**The bunny massacre**

**Thema I/TI 1.2 De geprogrammeerde applicatie**



**Super coole awesome adviesbureau**

**The bunny massacre**

**Thema I/TI 1.2 De geprogrammeerde applicatie**

*Auteurs: Jesse Koppelaar (281353)*

*Marnix Manuel (335008)*

*Marcel Stoepker (337590)*

*Robert Ziengs (332174)*

*Klas: PIV1C*

*Projectnaam: Vossen en Konijnen*

*Studierichting: hbo-ict*

*Schoolnaam: Hanzehogeschool Groningen (SCMI)*

*Docenten: Wieteke van Veen (VWIE)*

*Henkjan Hekman (HEHJ)*

*Datum: 27-01-2015*

*Plaats: van Doornveste*

**Voorwoord**

Wij hebben de opdracht van de Hanze Hogeschool te Groningen gekregen dit rapport > niet direct verwijzen, een adviesrapport te schrijven over de verbetering van de applicatie Vossen en Konijnen. Onze opleiding valt onder het Instituut voor Communicatie, Media & IT afdeling HBO-ICT.

De doelgroepen van dit rapport zijn de gebruikers en de eigenaar van de applicatie Vossen en Konijnen. Daaronder vallen de docenten van de ICT opleiding op de Hanze Hogeschool en onze projectgroep.

We willen als team graag onze docenten bedanken die hebben geholpen met het programmeerwerk van het project. Ook willen we de docenten bedanken die hebben geholpen met de verslaglegging van dit project. Als laatste > Tot slot willen we Danny Pijlman bedanken voor het meewerken aan ons project in de eerste week.

Groningen, 27-01-2015

Marnix Manuel

Robert Ziengs

Jesse Koppelaar

Marcel Stoepker

**Samenvatting > Inleiden, onderwerp introduceren.**

Eén van de belangrijkste verbeteringen in het programma is het aanmaken van de “mainclass”, deze verbeteringen zullen ervoor zorgen dat het programma geautomatiseerd wordt. Het is erg belangrijk dat er een overzichtelijke GUI wordt gemaakt waardoor de gebruiker met allerlei knoppen de instellingen van de simulatie kan aanpassen. Er zullen ook slangen en wolven aan het arsenaal van dieren toegevoegd worden, net als andere objecten zoals de “man-bear-pig” en de jagers. Gras wordt toegevoegd als voedselbron voor de konijnen. Om te zorgen dat de instellingen in balans zijn, hebben we bij het konijn ingesteld dat het alleen consumeert als het onder een bepaald voedselniveau zit.

> Herschrijven.

> Je schrijft het verslag achteraf.

> Let op alinea-indeling.

> Vermijd het gebruik van zullen en willen.

> Beschrijf stap voor stap wat je hebt gedaan.

> Wees concreet.

**Inhoudsopgave**

Hoofdstuk 1 - Inleiding 5

Hoofdstuk 2 - Probleemanalyse 6

2.1 Probleemstelling 7

2.2Analyseren 7

2.3Beperkingen 7

Hoofdstuk 3 - Uitbreidingen ?

3.1 De mainclass ?

3.2 De GUI ?

3.2.1Uitstraling ?

3.2.2Mogelijkheden ?

3.3 Het gras ?

3.4 Actors ?

3.4.1 Jager ?

3.4.2 Man-Bear-Pig ?

Jaagcyclus ?

Konijnen ?

Vossen ?

Slangen ?

Man-Bear-Pig ?

Jager ?

Hoofdstuk 4 - Conclusie 8

Literatuurlijst 9

UML Klassendiagram versie 1 ?

UML Klassendiagram versie 2 ?

UML Sequencediagram ?

Reflecties ?

Jesse ?

Marnix ?

Marcel ?

Robert ?

**Check de sheets en Schrijfwijzer Techniek.**

**Inleiding**

Het programma Vossen en Konijnen is een programma dat een omgeving waar vossen en konijnen in zitten simuleert. Het programma bootst na wat er gebeurt als je een aantal vossen bij konijnen zet in een gesloten reservaat.

Mijd de je-vorm!!

Mijd de we-vorm.

Wie is de opdrachtgever?

Het doel is om de applicatie te verbeteren, in opdracht van de opdrachtgever. De eisen/wensen van de opdrachtgever zijn...

