

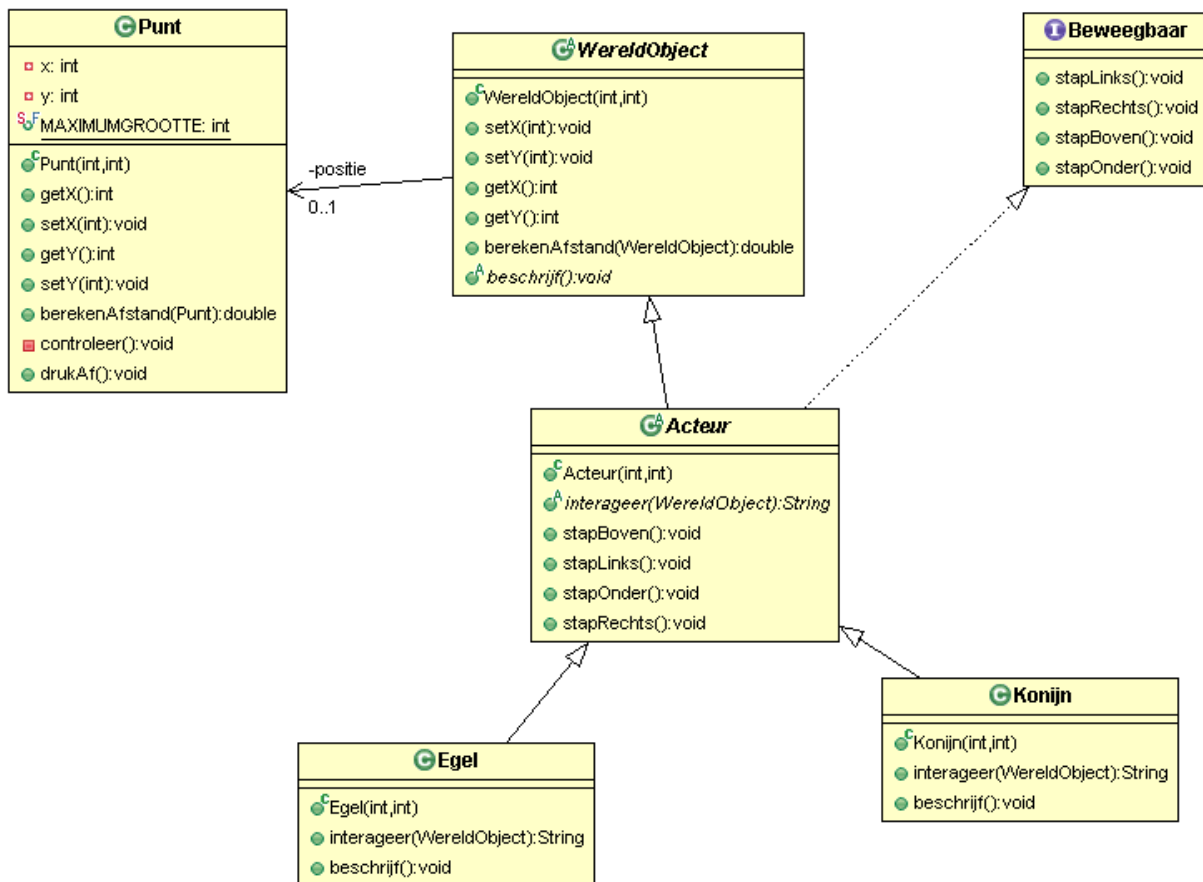
## Oefeningen Hoofdstuk 13

Maak een Java Project aan met naam "H13".

Per oefening maak je een aparte package. Voor oefening 1 geef je deze als naam "be.pxl.h13.oef1".

### Oefening1

Maak de interface Beweegbaar en de klassen Punt, WereldObject, Acteur, Egel en Konijn aan de hand van onderstaand klassendiagram.



