

Nascholing

Algoritmisch Denken



Martin Bruggink
Renske Smetsers





Petals Around the Rose



Tip: bloemblaadjes rond de roos

Petals Around the Rose



Tip: bloemblaadjes rond de roos: **ABSTRACTIE!!**

Speerpunten bijeenkomst #3

- Afrondend bijeenkomst
- Terugkoppeling & reflectie
 - Sorteren
 - Challenges (lesmateriaal)
- Verdere kennismaking met Alg. Denken
 - Puzzel opdracht
 - AD module
- Wat verder ter tafel komt

Terugblik presentatie Erik Barendsen

- Comp Thinking vaardigheden
- Gebruik symbolische representatie zoals flowcharts
- Traceren van code
- In eigen woorden samenvattend uitleggen wat een programma doet
- Beverwedstrijd als voorbeelden van toetsing 'algoritmisch denken'.

Nieuw examenprogramma, grondslagen

Algoritmen

De kandidaat kan

- een oplossingsrichting voor een probleem uitwerken tot een algoritme;
- daarbij standaardalgoritmen herkennen en gebruiken;
- de correctheid en efficiëntie van digitale artefacten onderzoeken via de achterliggende algoritmen.

Datastructuren

De kandidaat kan verschillende abstracte datastructuren met elkaar vergelijken op elegantie en efficiëntie.

Automaten

De kandidaat kan eindige automaten gebruiken voor de karakterisering van bepaalde algoritmen.

Grammatica's

De kandidaat kan grammatica's hanteren als hulpmiddel bij de beschrijving van talen.

Nieuw examenprogramma, programmeren

Ontwikkelen

De kandidaat kan, voor een gegeven doelstelling,

- programmacomponenten ontwikkelen in een imperatieve programmeertaal;
- daarbij programmeertaalconstructies gebruiken die abstractie ondersteunen;
- programmacomponenten zodanig structureren dat ze door anderen gemakkelijk te begrijpen en te evalueren zijn.

Inspecteren en aanpassen

De kandidaat kan

- structuur en werking van gegeven programmacomponenten uitleggen;
- zulke programmacomponenten aanpassen op basis van evaluatie of veranderde eisen.

Nieuw examenprogramma, keuzeprogramma: algoritmiek

Keuzethema Algoritmiek, berekenbaarheid en logica

Complexiteit van algoritmen

De kandidaat kan

(havo:) van gegeven algoritmen de complexiteit vergelijken, en kan klassieke 'moeilijke' problemen herkennen en benoemen.

(vwo:) het verschil tussen exponentiële en polynomiale complexiteit uitleggen, kan algoritmen op basis hiervan onderscheiden, en kan klassieke 'moeilijke' problemen herkennen en benoemen.

Berekenbaarheid

De kandidaat kan berekeningen op verschillende abstractieniveaus karakteriseren en relateren, en kan klassieke *onberekenbare* problemen herkennen en benoemen.

Logica

De kandidaat kan eigenschappen van digitale artefacten uitdrukken in logische formules.

Nieuw examenprogramma, keuzeprogramma: Programmeerparadigma's

Keuzethema Programmeerparadigma's

Alternatief programmeerparadigma

De kandidaat kan van minimaal één extra programmeerparadigma de kenmerken beschrijven en kan programma's volgens dat paradigma ontwikkelen en evalueren.

Keuze van een programmeerparadigma

De kandidaat kan voor een gegeven probleem een afweging maken tussen paradigma's voor het oplossen ervan.

Voorbeelden van mogelijke paradigma's zijn: functioneel, logisch, objectgeoriënteerd.

Alg. Denken: Diagonals

<werkblad>

AD diagonals opdracht

- Traceren
- Rainfall probleem – alle ‘plannen’ komen samen
- Diagonals – CT aspecten getoetst