Wat verstaan jullie onder gebruikersvriendelijk en overzichtelijk? Begrippen kort definiëren, dat mag ook in een volgend hoofdstuk, als je er hier maar naar verwijst.

Het doel van het project is om de simulatie zo uitgebreid en zo gebruiksvriendelijk mogelijk te maken. Een belangrijk aspect van gebruiksvriendelijkheid is dat het programma overzichtelijk is. Dit betekent onder andere dat de resultaten van de simulatie overzichtelijk moeten worden.

In de komende hoofdstukken gaan we het hebben over de problemen die we tegengekomen zijn bij het programma “vossen en konijnen”. Met als eerste hoofdstuk de probleemanalyse waarin wordt omschreven wat de problemen zijn waar ons team tegenaan gelopen is.

In de inleiding hoort te staan:

> aanleiding + beschrijving onderwerp (de opdrachtgever Pet & Co (fictieve naam) heeft de wens de applicatie van.. te verbeteren, heeft behoefte aan...;

> probleemstelling (opdracht/vraagstelling, eventuele deelvragen, doelstelling);

> afbakening/definiëring gebruikte begrippen;

> vooruitblik.

Hoofdstuk 1 beschrijft.... IN hoofdstuk 2 komt ... aan de orde... Daarna behandelt hoofdstuk 3 .... Vervolgens komt in hoofdstuk 4.... aan bod enz. Het adviesrapport sluit af met een conclusie en een advies aan de opdrachtgever o.i.d.

**Probleemanalyse > probleemstelling kort naar de inleiding. Eventueel in het eerste hoofdstuk uitgebreider behandelen, bijv. met de operationalisering van de begrippen.**

**Mijd de we-vorm. Elk hoofdstuk een titel geven + beginnen met een inleiding. Wat komt er in dit hoofdstuk aan de orde? Wat wordt achtereenvolgens behandeld? Waarom komt dit aan de orde (gezien de opdracht)? Elk nieuw onderwerp inleiden! Ook (sub)paragrafen maken, met een decimale indeling. Elk hoofdstuk eindigen met een slotpassage. Welke problemen/welke vragen zijn er in dit hoofdstuk opgelost en hoe? Tekst is nu te kort door de bocht, ik begrijp het niet. Onduidelijk wat jullie bedoelen. Deel de tekst op in deelonderwerpen en maak daar aparte hoofdstukken van. Maak alles concreet en licht het toe met korte voorbeelden.**

**Hoofdstuk 2 - Probleemanalyse**

In dit hoofdstuk zal de probleemanalyse worden besproken. Ten eerste zal er wat verteld worden over probleemstelling van de applicatie. Daarna zal de applicatie uitgebreid geanalyseerd worden en tot slot zullen alle berperkingen on een rijtje worden gezet.

**2.1 Probleemstelling**

De applicatie Vossen en Konijnen simuleerd een eet cyclus van een vos en een konijn. Deze eetcyclus is nog maar op kleine schaal vergeleken met het echt. In de natuur zijn namelijk veel meer verschillende factoren die meespelen op het beweeg en jaag gedrag van bijvoorbeeld een vos. En in de natuur zijn natuurlijk ook niet alleen maar vossen en konijnen maar veel meer verschillende soorten roofdieren en prooien. Doormiddel van de Vossen en Konijnen simulatie kan er een klein stukje van de natuur ..... worden nagebootst. Hierdoor kunnen onderanderen natuurdeskundigen als boswachters voordeel aan hebben voor bijvoorbeeld onderzoek en leefbewegingen van dieren.

**2.2 Analyseren**

De applicatie Vossen en Konijnen bestaat uit twee verschillende kanten namelijk de kant voor de klanten en de kant voor de programmeurs. De buiten kant ofterwijl de kant voor de klanten bestaat uit een klantvriendelijke interface(GUI) waar een goed en overzichtelijk pagina op af wordt gebeeld. En de binnenkant bestaat eigenlijk alleen maar uit programmeer taal(code).

Wanneer de simulatie wordt gestart komt er een pagina naar voren met daarop een veld van 50 bij 50 vakken met hierin twee verschillende kleuren namelijk geel en blauw. De gele blokjes stellen de konijnen voor en de blauwe vakken de vossen. Als deze simulatie wordt gestart bewegen de blauwe en gele blokjes alle kanten op. De vossen en konijnen zoeken hier namelijk een lege plek om de volgende stap op te nemen. Naar mate de simulatie langer loopt blijkt dat er steeds minder konijnen overblijven sinds het begin. Dit komt namelijk doordat de vossen op de konijnen jagen om te overleven.

