

INHOUD

| | |
|---|-----------|
| HOOFDSTUK 1: INLEIDING | 9 |
| 1.1 Wat kun je met Java doen? | 9 |
| 1.2 Over Java | 10 |
| 1.3 Gebruik van dit boek | 11 |
| 1.4 Installatie | 12 |
| 1.5 Het eerste programma | 16 |
| 1.6 Vaak voorkomende problemen | 18 |
| 1.6.1 Invalid jdkhome specified | 19 |
| 1.6.2 Het indrukken van de F6-toets heeft geen effect | 19 |
| 1.6.3 Eén of meer projecten ontwikkelden fouten tijdens de compilatie | 19 |
| 1.7 Ontleding van het eerste programma | 19 |
| | |
| HOOFDSTUK 2: OBJECTGEORIËNTEERD PROGRAMMEREN | 23 |
| 2.1 Abstractie | 23 |
| 2.2 Klasse | 23 |
| 2.3 Attribuut | 25 |
| 2.3.1 Type | 25 |
| 2.3.2 Een attribuut declareren | 26 |
| 2.3.3 Een attribuut initialiseren | 27 |
| 2.4 Methode | 27 |
| 2.4.1 Definitie | 27 |
| 2.4.2 De leden van een object benoemen | 29 |
| 2.4.3 Parameter | 30 |
| 2.4.4 Teruggeefwaarde | 30 |
| 2.4.5 Overladen | 32 |
| 2.5 Objecten construeren | 32 |
| 2.6 Statische attributen en methodes | 35 |
| 2.7 Primitieve types en referentietypes | 37 |
| | |
| HOOFDSTUK 3: BASISCONSTRUCTIES | 39 |
| 3.1 Opdrachten | 39 |
| 3.2 Veranderlijken | 39 |
| 3.3 Toekenningsopdrachten | 40 |
| 3.3.1 Uitdrukking | 41 |
| 3.3.2 Initialisatie van een veranderlijke of een attribuut | 41 |
| 3.3.3 Afgekorte toekenning | 42 |
| 3.3.4 Ophoging en verlaging | 42 |
| 3.4 Bewerkingen | 43 |
| 3.4.1 Rekenkundige bewerkingen | 43 |
| 3.4.2 Samenstelling van tekststrengen | 44 |
| 3.4.3 Andere bewerkingen | 45 |
| 3.5 Invoer langs het toetsenbord | 46 |
| 3.6 Uitdrukking | 49 |
| 3.7 De uitvoering van een Java-programma | 52 |
| | |
| HOOFDSTUK 4: CONTROLESTRUCTUREN | 55 |
| 4.1 Voorwaardelijke opdrachten | 55 |
| 4.1.1 Vergelijkingen | 56 |
| 4.1.2 Logische bewerkingen | 58 |
| 4.1.3 Algemene vorm van de voorwaardelijke opdracht | 60 |
| 4.1.4 Voorwaardelijke uitdrukking | 61 |
| 4.1.5 Keuzeopdracht | 62 |
| 4.2 Herhalingen | 64 |
| 4.2.1 Lusopdracht 'while' | 64 |
| 4.2.2 Genestelde lussen | 67 |

