

# Java's spelregels

0. **Java** bestaat enkel uit *klassen* [class].
1. Een **klasse** bestaat uit *attributen* en *methodes*.
2. Een **attribuut** is van het enkelvoudige type [int, float, double, boolean, char] of het type is een klasse (attribuut is dan een object).
  - Een niet-ingevulde object-variabele heeft de waarde null.
3. Een **methode** is zoals een procedure (of ook functie genoemd) met een *header* ('eerste lijn' van de methode, met zijn *parameters*) en een *implementatie*.
  - Er zijn 2 speciale methodes : de *constructor* (met zelfde naam als klasse) die opgeroepen wordt bij het aanmaken van een object en de *main* (1 per klasse) die het begin van een programma angeeft.
4. Een **object** is een *geïnstantieerde klasse* [new], waarbij elk object eigen waarden voor elk attribuut heeft. Bij het instantiëren wordt de constructor van de klasse uitgevoerd.
  - Op de *eerste lijn van een constructor* kan met *this (...)* een andere constructor van dezelfde klasse opgeroepen worden en met *super (...)* die van de superklasse.
  - De **default constructor** heeft geen parameters en een lege implementatie. Als er geen andere constructor bestaat, wordt de default constructor automatisch door java aangemaakt.
5. De **implementatie** van een methode bestaat uit *statements* en *lokale variabelen* (enkelvoudig of objecten).
  - Een **lokale variabele** bestaat enkel binnen een methode, net zoals de parameters van de methode.
  - De statements hebben toegang tot alle attributen en methodes van het eigen object en die van object variabelen via de *'.'*.
  - De statements zijn gelijkaardig aan die van procedurele talen (zoals Modula-2, Pascal of C)
6. Een klasse kan **erven** van 1 *superklasse* [extends].
  - De *subklasse* erft alle attributen en methodes van de superklasse over.
  - Methodes met dezelfde header als een methode van de superklasse *overschrijven* die methode, bij een object zal de nieuwe methode opgeroepen worden ipv. Die van de superklasse.
7. Een **abstracte klasse** heeft minstens 1 *abstracte methode* [abstract]
  - Een **abstracte methode** heeft *geen implementatie*, enkel een header [abstract]
  - Een abstracte klasse kan *niet geïnstantieerd* worden.
  - Een subklasse moet *alle abstracte methodes* van de superklasse *overschrijven* om geïnstantieerd te kunnen worden.
8. Een **interface** is een abstracte klasse met *enkel abstracte methodes* [interface].
  - Een klasse mag meerdere interfaces als superklasse hebben [implements].
9. **Finale** attributen of lokale variabelen zijn constantes [final].
10. **Inner classes** zijn klassen gedefinieerd in een andere 'outer' klasse.
  - Een inner klasse kan enkel aangesproken worden door die 'outer' klasse.
  - Een inner klasse heeft toegankelijk tot het object waarbinnen het gemaakt is.
11. **Statische attributen en methodes** bestaan op klasseniveau [static] en worden opgeroepen via de klasse (ClassA.attribuut of ClassA.method (...)).
  - Een statisch attribuut heeft maar 1 waarde per klasse.
  - Een statische methode heeft enkel toegang tot statische attributen van de klasse.
12. Een **package** groepeert klassen [package], die *geïmporteerd* kunnen worden in andere klassen buiten de package [import].
  - Een klasse die je wilt exporteren moet je als *public* declareren in een *file* met dezelfde naam.
  - Binnen een package hebben alle klassen zonder meer toegang tot elkaar.
13. **Visibiliteit** van methodes en attributen:
  - Zonder keyword: vrij toegankelijk binnen package.
  - *public* : voor iedereen toegankelijk.
  - *protected* : enkel toegankelijk voor subclasses.
  - *private* : enkel voor de klasse zelf toegankelijk.

Niet-vermelde details: final methods, naam- en folderstructuur van packages