**2.3 Beperkingen**

Er zitten nog genoeg beperkingen in de simulatie. Zo kan de gebruiken van de applicatie geen eigen invoer in te stellen en dus ook niet hun eigen inbreng in de applicatie toevoegen. Ook zijn er nog niet genoeg dieren waardoor de simulatie ongeloofwaardig wordt en niet te vergelijken valt met de natuur. Ook zullen er later in de simulatie niet-dieren in komen, hierbij kunt u bijvoorbeeld aan een jager denken of een fictioneel monster wat niet onder dieren valt.

Een cruciale beperking aan de applicatie is dat er nog geen GUI in is geimplementeerd waardoor de hele applicatie niet kan worden uitgevoerd. Ook zijn er nog verschillende settings van de konijnen en vossen die nog niet volledig zijn uit gewerkt. Zo eten de konijnen niks en kunnen alleen dood gaan aan ouderdom of opgegeten worden door vossen. En zit er geen limiet aan de hoeveelheid wat de vossen opkunnen en blijven hierdoor de hele simulatie door alle konijnen opeten.

**Hoofdstuk 3 - Uibreidingen**

In dit hoofdstuk zullen alle uitbreidingen die later toegevoegd zijn worden besproken.

**3.1 De mainclass**

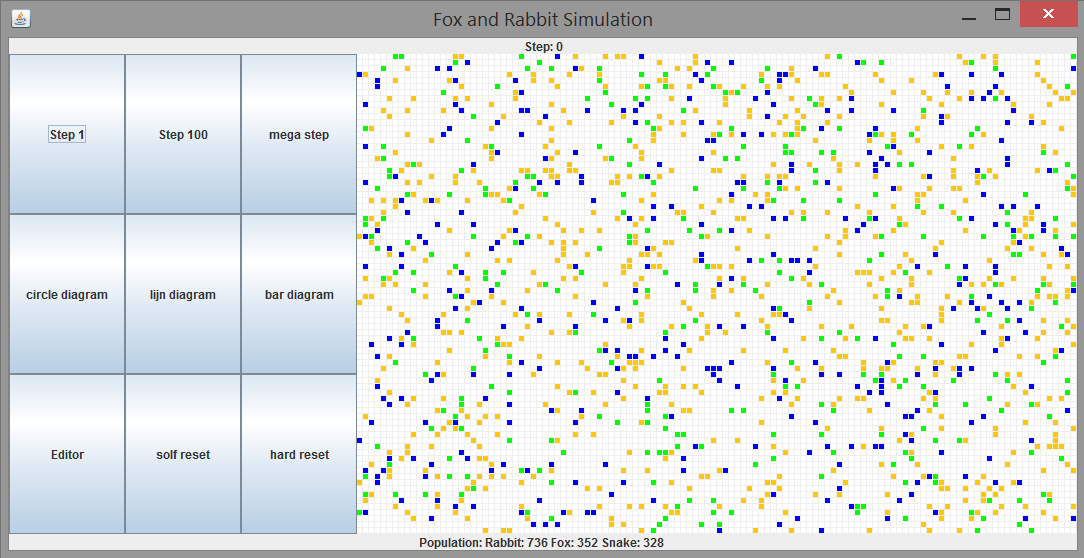
De eerste uitbreiding van de simulatie is de mainclass. Voor dat de hele simulatie zal gaan werken moet de mailclass worden aangemaakt. Een main class is het hart van het programma waar in staat welke objecten aangemaakt moeten worden en wat op het aangemaakte object voor “methods” uitgevoerd moet worden. *Voorbeeld: De main class maakt een auto aan en laat daar de methode rijden op uitvoeren.* Uitleg: Het object wat aangemaakt is “de auto”. De methode die uitgevoerd wordt op de auto is “rijden”. Dit zorgt ervoor dat, bij de randvoorwaarden die jij geeft in de mainclass, de auto **automatisch** gaat rijden.

