 bewerking ervoor zorgen dat ze op hetzelfde adres wonen.

De oefening wordt stapsgewijs opgebouwd, met na elke stap de mogelijkheid tot testen in de main-methode.

Vermijd om 2 keer dezelfde code te programmeren en hergebruik zoveel mogelijk code die je al hebt!

#### Klasse Gemeente

1. Maak een klasse Gemeente met de volgende kenmerken: postcode (int), gemNaam (String). De kenmerken mogen van buiten de klasse niet toegankelijk zijn.
2. Voorzie een constructor om een gemeente te creëren door een waarde mee te geven voor beide eigenschappen.
3. Voorzie getters en setters om de waarde te kunnen opvragen en wijzigen.
  - Voorzie een controle op de postcode, deze moet 4 posities lang zijn. Indien ze te lang is worden de eerste 4 posities genomen. Indien ze te kort is wordt ze achteraan aangevuld met nullen.
4. Voorzie een methode toString() die de gemeentegegevens teruggeeft zoals "3290 Diest".

#### Extra

Zorg ervoor dat de gemeentenaam omgevormd wordt volgens volgende regels:

- Eerste letter wordt een hoofdletter.
- Verder worden alle letters omgezet naar kleine letters behalve tekens na een spatie of \*'-', deze blijven zoals ze ingevoerd werden.
- Cijfers worden verwijderd.

Voorbeeld: hE78rk-7dE-Stad

wordt omgevormd tot: Herk-de-Stad

Maak een klasse `HuwelijkApp` met een `main`-methode waarin je een gemeente-object aanmaakt en alle methoden uittest.

## klasse Adres

1. Maak een klasse `Adres` met de volgende kenmerken: `straat` (`String`), `huisnr` (`String`), `gemeente` (`Gemeente`). De kenmerken mogen van buiten de klasse niet toegankelijk zijn.
2. Voorzie een constructor om een adres te creëren door 4 waarden in te geven, vb "kerkstraat", "12b", 3290, "Diest".
3. Voorzie getters en setters om de waarde van straat en huisnr te kunnen opvragen en wijzigen. Voorzie een `get`-methode om de gemeente op te halen.
4. Voorzie een methode `toString()` die de gemeentegegevens teruggeeft zoals:  
kerkstraat 12b  
3290 Diest

Maak in de `main`-methode van de klasse `HuwelijkApp` een adres-object aan en test alle methoden uit.

Pas de gemeentenaam van een adres aan.

## klasse Datum

1. Maak een klasse `Datum` (of gebruik de klasse `Datum` van `H8d2Oef2`) volgens onderstaand UML-schema:

Datum
- dag : int - maand : int - jaar : int - maandNamen : String [ ] = {"januari", "februari", "...", "december"}
+ Datum() + Datum( d : int, m : int, j : int ) + Datum( d : Datum ) + setDag( d : int ) + setMaand( m : int ) + setJaar( j : int ) + getDag() : int + getMaandNr() : int + getMaandNaam() : String + getJaar() : int + toString() : String

Hou rekening met het volgende:

- De default-datum is 1/1/2017.
- De waarde voor 'maand' mag nooit meer zijn dan het aantal maanden in de maandNamen-array en nooit kleiner dan 1. Als er een waarde wordt ingegeven kleiner dan 1, wordt de waarde 1. Als er een waarde wordt ingegeven groter dan 12, wordt de waarde 12.
- De methode toString() geeft de datum als een String terug als volgt: 5 april 2017.

## klasse Persoon

1. Maak een klasse Persoon met de volgende kenmerken: naam (String), voornaam (String), geboortedatum (Datum) en adres (Adres). De kenmerken mogen van buiten de klasse niet toegankelijk zijn.
2. Voorzie een constructor om een persoon aan te maken met volgende parameters: "Aerts", "Jef", 29, 11, 1990, "lindestraat", "23D", 3500, "Hasselt".  
Voorzie een tweede constructor om een persoon aan te maken door naam, voornaam, geboortedatum en adres mee te geven.  
Roep op een zinvolle manier 1 constructor op vanuit de andere.
3. Voorzie get- en set-methoden voor naam en voornaam en adres.  
Voorzie een get-methode voor geboortedatum.
4. Maak een methode voegVoornamenToe() om in 1 keer meerdere voornamen aan een persoon te kunnen toevoegen, vb p.voegVoornamenToe("An", "Els", "Kim");  
Alle voornamen worden toegevoegd aan de String voornaam met telkens een spatie tussen.
5. Voorzie een methode toString() die de persoonsgegevens als volgt teruggeeft:  
An Els Kim Van Brussel  
sparreweg 18  
3500 Hasselt

Maak in de main-methode van de klasse HuwelijkApp 2 persoon-object aan en test alle methoden uit.