### Punt.java

- De eigenschappen `x` en `y` geven de `x,y`-coördinaat van het punt weer. `MaximumGrootte` is de maximale waarde voor deze coördinaten. `MaximumGrootte` wordt vastgelegd op waarde 99.
- Via de methode `berekenAfstand()` wordt de afstand berekend tussen het huidige punt en een punt `p2` dat als argument meegegeven wordt. De afstand tussen 2 punten  $(x_1, y_1)$  en  $(x_2, y_2)$  kan berekend worden aan de hand van de formule:

$$afstand = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

- Via de methode controleer() wordt er voor gezorgd dat x en y tussen de waarden 0 en maximumGrootte liggen. Een negatieve waarde voor x of y wordt gelijk gesteld aan 0, een te grote waarde voor x of y wordt gelijk gesteld aan maximumGrootte. Zorg ervoor dat deze methode aangeroepen wordt telkens als x of y gewijzigd wordt.
- Via de methode drukAf() worden de gegevens afgedrukt in de vorm x = 21, y = 68.

## WereldObject.java

- WereldObject is een abstracte klasse.
- De klasse WereldObject heeft positie als eigenschap. Positie is een object van de klasse Punt. Via de constructor worden de waarde van x en y van positie vastgelegd.
- Via de methode berekenAfstand() wordt de afstand berekend tussen het huidige WereldObject en een WereldObject dat als argument meegegeven wordt.
- Beschrijf is een abstracte methode.

## Acteur.java

- Acteur is een abstracte klasse. In de constructor van de klasse Acteur wordt de constructor van de superklasse aangeroepen om de x,y waarde van positie vast te leggen.
- Interageer() is een abstracte methode.
- Via
  - stapBoven() wordt 1 opgeteld bij de y-waarde van positie,
  - stapLinks() wordt 1 afgetrokken van de x-waarde van positie,
  - stapOnder() wordt 1 afgetrokken van de y-waarde van positie,
  - stapRechts() wordt 1 opgeteld bij de x-waarde van positie.

## Egel.java en Konijn.java

- In de constructor van de klasse Egel en Konijn wordt de constructor van de superklasse aangeroepen om de x, y waarde van positie te bepalen.
- Via de methode interageer() wordt de interactie tussen konijnen en egels bepaald. Een egel groet een egel als deze zich binnen een straal van 20 bevindt, een egel groet een konijn als het konijn zich binnen een straal van 10 bevindt. Konijnen groeten niemand. Met 'groeten' wordt bedoeld het geven van de boodschap 'Dag konijn' of 'Dag egel'.

- Via de methode `beschrijf()` ten slotte wordt een beschrijving gegeven van de vorm  
 Ik ben een konijn op positie  $x = 11$  ,  $y = 44$   
 of Ik ben een egel op positie  $x = 61$  ,  $y = 33$ .

### ActeurApp.java

- Maak de klasse `ActeurApp`. In de main-methode van deze klasse wordt de array 'puntenTab' aangemaakt. Zorg er voor dat enkel objecten van de klasse `Punt` in deze array geplaatst kunnen worden.  
 Maak vervolgens 100 objecten van de klasse `Punt` aan. Gebruik hierbij willekeurige (random) waarden tussen 0 en 99 (0 en 99 inbegrepen) voor  $x$  en  $y$ .  
 Doorloop de tabel 'puntenTab' en druk van elk punt de gegevens af.
- Bereken de afstand tussen het eerste en het laatste punt in `puntenTab` en druk dit af.
- Maak een `Egel`-object en een `Konijn`-object aan.
- Pas de methode `interageer()` toe op het `Egel`-object.
- Pas de methode `beschrijf()` toe op het `Konijn`-object.
- Het `Konijn` zet een stap naar links.
- De `Egel` zet een stap naar rechts.
- Maak een array 'act' waarin je zowel het `Konijn`-object als het `Egel`-object toevoegt.  
 Doorloop de array en druk van elk element de bijhorende informatie af.