**3.2 De GUI**

Voor de verbetering van de invloed van de gebruiker moet er een GUI aangemaakt worden. Een GUI is een overzichtelijke interface waarop alle mogelijke toepassingen van de simulatie weergegeven staan. Hierdoor kan de gebruiker makkelijk zijn aanpassingen toepassen. Zo kan de gebruiker bijvoorbeeld snel de breeding snelheid van een konijn aanpassen waardoor deze zich sneller voortplant. Voor het menu van de GUI hebben we voor een “sliders” gekozen zodat er toch limieten aan de veranderingen zitten. Dat zorgt ervoor dat de gebruiker de simulatie niet zodanig kan aanpassen dat de simulatie uit balans raakt.

**3.2.1 Uistraling**

Om een GUI zo klantvriendelijk mogelijk te maken is het van belang dat de interface van de GUI zo makkelijk en overzichtelijk mogelijk is.

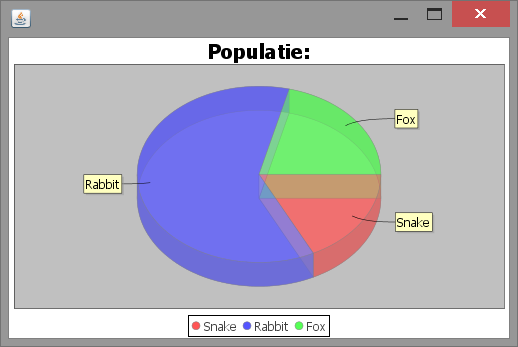


Zoals op de afbeelding te zien is is het zeer overzichtelijk.

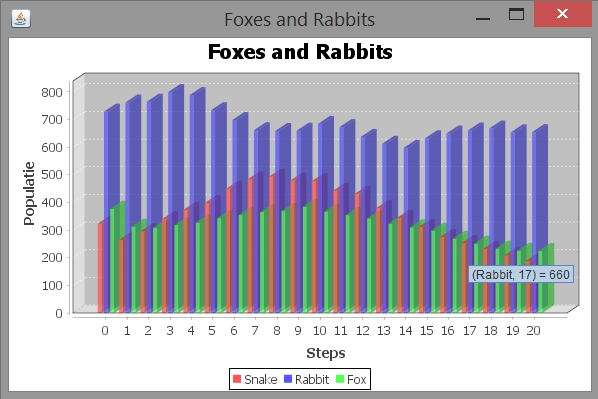
**3.2.2 Mogelijkheden**

Zoals op de afbeelding te zien is zijn er veel mogelijkheden. Zo heb je de eerste rij knoppen die bestaat uit “step 1”, “step 100” en “mega step”. Deze knoppen zorgen ervoor dat de simulatie 1, 100 of 4000 stappen simuleert.

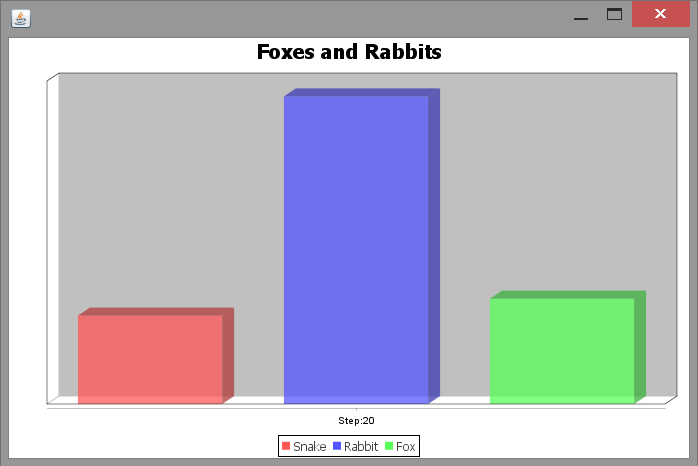
De tweede rij met knoppen bestaat uit circle diagram, Histogram en bar diagram. Elke knop leest de gegevens van de huidige stap uit en weergeeft ze in een diagram weer.