| | | |
|---|---|------------|
| 4.2.3 | Rij | 68 |
| 4.2.4 | for | 70 |
| 4.2.5 | do/while | 72 |
| 4.2.6 | Een iteratie onderbreken | 75 |
| 4.2.7 | De lusopdracht afbreken | 75 |
| 4.2.8 | Labels | 76 |
| 4.2.9 | for each-lussen | 77 |
| 4.3 | Recursie | 78 |
| HOOFDSTUK 5: OVERERVING | | 81 |
| 5.1 | Subtypes | 81 |
| 5.2 | Overerving van attributen en methodes | 84 |
| 5.3 | Een geërfde methode onderdrukken | 87 |
| 5.4 | Abstracte klassen en abstracte methodes | 90 |
| 5.5 | Interface | 92 |
| 5.6 | De bijzondere rol van de klasse Object | 95 |
| HOOFDSTUK 6: PAKKETTEN, AFSCHERMING EN GROEPERIG | | 97 |
| 6.1 | De klassenbibliotheek | 97 |
| 6.2 | Gebruik van pakketten | 99 |
| 6.3 | Werken met tekstveranderlijken | 102 |
| 6.3.1 | De klasse java.lang.String | 102 |
| 6.3.2 | StringBuffer | 111 |
| 6.4 | Werken met bestanden: het pakket java.io | 112 |
| 6.4.1 | Sequentiële bestanden versus willekeurige toegang | 112 |
| 6.4.2 | Wat is de kleinste eenheid van informatie? | 113 |
| 6.4.3 | Afgeleide streams | 116 |
| 6.4.4 | De standaard-bytestromen van het systeem | 117 |
| 6.4.5 | Geheugenstromen | 118 |
| 6.5 | Toegangsbeperking | 120 |
| 6.6 | Inwendige klassen | 122 |
| 6.6.1 | Statische inwendige klassen | 124 |
| 6.6.2 | Gewone inwendige klassen | 125 |
| 6.6.3 | Anonieme klassen | 127 |
| 6.7 | Bereik, zichtbaarheid, levensduur en toegankelijkheid | 127 |
| HOOFDSTUK 7: KWALITEIT | | 131 |
| 7.1 | Uitzonderingen | 131 |
| 7.1.1 | Paranoia | 131 |
| 7.1.2 | Definitie | 133 |
| 7.1.3 | Exceptions doorgeven | 133 |
| 7.1.4 | Exceptions opvangen | 135 |
| 7.1.5 | Exceptions genereren | 136 |
| 7.1.6 | Exceptions definiëren | 137 |
| 7.2 | Commentaar | 139 |
| 7.3 | De documentatiegenerator Javadoc | 140 |
| 7.4 | Levenscyclus systeemontwikkeling | 143 |
| 7.5 | Domeinklassen | 145 |
| 7.6 | Technisch ontwerp | 147 |
| 7.6.1 | Architectuur | 147 |
| 7.6.2 | Ontwerppatronen | 151 |
| 7.6.3 | Voorbeeld van een patroon: factory method | 151 |
| 7.6.4 | Voorbeeld van een patroon: adapter | 152 |
| 7.6.5 | Frameworks | 155 |
| 7.7 | Moduletest met JUnit | 155 |
| 7.8 | Asserties | 160 |

| | |
|--|------------|
| HOOFDSTUK 8: BIJZONDERE PROGRAMMEERTECHNIEKEN | 163 |
| 8.1 Annotaties | 163 |
| 8.1.1 Definitie en gebruik | 163 |
| 8.1.2 Zelf een annotatietype declareren | 164 |
| 8.2 Abstracte gegevenstypes | 165 |
| 8.3 Generische types | 168 |
| 8.4 Collecties | 170 |
| | |
| HOOFDSTUK 9: WEBTOEPASSINGEN | 177 |
| 9.1 De technologie van het web | 177 |
| 9.1.1 De context: web servers en browsers | 177 |
| 9.1.2 HTML | 179 |
| 9.2 Servlet | 181 |
| 9.2.1 Definitie en voorbeeld | 181 |
| 9.2.2 Informatie doorgeven met formulieren | 185 |
| 9.2.3 De levenscyclus van een servlet | 187 |
| 9.3 JavaServer Pages (JSP) | 191 |
| 9.4 Sessie | 193 |
| | |
| HOOFDSTUK 10: GRAFISCHE GEBRUIKERSINTERFACES | 195 |
| 10.1 Swing | 195 |
| 10.2 Opbouw van een interface | 195 |
| 10.3 De opmaak beheersen | 201 |
| 10.3.1 FlowLayout | 202 |
| 10.3.2 BorderLayout | 202 |
| 10.3.3 GridLayout | 203 |
| 10.3.4 GridBagLayout | 205 |
| 10.3.5 Opmaakbeheerders combineren | 208 |
| 10.4 Gebeurtenissen afhandelen | 210 |
| 10.5 Menu | 215 |
| 10.6 Teken en | 218 |
| 10.7 Applets | 221 |
| 10.7.1 Definitie en voorbeeld | 221 |
| 10.7.2 Het leven van een applet | 222 |
| 10.8 Model/View/Controller | 225 |
| 10.9 Tabellen en bomen weergeven | 231 |
| 10.9.1 Tabellen | 231 |
| 10.9.2 Bomen | 234 |
| | |
| HOOFDSTUK 11: HERBRUIKBARE COMPONENTEN (BEANS) | 239 |
| 11.1 Begrippen | 239 |
| 11.2 Een bestaande Bean gebruiken in een toepassing | 240 |
| 11.3 Archieven | 241 |
| 11.3.1 Wat is een archief? | 241 |
| 11.3.2 Manifest | 242 |
| 11.4 Zelf een Bean ontwikkelen | 244 |
| 11.5 Eigenschap | 245 |
| 11.6 Gebeurtenis | 246 |
| 11.7 Gebonden eigenschappen | 249 |
| 11.8 Beperkte eigenschappen | 251 |
| 11.9 BeanInfo | 253 |
| | |
| HOOFDSTUK 12: PARALLEL PROGRAMMEREN | 255 |
| 12.1 Begrippen | 255 |
| 12.2 Timers | 255 |
| 12.3 Animatie | 257 |
| 12.4 De klasse <code>Thread</code> en de interface <code>Runnable</code> | 259 |
| 12.5 Prioriteit en activering | 260 |