Wijzig de gemeentenaam van het adres van een persoon.

Wijzig de geboortedatum van een persoon.

## klasse Huwelijk

1. Maak een klasse Huwelijk waarin je 2 partners en een huwelijksdatum bijhoudt.
2. De initialisatie van een huwelijk-object gebeurt door opgave van 2 personen en 3 getallen (waaruit de datum samengesteld wordt).  
Pas het adres van de tweede partner aan aan het adres van de eerste.  
Geef een melding van initialisatie zoals in volgend voorbeeld: Niels Bex en An Vos zijn gehuwd op 5 mei 2015. Proficiat!
3. Zorg ervoor dat van een huwelijk de 2 partner-objecten en het datum-object kunnen opgevraagd worden.
4. Maak een methode adresWijziging() die bvb volgende waarden binnenkrijgt als parameter: "lentestraat", "15C", 3500, "Hasselt" en stel het adres van beide partners in op dit nieuwe adres.
5. Voorzie een methode print() die de gegevens als volgt afdruckt:

```
An Adams
lentestraat 3B
3290 Diest
Geboren op 29 september 1990

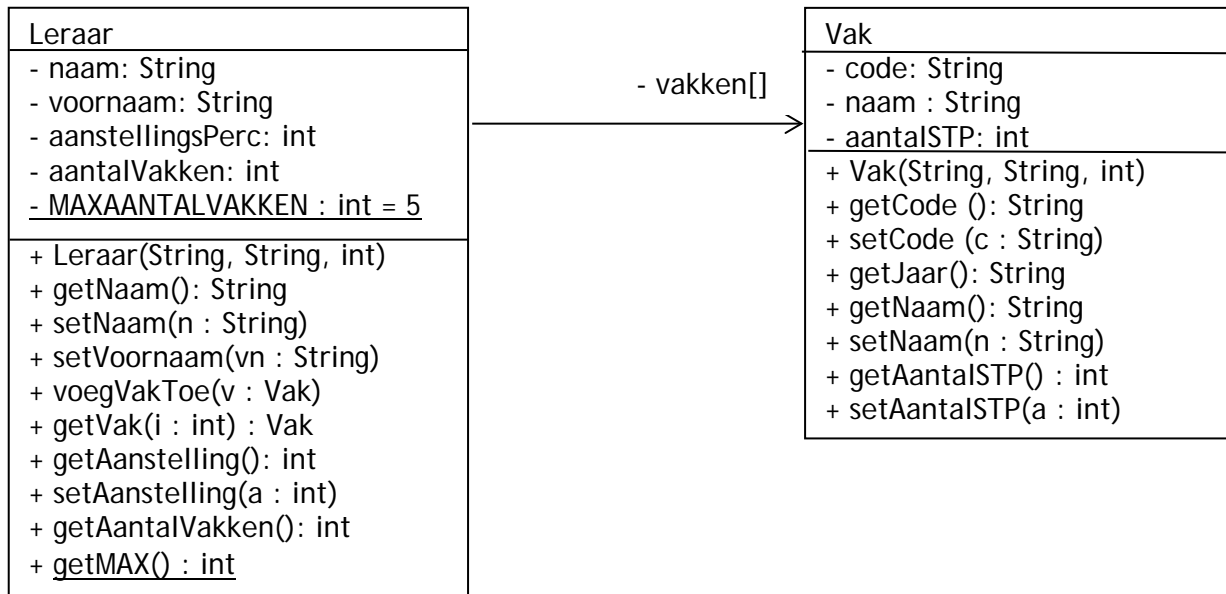
Jo Bex
lentestraat 3B
3290 Diest
Geboren op 12 juni 1992

het huwelijk vond plaats op 3 maart 2016
```

Maak in de main-methode van de klasse HuwelijkApp een huwelijk-object aan.  
Vraag het huwelijksjaar van het huwelijk op.  
Druk de persoonsgegevens van partner1 af.  
Doe een adreswijziging.  
Druk alle gegevens van het huwelijk af.

