

## JAVA voor beginners

### Inhoud

Het eerste deel begint met een situering van JAVA als programmeertaal en als platform. Vervolgens wordt de syntaxis van JAVA grondig onder de loep genomen: variabelen, literals, operatoren, programmaverloopstatements en reeksen. Kortom, de basiscomponenten van iedere programmeertaal. De cursus gaat verder met een korte theoretische inleiding in het object georiënteerd programmeren. Deze abstracte principes worden nadien heel praktisch en stapsgewijs uitgewerkt aan de hand van een concreet voorbeeld. Hierbij worden klassen zelf gedefinieerd en wordt er uiteindelijk een complete klasse-hiërarchie stelselmatig opgebouwd. Aan de hand van dit praktisch voorbeeld leert de cursist de mogelijkheden en voordelen van het object georiënteerd programmeren en tevens de wijze waarop dat in JAVA mogelijk gemaakt wordt. Vervolgens worden interfaces en exception handling behandeld om af te sluiten met het gebruik en maken van packages en het genereren van documentatie met de JAVADOC tool.

### Inhoudsopgave

#### Hoofdstuk 1: Inleiding.

- 1.1 Java als programmeertaal
  - 1.1.1 Soorten programmeertalen
  - 1.1.2 Java versus andere programmeertalen
  - 1.1.3 Kenmerken van Java als programmeertaal
- 1.2 Java als platform
- 1.3 Soorten Java toepassingen

#### Hoofdstuk 2: De Java Development Kit.

#### Hoofdstuk 3: Java programma's gebruiken.

#### Hoofdstuk 4: Mijn eerste Java-toepassing.

- 4.1 Het maken van de broncode
- 4.2 Het compileren van de broncode
- 4.3 Het uitvoeren van de bytecode
- 4.4 De opbouw van het programma
  - 4.4.1 Commentaar in Java-code
  - 4.4.2 Het pakket definiëren
  - 4.4.3 Het definiëren van een klasse
  - 4.4.4 De methode main()
  - 4.4.5 Het eigenlijke werk

#### Hoofdstuk 5: De Java programmeertaal.

- 5.1 Variabelen en literals
  - 5.1.1 De declaratie van variabelen
  - 5.1.2 Het data type
  - 5.1.3 Literals
  - 5.1.4 De naam
  - 5.1.5 Final variables
  - 5.1.6 Typeconversie
- 5.2 Operatoren
  - 5.2.1 Rekenkundige operatoren
  - 5.2.2 Relationele operatoren
  - 5.2.3 Logische operatoren
  - 5.2.4 Shift-operatoren
  - 5.2.5 Bit-operatoren
  - 5.2.6 Toekenningsoperatoren
  - 5.2.7 Conditionele operatoren
  - 5.2.8 Overige operatoren
  - 5.2.9 Prioriteitsregels
- 5.3 Uitdrukkingen, statements en blokken

- 5.3.1 Uitdrukkingen
- 5.3.2 Statements
- 5.3.3 Codeblok
- 5.4 Programmaverloop statements
  - 5.4.1 Het while en do while statement
  - 5.4.2 Het for statement
  - 5.4.3 Het if else statement
  - 5.4.4 Het switch statement
  - 5.4.5 Het return statement

## **Hoofdstuk 6: Object georiënteerd programmeren.**

- 6.1 Objecten
  - 6.1.1 Private en publieke eigenschappen
  - 6.1.2 Private en publieke methoden
  - 6.1.3 Voordelen van OOP
- 6.2 Boodschappen
- 6.3 Klassen
- 6.4 Overerving en klasse-hiërarchie
  - 6.4.1 Subklassen en superklassen
  - 6.4.2 Overerving
  - 6.4.3 Klasse-hiërarchie
  - 6.4.4 Abstracte klassen
- 6.5 Interface
- 6.6 Meervormigheid (polymorfisme)
- 6.7 De relatie "IS EEN" versus "HEEFT EEN"

## **Hoofdstuk 7: Werken met objecten .**

- 7.1 Objecten maken
  - 7.1.1 De creatie van het object
  - 7.1.2 De initialisatie van het object
  - 7.1.3 De declaratie en initialisatie van de referentievariabele
  - 7.1.4 Pakketten importeren
- 7.2 Objecten gebruiken
  - 7.2.1 Publieke eigenschappen gebruiken
  - 7.2.2 Publieke methoden gebruiken
- 7.3 Objecten opruimen

## **Hoofdstuk 8: Reeksen.**

- 8.1 Inleiding
- 8.2 Reeksen maken
- 8.3 Reeksen gebruiken
- 8.4 De for-each lus
- 8.5 Reeksen van objecten
- 8.6 Reeksen van reeksen

## **Hoofdstuk 9: Klassen definiëren.**

- 9.1 Inleiding
- 9.2 De declaratie van de klasse
- 9.3 De klasse-omschrijving (body)
  - 9.3.1 Eigenschappen
  - 9.3.2 Methoden
  - 9.3.3 Constructors
  - 9.3.4 Instance-members en class-members
  - 9.3.5 De finalize() methode
- 9.4 Oefeningen 116

## **Hoofdstuk 10: Overerving en klasse-hiërarchie.**

- 10.1 Overerving
  - 10.1.1 Eigenschappen van subklassen
  - 10.1.2 Methoden van subklassen
  - 10.1.3 Constructors van subklassen