## Oefening2

Maak een klasse `BouwGrond`:

- eigenschappen: `perceelNr` (String), `perceelGrootte` (float) en `bouwVoorschrift` (String).
- 1 constructor met een binnenkomende waarde voor `perceelNr` en `perceelGrootte`.
- voorzie getters en setters voor alle eigenschappen.
- zorg ervoor dat geen object van de klasse `BouwGrond` gemaakt kan worden.

Ter info: de `perceelGrootte` is de grootte van de grond uitgedrukt in are (1 are = 100 m<sup>2</sup>).

In elke klasse afgeleid van de klasse `BouwGrond` moeten de volgende methoden geïmplementeerd worden:

- + `wijsEenNotarisToe(String, LocalDate): void`
- + `doeEenBod (double, LocalDate): void`

Maak hiervoor de interface `Verkoopbaar` en gebruik deze interface in de klasse `BouwGrond`. De interface beschikt ook over de constante `minPrijsM2` (dit is een minimum prijs per m<sup>2</sup> bouwgrond waaronder nooit verkocht wordt). Deze krijgt een waarde van 83 euro.

Maak een klasse `TeVerkopenBouwGrond`.

Deze heeft alle eigenschappen van een `BouwGrond` en daarenboven nog:

- `notaris`: dit is de voor- en familienaam van de notaris die de bouwgrond verkoopt
- `datumTeKoop`: dit is de datum waarop de bouwgrond te koop werd gesteld.
- `hoogsteBod`: dit is het hoogste bod tot nog toe op deze bouwgrond.
- `datumHoogsteBod` : dit is de datum waarop het hoogste bod werd gedaan.

Via de constructor worden waarden voor `perceelNr`, `perceelGrootte` en `bouwVoorschrift` voorzien.

Voorzie de nodige getters.

De methoden van de interface worden in deze klasse uitgewerkt:

- `wijsNotarisToe()`: Er kan slechts 1 maal per `TeVerkopenBouwGrond` een notaris worden toegewezen! Op dat moment wordt ook de `datumTeKoop` ingesteld. Deze kan later niet meer gewijzigd worden.
- `doeEenBod()`: een bod kan slechts geregistreerd worden als een notaris aan de `TeVerkopenBouwGrond` is toegekend en de datum van het bod minstens 10 dagen na

de datumTeKoop valt. Druk een melding af indien toch een bod wordt gedaan zonder notaris of als er te vroeg wordt geboden.

Een bod wordt enkel geregistreerd als het groter is dan een eventueel vorig bod, en als het hoog genoeg is rekening houdend met de minimumprijs (bereken de minimumprijs adhv de eigenschappen minPrijsM2 en grootte).

Bij registratie van een bod wordt het volgende afgedrukt. Uiteraard telkens met de juiste gegevens.

Perceelnr: 12ER

Perceelgrootte: 12.4

Bouwvoorschrift: Open Bebouwing

Te koop gesteld op woensdag 07 december 2016 bij notaris Dirk Peeters

Vorig hoogste bod 150.000 geregistreerd op 23/12/16

Nieuw hoogste bod 175.000 geregistreerd op 04/01/17

Opmerking : De tekst vorig hoogste bod dient alleen te verschijnen als er al een bod geweest is.

Maak een klasse VerkoopApp en test het voorgaande uit.

## Extra oefeningen hoofdstuk 13

Gebruik het Java Project met de naam "H13".

Per oefening maak je een aparte package. Voor oefening 1 geef je deze als naam "be.pxl.h13.exoef1".

### Extraoefening 1

Maak de klasse **Huisnummer**.