Cirkel diagram

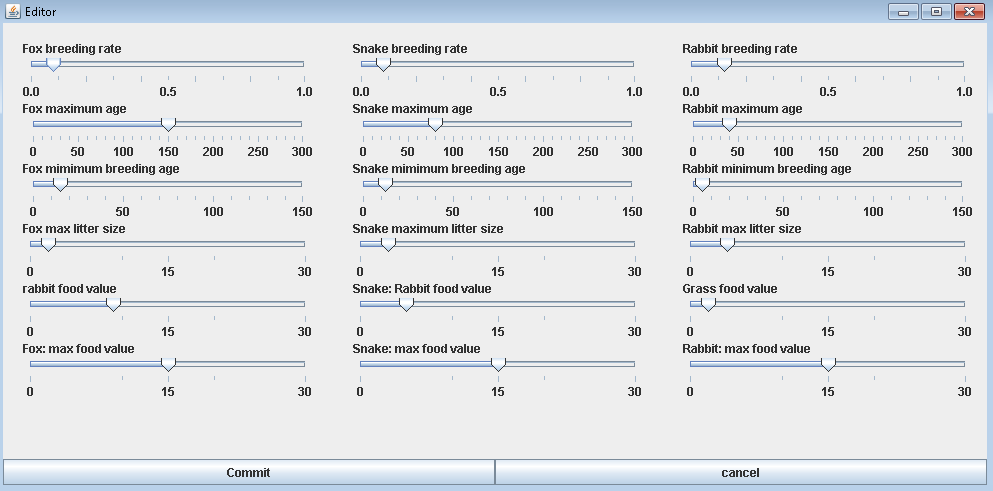


Histogram



Bar diagram

En als laatste worden de knoppen “Editor”, “Soft Reset” en “Hard Reset” weergegeven. Bij de knop “Editor” is het mogelijk om de instellingen van ieder dier aan te passen. Doordat deze optie is toegevoegd is er een extra reset knop gekomen. Deze knop moest er bij komen omdat anders de veranderde settings blijven staan op deze instellingen. De soft reset knop reset alleen de simulatie en de hard reset knop reset de simulatie en de instellingen naar hun oorspronkelijke instellingen.



Editor

**Jaagcyclus**

In de simulatie zijn verschillende extra dieren en niet-dieren toegevoegd waardoor de jaagpatroon ook aangepast moest worden. Bijna elk dier en niet-dier heeft nu een jaag-objective. In de subhoofdstukken worden voor elk dier en niet-dier hun jaagpatroon verklaard.

**Konijnen**

Konijnen zijn de enige dieren in de hele simulatie die niet jagen op andere dieren. Om de overlevings factor van een konijn te verlagen hebben wij gras geïmplementeerd. Konijnen kunnen na deze implementatie verhongeren en hier aan dood gaan.

**Vossen**

De vossen zijn de eerste dieren die we tegen kwamen in de simulatie die jaagden op een andere diersoort. De vossen jaagden namelijk op konijnen en dit doen ze nog steeds. Ook zijn er verschillende methodes en ... toegepast. Zo kan de vos niet alleen dood gaan aan

**Slangen**

Slangen is het eerste en het enige extra dier die is toegevoegd aan de simulatie. De Snake-klasse is zeer goed te vergelijken met de Fox-klasse niet alleen omdat.... maar ook omdat ... De jaag objective van de slangen zijn net zoals de vossen de konijnen.

**Man-Bear-Pig**

Eén van de twee niet-dieren is de Man-Bear-Pig, Dit fictionele monster refereert naar het televisieprogramma “Southpark” waar in een aflevering Al Gore jaagt op dit wezen.De Man-Bear-Pig zal zoals geen ander alles opeten wat op zijn pad komt en is hierdoor een interessante implementatie aan de simulatie.

**Jager**

**Toekomstige verbeteringen**

Betere pathfinding jager

Dynamische grafieken

Eén van de eerste problemen die we hebben opgemerkt in het huidige programma is dat de applicatie nog handmatig aangestuurd moet worden. Dit zal moeten worden geautomatiseerd. Dit kan door middel van het aanmaken van een “main class”. Een main class is het hart van het programma waar in staat welke objecten aangemaakt moeten worden en wat op het aangemaakte object voor “methods” uitgevoerd moet worden. *Voorbeeld: De main class maakt een auto aan en laat daar de methode rijden op uitvoeren.* Uitleg: Het object wat aangemaakt is “de auto”. De methode die uitgevoerd wordt op de auto is “rijden”. Dit zorgt ervoor dat, bij de randvoorwaarden die jij geeft in de mainclass, de auto **automatisch** gaat rijden. > meer alinea's

Voor de verbetering van de invloed van de gebruiker moet er een GUI aangemaakt worden. Dit zorgt ervoor dat er knoppen toegevoegd kunnen worden door onze programmeurs. Knoppen die er voor zorgen dat de gebruiker van het programma dingen ??? term operationaliseren kan kiezen die het programma gaat uitvoeren.