| | | |
|---|---|------------|
| 12.6 | Synchronisatie | 264 |
| 12.6.1 | join | 264 |
| 12.6.2 | synchronized | 265 |
| 12.6.3 | wait en notify | 269 |
| HOOFDSTUK 13: RELATIONELE GEGEVENS BANKEN | | 273 |
| 13.1 | Platformonafhankelijke databasetaal | 273 |
| 13.2 | Platformonafhankelijke verbinding | 274 |
| 13.2.1 | JDBC | 274 |
| 13.2.2 | Voorbeeld van een ODBC-verbinding | 274 |
| 13.2.3 | Voorbeeld van een Derby-verbinding | 276 |
| 13.3 | Een gegevensbank bevragen | 278 |
| 13.4 | De inhoud van een gegevensbank wijzigen | 281 |
| 13.5 | Transactie | 283 |
| 13.6 | Gevorderde technieken | 286 |
| 13.6.1 | Prepared statement | 286 |
| 13.6.2 | Gegroepeerde updates | 287 |
| 13.6.3 | Stored procedures | 288 |
| 13.7 | Object-relational mapping | 289 |
| 13.7.1 | Persistente objecten | 289 |
| 13.7.2 | Primaire sleutel | 290 |
| 13.7.3 | Een kennismaking met Hibernate | 291 |
| HOOFDSTUK 14: GEDISTRIBUEERDE BEDRIJFSTOEPASSINGEN | | 295 |
| 14.1 | De noodzaak van gedistribueerd programmeren | 295 |
| 14.2 | Een webservice bouwen | 297 |
| 14.3 | De logica van een methodeaanroep | 304 |
| 14.4 | Probleem: afhankelijkheden beperken | 307 |
| 14.5 | Enterprise JavaBeans | 307 |
| 14.5.1 | Wat is EJB, en waarom heb ik EJB nodig? | 307 |
| 14.5.2 | Soorten enterprise beans | 308 |
| 14.5.3 | Toegang tot beans | 309 |
| 14.5.4 | Parameters en teruggeefwaarden | 310 |
| 14.5.5 | Voorbeeld van een session bean | 311 |
| 14.5.6 | Voorbeeld van remote access | 314 |
| 14.6 | Spring | 316 |
| 14.6.1 | Inversion of Control | 316 |
| 14.6.2 | Een elementair voorbeeld | 317 |
| HOOFDSTUK 15: JAVA OP DE GSM | | 327 |
| 15.1 | JME: schaarse hulpbronnen | 327 |
| 15.2 | Een simulator voor onze gsm-omgeving | 327 |
| 15.3 | Een eerste voorbeeldprogramma | 328 |
| 15.4 | Een MIDlet overbrengen naar een gsm | 335 |
| 15.5 | Een spel | 336 |
| BIBLIOGRAFIE | | 345 |
| TREFWOORDENREGISTER | | 347 |