- Elk huisnummer wordt gekenmerkt door een getal en een symbool.
- Maak de constructor voor Huisnummer, die als volgt gebruikt kan worden  

```
HuisNummer h1 = new HuisNummer(12, 'a');
```

In deze constructor wordt gecontroleerd of het getal groter is dan 0 en kleiner is dan 1000. Ook moet symbool een letter zijn of een spatie. Indien niet aan deze voorwaarden voldaan is wordt voor getal waarde -1 en voor symbool een spatie gebruikt.
- Maak ook de copy constructor die als volgt gebruikt kan worden  

```
HuisNummer h2 = new HuisNummer(h1);
```
- Maak de methode getHuisnummer(). Deze methode heeft geen argument en geeft een String met daarin getal en symbool in terug.  

```
String s = h1.getHuisnummer();
```

// s bevat nu "12a"
- Zorg ervoor dat 2 huisnummers als gelijk beschouwd worden als ze dezelfde waarde voor getal en symbool hebben (equals).

Maak de abstracte klasse **Structuur**.

- Structuur heeft als eigenschap de Huisnummer nummer.
- Maak de abstracte methode beschrijf(). Deze methode heeft geen argumenten en er wordt niets teruggekeerd.
- Maak de constructor die een Huisnummer als argument meekrijgt. In deze constructor wordt de copy constructor van HuisNummer aangeroepen om nummer zijn waarde te geven.
- Voorzie de getter voor nummer.
- Zorg ervoor dat 2 Structuren als gelijk beschouwd worden als ze gelijke Huisnummers hebben (equals).

Maak de klasse **Huis**.

- Huis is afgeleid van Structuur.
- Als eigenschap heeft Huis de variabele eigenaar van datatype String.
- Maak de constructor voor Huis die als volgt gebruikt kan worden.  

```
Huis h = new Huis("Geert Veren", new HuisNummer(12,'a'));
```
- Zorg dat de methode beschrijf() toegepast op h de volgende uitvoer heeft:  
 Geert Veren woont op nummer 12a

Maak de klasse **Park**.

- Park is afgeleid van Structuur.
- Er zijn geen extra eigenschappen in Park.
- Park kan aangemaakt worden als.  

```
Park p = new Park(new HuisNummer(13,' '));
```
- Zorg dat de methode beschrijf() toegepast op p de volgende uitvoer heeft:  
 Park gelegen op nummer 13

Maak de interface **Doorloopbaar**.

- Deze bevat methodes getAantal() en getWaarde().
  - ✓ getAantal() heeft geen argumenten en geeft een int terug.
  - ✓ getWaarde() heeft een int als argument en geeft een Object terug.
- Deze bevat de constante MAX. Deze waarde wordt ingesteld op 10.

Maak de klasse **Straat**.

- Straat maakt gebruik van de interface Doorloopbaar.
- Straat heeft als eigenschap een array met naam 'structuren' waarin objecten van de klasse Structuur geplaatst kunnen worden. De grootte van deze array wordt bepaald door de constante MAX.
- In de default constructor wordt de array geïnitieerd.
- Maak de methode voegStructuurToe(). Deze methode krijgt een Structuur s als argument mee. Er wordt eerst gekeken of s al in de array staat. Indien dit niet zo is dan wordt s achteraan toegevoegd.

Uitbreiding : maak een nieuwe grotere array (telkens met grootte MAX vergroten) aan zodra de oorspronkelijke array vol is. Zorg dat alle structuren uit de kleinere array in de grotere array zitten.

Maak de klasse StraatApp.

- Maak in de main methode de Straat s aan.
- Voeg aan s een huis met eigenaar Jan en nummer 12a toe.
- Probeer aan s een huis met eigenaar Geert nummer 12a toe te voegen. (Dit lukt niet omdat 12a al toegevoegd is.)
- Voeg aan s een park met nummer 13b toe.
- Vervolledig in de main ook onderstaande code

```
for (int i = 0; i < s.getAantal(); i++){
    *** s.***.***;
}
```

om een opsomming van de beschrijvingen in s te krijgen:

Jan woont op nummer 12a

Park gelegen op 13b