*Voorbeeld: In kader van dit programma kan de gebruiker bijvoorbeeld kiezen om de maximale leeftijd van de wolf omhoog te brengen.* Voor het menu van de GUI hebben we voor een “pulldown menu” gekozen zodat er toch limieten aan de veranderingen zitten. Dat zorgt ervoor dat de gebruiker de simulatie niet zodanig kan aanpassen dat de simulatie uit balans raakt.

Gebruik tussenkopjes.

Voor de uitbreiding van het programma zullen we het monster “man-bear-pig” toevoegen aan het programma. Dit monster refereert naar het televisieprogramma “Southpark” waar in een aflevering Al Gore jaagt op dit wezen.Het monster zal alles opeten wat het op zijn pad tegenkomt. Dit is volgens ons team een leuke toevoeging in de simulatie. Voor verdere uitbreiding van het programma zullen er extra dieren aan toe gevoegd moeten worden. In ons geval zijn dit slangen en wolven met ieder hun specifieke eigenschappen. De slangen zullen konijnen en wolven opeten. De wolven zullen konijnen en vossen opeten. De vossen zullen konijnen en slangen opeten.

Het konijn is de enige herbivoor in de simulatie, wat inhoudt dat dit dier niet op andere dieren gaat jagen. Voor dit dier zal wel een voedselbron aangemaakt moeten worden. Hiervoor worden op locaties in de simulatie gras toegevoegd. Het is van groot belang dat de variabelen aangemaakt worden bij deze dieren en in balans zijn. Dat wil zeggen, de instellingen moeten niet zo zijn, dat één van de diersoorten in een aantal seconden is uitgestorven.

Eén van de instellingen van de dieren is het voedselniveau. Deze zal bij de “superclass” dier terecht komen. Een superclass is een blauwdruk van de klassen die er onder vallen. De klassen die daar onder vallen worden “subclasses” genoemd en zal alle methodes overnemen van de superclass. Om te zorgen dat de instellingen in balans zijn, hebben we de keuze gemaakt dat konijnen alleen zullen consumeren als ze onder een bepaald voedselniveau zitten.??????

**Conclusie**

We zijn tot de conclusie gekomen dat we bij het programmeren en verbeteren van een simulatie met meerdere factoren rekening moeten houden. Er moet een goed evenwicht zijn in het voedsel en de omgeving van de dieren. Ook zijn er wat extra factoren zoals "man-bear-pig" teweeg gebracht, met de reden dat het gebruik van de applicatie wat leuker wordt. Er moest voor gezorgd worden, dat het bedienen van het programma makkelijker werd. Door het installeren van een pulldown menu is het toevoegen van de limieten van de gebruiker gelukt. De limieten zijn ter preventie van een onevenwichtig balans in de populatie van de diersoorten. Het pulldown menu is erg gebruiksvriendelijk, het is namelijk erg overzichtelijk.

Let op alinea-indeling. Kruip in de huid van een fictief ICT-bureau en formuleer een professioneel advies richting opdrachtgever. Let op: maak steeds een tweedeling tussen het probleem en de oplossing.

De conclusie koppelt terug naar de inleiding (geeft een antwoord op de vraag of de oplossing voor het probleem zoals die in de inleiding geformuleerd is/zijn), inclusief een terugblik. Hoe ben je tot deze conclusie gekomen? Hoe ben je tot deze oplossing gekomen? Hoe ben je tot dit advies gekomen? Check de checklist en de sheets!

**Bronnen > Literatuurlijst > zie de Schrijfwijzer Techniek voor de officiële notatie.**

*Sites:*

[www.stackoverflow.com](http://www.stackoverflow.com/)

http://docs.oracle.com

*Literatuurlijst:*

*Barnes, David J. & Kölling, Michael (2012) . Programmeren in JAVA met BLUEJ(5e druk), Amsterdam: Pearson Benelux*

*Barnes, David J. & Kölling, Michael (2009) . Programmeren in JAVA met BLUEJ(4e druk), Amsterdam: Pearson Benelux*