

## JAVA voor gevorderden

### Inhoud

Dit deel vormt het tweede luik van de cursus JAVA. Er wordt gestart met de Abstract Windows Toolkit waarbij de opbouw van een grafische toepassing in JAVA behandeld wordt. Er wordt hierbij een kleine applicatie gebouwd die doorheen het vervolg van de cursus telkens uitgebreid wordt met nieuwe elementen. Vervolgens worden Collections en de JAVA objecten behandeld voor het lezen en schrijven van o.a. bestanden (streams). Dit wordt gevolgd door een overzicht van de systeem-objecten van de JAVA Runtime Environment. Hierna worden de cursisten ondergedompeld in de wereld die multithreading heet. Er wordt geleerd hoe men in JAVA multithreaded applicaties kan maken en hoe threads gesynchroniseerd kunnen worden. Dit wordt allemaal uitgelegd a.h.v. praktische voorbeelden die tijdens de praktijkoefeningen uitgewerkt en getest worden. De cursus sluit af met een bijzondere vorm van zo'n grafische applicatie: de applet. De levenscyclus van de applet wordt doorlopen evenals de mogelijkheid om vanuit een HTML pagina parameters door te geven. Tevens wordt aandacht besteed aan het gebruik van de JAVA-plugin.

### Inhoudsopgave

#### Hoofdstuk 1: Grafische applicaties in JAVA.

- 1.1 Inleiding
- 1.2 De klasse-hiërarchie van Swing
  - 1.2.1 De Component-klasse
  - 1.2.2 De Container-klasse en subklassen
- 1.3 Mijn eerste GUI-programma
  - 1.3.1 Een nieuw venster maken
  - 1.3.2 Componenten toevoegen
  - 1.3.3 Een menu toevoegen
- 1.4 Layout managers
  - 1.4.1 De Layout Manager
  - 1.4.2 Absolute positionering van componenten
- 1.5 Event-handling
  - 1.5.1 Events afhandelen
  - 1.5.2 Geneste klassen als listeners
  - 1.5.3 Anonieme klassen
  - 1.5.4 Adapters
  - 1.5.5 Het event-object
- 1.6 Dialoogvensters
- 1.7 Zelf tekenen met de klasse Graphics
  - 1.7.1 De methode paintComponent()
  - 1.7.2 De methode repaint()
  - 1.7.3 De klasse Graphics
  - 1.7.4 Werken met afbeeldingen

#### Hoofdstuk 2: Generieken.

- 2.1 Inleiding
- 2.2 Generieke klassen
  - 2.2.1 Generieken definiëren
  - 2.2.2 Het gebruikte type inperken
  - 2.2.3 Generieke types en subtypes
- 2.3 Generieke methoden
  - 2.3.1 Formele generieke parameters 55
  - 2.3.2 Formele generieke parameters met wildcards
  - 2.3.3 Formele generieke parameters met bounded wildcards
  - 2.3.4 Type parameters
- 2.4 Achter de schermen van de generieken
- 2.5 Reeksen en generieken

2.6 Samenwerking tussen oude en nieuwe code

### **Hoofdstuk 3: Collections.**

3.1 Het Collections Framework

3.2 De Collection interface en implementerende klassen

3.2.1 De collection interfaces

3.2.2 De collection klassen

3.3 Itereren over een verzameling

3.4 De Map interface en implementerende klassen

3.4.1 De map interfaces

3.4.2 De map klassen

3.5 Het sorteren van verzamelingen

### **Hoofdstuk 4: Lezen en schrijven (I/O).**

4.1 De File-klasse

4.2 Streams

4.3 Stream-klassen

4.3.1 Character-streams

4.3.2 Byte-streams

4.3.3 Filter-streams

4.4 Random access files

4.5 Object Serialization

4.5.1 Objecten serialiseren en deserialiseren

4.5.2 Klassen serialiseerbaar maken

4.5.3 Transiente variabelen

4.5.4 Het serialisatie-mechanisme aanpassen

4.5.5 Serialisatie en overerving

4.5.6 Versienummering

4.6 Programma attributen

4.7 Gegevens formatteren met de klasse Formatter

### **Hoofdstuk 5: Systeembronnen gebruiken.**

5.1 Inleiding

5.2 De System klasse

5.2.1 Standaard I/O streams

5.2.2 Systeem eigenschappen

5.2.3 Finalization en Garbage Collection

5.2.4 Overige methoden

5.3 Het Runtime object

5.4 De ProcessBuilder

### **Hoofdstuk 6: Multithreading.**

6.1 Inleiding: multiprocessing en multithreading

6.2 Een nieuwe thread creëren

6.2.1 Subklasse van de klasse Thread

6.2.2 De interface Runnable

6.3 De levenscyclus van threads

6.4 Prioriteiten van threads

6.4.1 De scheduler

6.4.2 Coöperatieve en preëmptieve multitasking

6.4.3 Prioriteiten instellen

6.5 Daemon threads

6.6 De wachttoestand

6.6.1 De slaaptoestand

6.6.2 De geblokkeerde toestand 133

6.7 Wachten op de beëindiging van een thread

6.8 Synchronisatie van threads (monitoring)

6.8.1 Object locking

6.8.2 Wait() en notify()

6.9 De Timer-klasse en de TimerTask-klasse

6.10 Grafische animaties

### **Hoofdstuk 7: Java Archive bestanden (JAR).**

7.1 Basisprincipes van JAR

7.1.1 Een JAR-bestand maken

7.1.2 De inhoud van een JAR-bestand bekijken

7.1.3 Een JAR-bestand uitpakken

7.1.4 Een JAR-bestand opnemen in het classpath

7.2 JAR-bestanden als applicaties gebruiken

7.2.1 Het manifest-bestand

### **Hoofdstuk 8: Applets.**

8.1 Inleiding

8.2 De JApplet-klasse

8.2.1 Het maken van de broncode

8.2.2 Een HTML-document voor een applet

8.2.3 De applet uitvoeren

8.3 De levenscyclus van een applet

8.3.1 De initialisatie van de applet

8.3.2 Het starten van de applet

8.3.3 Het stoppen van de applet

8.3.4 Het opruimen van de applet

8.4 De applet-API

8.4.1 Parameters

8.4.2 Bestanden laden en lezen

8.4.3 Geluiden afspelen in een applet

8.4.4 Interactie met de browser

8.5 Rechtstreeks met de server communiceren

8.5.1 Gegevens lezen van een URL

8.5.2 Gegevens schrijven naar een URL

8.6 Het gebruik van JAR-bestanden

8.7 Beperkingen van Applets

8.8 De JAVA Plugin

### **Doelgroep**

Deze cursus is bedoeld voor programmeurs die zich verder willen verdiepen in de programmeertaal JAVA. De cursus vormt tevens een goede basis voor mensen die zich willen voorbereiden op het examen SUN™ Certified Programmer for the JAVA2 Platform.

### **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.

### **Duur**

5 dagen.

### **Voorkennis**

De basiscursus gevolgd hebben of een gelijkaardige kennis verworven hebben.