- 10.1.4 Klasse-eigenschappen en klasse-methoden
- 10.2 Final klassen en methoden
- 10.3 Abstracte klassen
- 10.4 De superklasse Object
  - 10.4.1 Klasse-hiërarchie
  - 10.4.2 De operator instanceof
  - 10.4.3 Methoden van de Object-klasse
- 10.5 Polymorfisme
- 10.6 Associaties tussen objecten
- Hoofdstuk 11: Eenvoudige objecten.**
  - 11.1 Strings
    - 11.1.1 De String klasse
    - 11.1.2 De variabelen en methoden van de String klasse
    - 11.1.3 De StringBuffer en StringBuilder klassen
    - 11.1.4 Strings samenvoegen met de + operator
  - 11.2 Wrappers voor primitieve datatypes
  - 11.3 De Math klasse
- Hoofdstuk 12: Interfaces.**
  - 12.1 Interfaces versus meervoudige overerving
  - 12.2 Een interface definiëren
    - 12.2.1 De declaratie van de interface
    - 12.2.2 De beschrijving van de interface
  - 12.3 Een interface implementeren in een klasse
  - 12.4 De interface als data-type
- Hoofdstuk 13: Geneste en anonieme klassen.**
  - 13.1 Gewone geneste klassen (inner classes)
  - 13.2 Lokale geneste klassen (local inner classes)
  - 13.3 Anonieme geneste klassen (anonymous inner classes)
  - 13.4 Static geneste klassen (static nested classes)
- Hoofdstuk 14: Het opsommingstype.**
  - 14.1 Inleiding
  - 14.2 Definitie van het opsommingstype
  - 14.3 Eigenschappen, methoden en constructors
- Hoofdstuk 15: Exception handling.**
  - 15.1 Inleiding
  - 15.2 Exceptions afhandelen
    - 15.2.1 Een exception veroorzaken
    - 15.2.2 Een exception opvangen
    - 15.2.3 Meerdere exceptions opvangen
    - 15.2.4 Gemeenschappelijke exception handlers
    - 15.2.5 Het finally blok
  - 15.3 Exceptions genereren
    - 15.3.1 Het throw statement
    - 15.3.2 Exceptions bij vervangen methoden
  - 15.4 Soorten exceptions
    - 15.4.1 Exceptions versus errors
    - 15.4.2 Checked exceptions versus runtime exceptions
  - 15.5 Zelf een exception-klasse maken
- Hoofdstuk 16: Assertions.**
  - 16.1 Inleiding
  - 16.2 Assertions in de code gebruiken 187
  - 16.3 Aanbevolen gebruik van assertions
  - 16.4 Assertions tijdens de uitvoering activeren
  - 16.5 Compatibiliteit met vorige versies
    - 16.5.1 Versies t.e.m. 1.3
    - 16.5.2 Versies vanaf 1.4

## **Hoofdstuk 17: Javadoc.**

- 17.1 Inleiding
- 17.2 Javadoc tags
  - 17.2.1 Documentatie van klassen en interfaces
  - 17.2.2 Documentatie van eigenschappen
  - 17.2.3 Documentatie van methoden en constructors
  - 17.2.4 Documentatie van packages
  - 17.2.5 Overzichtsdocumentatie
- 17.3 Javadoc tool

### **Doelgroep**

Deze cursus is bedoeld voor programmeurs die de syntaxis en ontwikkelomgeving van JAVA willen leren kennen en leren beheersen en een goede basis willen verwerven om zich verder te verdiepen in de mogelijkheden van JAVA. De cursisten moeten reeds enige programmeerervaring hebben. Ze moeten vertrouwd zijn met het gebruik van variabelen, operatoren, programmeeralgoritmen enz...

### **Prijs:**

- 1.375 € (excl. 21% BTW), inclusief de syllabus, broodjes op de middag, koffie, thee en/of andere dranken.
- wanneer een bedrijf of organisatie twee of meerdere deelnemers inschrijft voor dezelfde opleiding wordt vanaf de tweede deelnemer een korting gegeven van 10%.

### **Inschrijven:**

- om in te schrijven bel 09 224 42 17 of mail naar [training@beanpole.be](mailto:training@beanpole.be)
- BeanPole houdt zich het recht voor om opleidingen waarvoor het minimum aantal van drie inschrijvingen niet werd gehaald te annuleren. Annuleringen vanwege BeanPole zullen u minimum 5 werkdagen voor de start van de opleiding doorgegeven worden.
- U mag uw inschrijving annuleren tot 5 werkdagen voor de start van de opleiding. Nadien wordt u het volledig verschuldigde bedrag voor de opleiding aangerekend.

### **Syllabus**

Voor deze cursus wordt gebruik gemaakt van een op maat gemaakte syllabus waarin theorie wordt afgewisseld met praktische oefeningen. De syllabus werd ontwikkeld door Noël Vaes en is tevens apart verkrijgbaar via de website [www.noel-vaes.be](http://www.noel-vaes.be).

### **Taal**

De syllabus en de lessen zijn in het Nederlands.

### **Wanneer:**

Van maandag 14 september 2009 t.e.m. vrijdag 18 september 2009 (5 dagen), van 09:00 tot 17:00

### **Voorkennis**

Enige programmeerervaring: vertrouwd zijn met de basisbegrippen van een programmeertaal en met programmeeralgoritmen.