## Oefening2

Gegeven onderstaand UML-diagram:



Bijkomende informatie :

1. Een vakcode bestaat uit 2 cijfers, 3 letters, gevolgd door 4 cijfers. Je hoeft dit niet volledig te controleren. Doe wel een controle of het jaar 1, 2 of 3 is. Zo niet, maak er jaar 1 van.  
Bijkomende uitleg:  
cijfer1: departementscode (4 = IT, 3 = Business)  
cijfer2: jaar waarin het vak gegeven wordt  
3 letters: afkorting voor opleiding  
4 cijfers: eigenlijke vakcode  
vb 41TIN1130 is de code voor het opleidingsonderdeel 'Java Essentials' van het eerste jaar van het departement IT.
2. aantalSTP= aantal studiepunten. Dit kan nooit meer zijn dan 18 en moet een veelvoud van 3 zijn. Controleer dit, en rond af naar het dichtstbijzijnde correcte getal.
3. Een leraar kan voor maximum 100% aangesteld worden. Bij een te groot percentage wordt een foutmelding gegeven en wordt het percentage afgeknot op 100. Zorg er ook voor dat het aanstellingspercentage steeds een veelvoud van 10 is. Rond steeds af naar het dichtstbijzijnde tiental.
4. MAXAANTALVAKKEN: elke leraar kan maximaal 5 vakken geven : controleer hierop als je vakken aan een leraar toevoegt. Hij/zij kan ook geen vakken geven.
5. aantalVakken: het aantal vakken dat reeds toegewezen is aan de leraar.
6. Derde parameter in constructor Leraar = aanstellingspercentage

Maak een klasse PXLApp met een main-methode waarin je het volgende doet:

1. Kopieer volgende array-definities in je main-methode:

```
String codeTabel[] = {"41TIN1100", "48TIN1130", "41TIN1140",
    "41TIN1160", "41TIN1240", "41TIN1290",
    "41TIN1300", "42TIN1220", "42TIN1230",
    "43AON3120", "43AON3130"};
String [] naamTabel = {"Web Essentials", "Java Essentials",
    ".Net Essentials", "SQL Essentials",
    "Desktop OS", "Math for IT", "Programming Basics",
    ".Net Advanced", "Java Advanced", "Programming Expert",
    "Mobile development"};
int [] stpTabel = {7, 6, 6, 3, 6, 6, 5, 3, 3, 3, 3};
```

2. Definieer een array van Vak-objecten:

```
Vak [] vakken = new Vak[codeTabel.length];
```

en vul deze op met vakken die je samenstelt op basis van de gegeven arrays.

3. Maak een Leraar-object en voeg vakken toe.
4. Test alle methoden uit.
5. Voorzie een methode print() waarbij je de nodige waarden als parameters meegeeft om een afdruk te maken als volgt:

```
Leraar Daems Greta is aangesteld voor 70%
Volgende vakken behoren tot het takenpakket:
Vakken van jaar 1:
    41TIN1130    Java Essentials           6
    41TIN1300    Programming Basics           6
Vakken van jaar 2:
    42TIN1220    .Net Advanced                 3
    42TIN1230    Java Advanced                 3
Vakken van jaar 3:
    43AON3120    Programming Expert            3
```

## Extra - uitbreiding (moeilijker):

In de main-methode worden de leraargegevens (naam, voornaam, aanstellingsperc) opgevraagd via het toetsenbord. Vervolgens wordt gevraagd hoeveel vakken men wil ingeven voor deze leraar.

Alle vakken worden op het scherm getoond met een volgnr. De gebruiker geeft een nr in waarna het bijhorend vak uit de array wordt toegekend aan de leraar. Vervolgens worden de vakken terug getoond, behalve het reeds toegekende vak, ....

Bouw eveneens een controle in of het gekozen vak al niet behoort tot het takenpakket van de betreffende leraar.