Ervaringen mbt sorteren / challenges

Challenge: Bit Parity

- Demo

Bit Parity

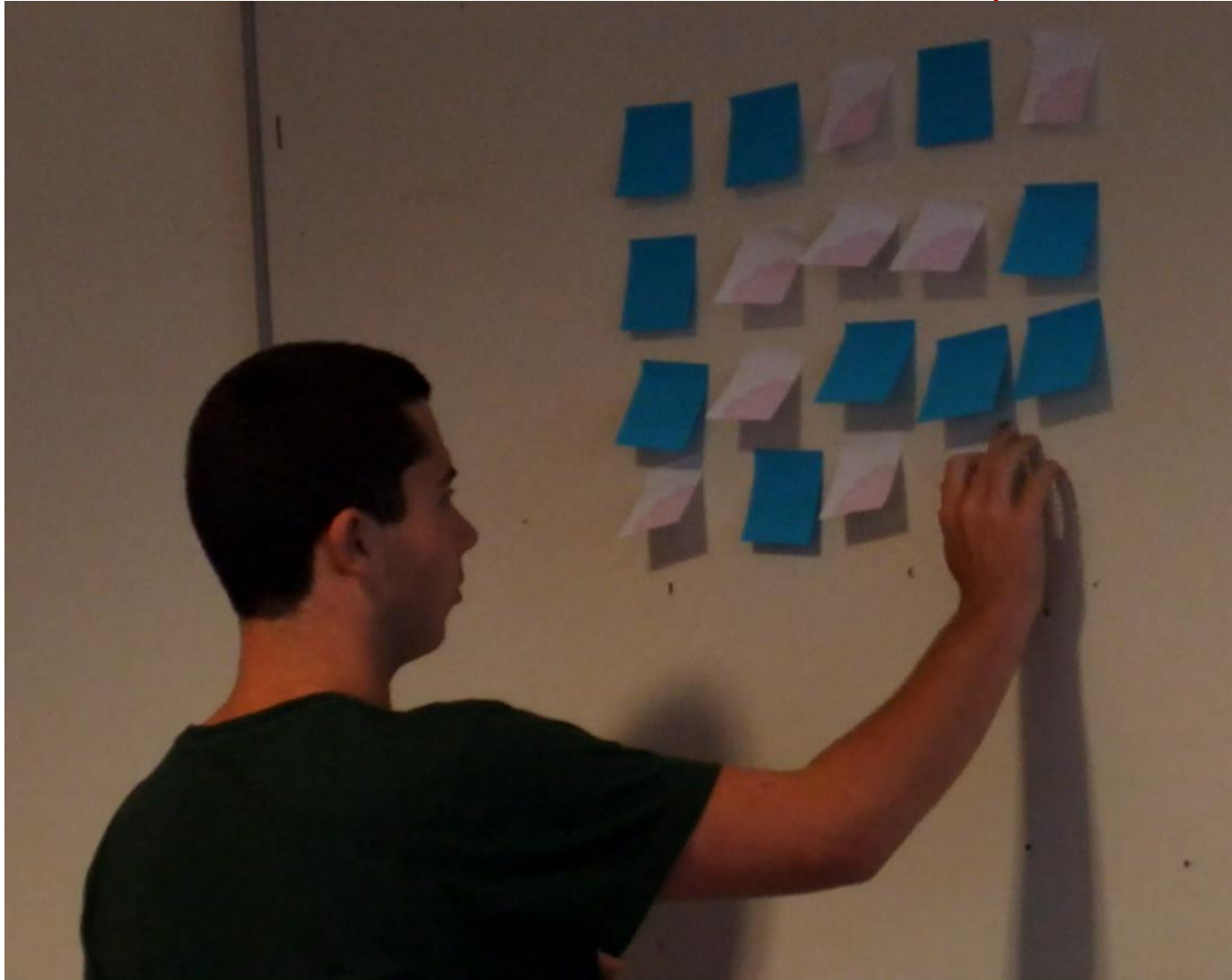
- *Achtergrond:*
 - *Foutdetectie en -correctie algoritme*
 - *krasje op CD*
 - *Cyclic Redundancy Check*
 - *data transmissie over een netwerk*

Kies jouw eigen uitdaging:



<http://course.cs.ru.nl/greenfoot/docenten.html>

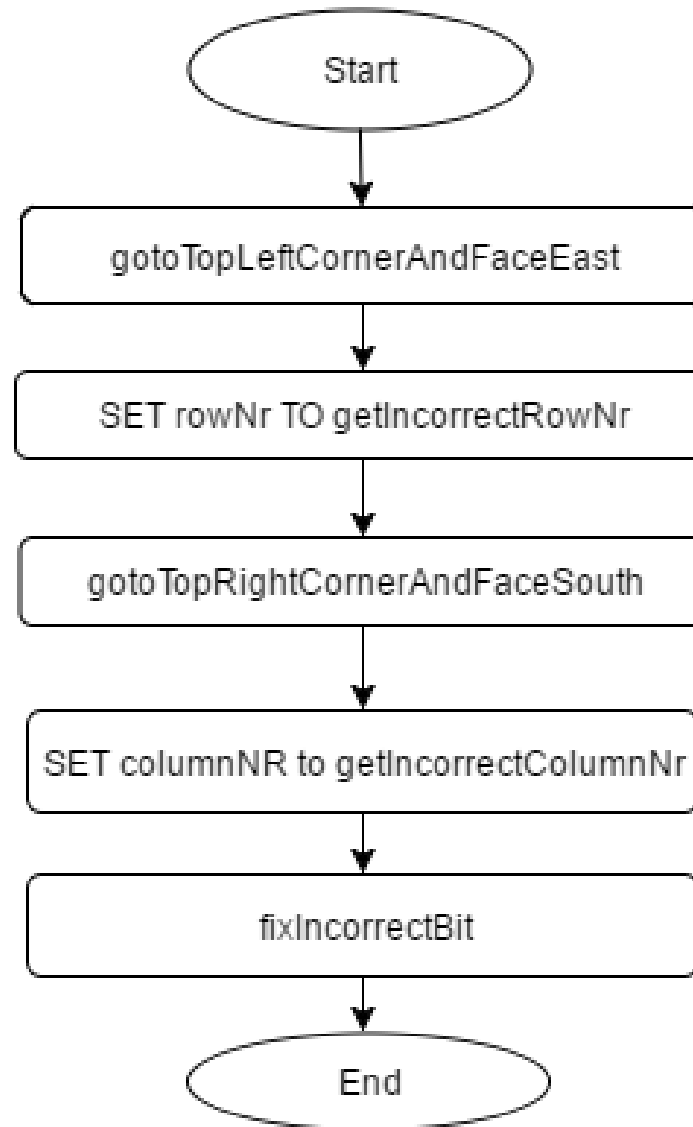
Reflectie BitParity



BitParity and the big five



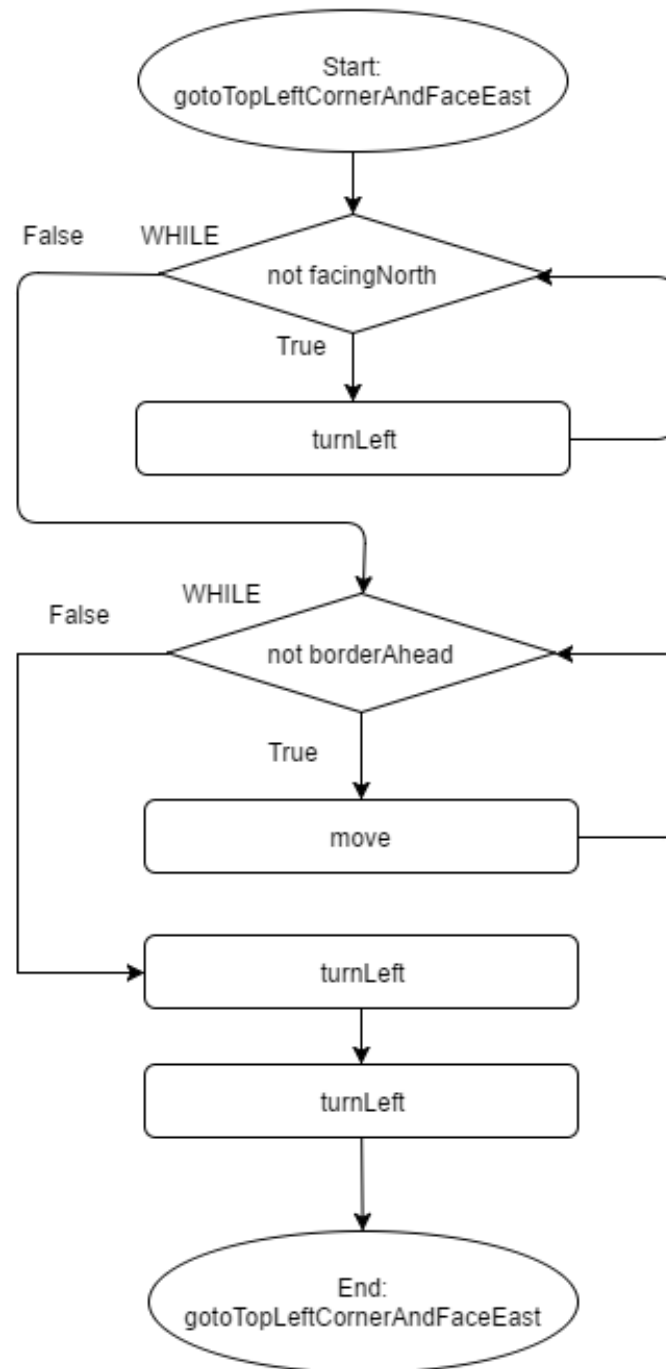
Rhino deel 2:



Stap 1:
beschrijven op
hoog niveau

Beginsituatie: Dodo op een willekeurige positie in wereld

Rhino deel 2:



Stap 2:
complexe
stappen
opdelen in
substappen

BitParity & the big five

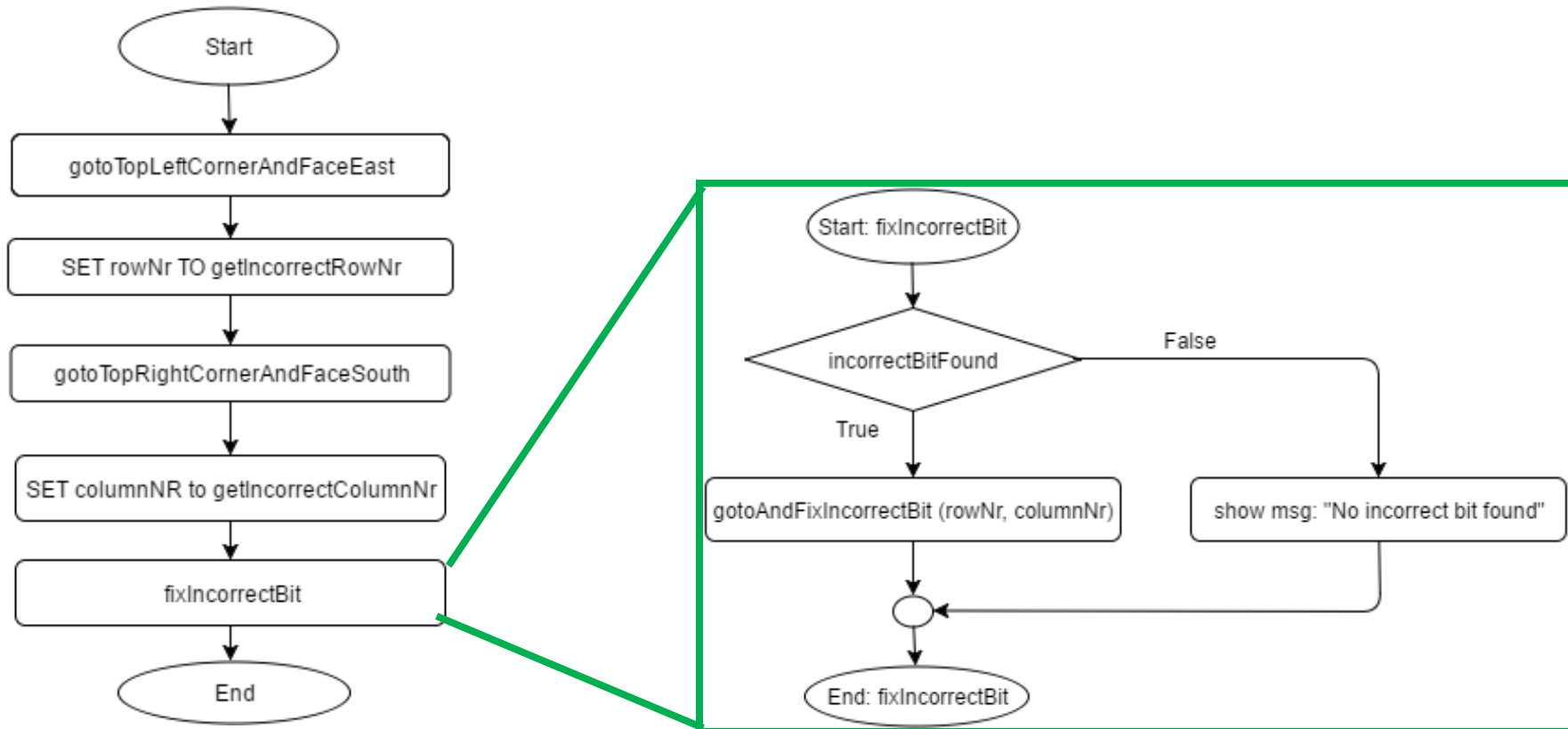
- Algoritmiëk: denken in regels en opeenvolgingen (van deeltaken)
- Abstractie: weglaten van details
- Decompositie: opdelen in deelproblemen
- Generalisatie: patronen & generieke opl.
- Evaluatie: voldoet oplossing aan eisen?
had het beter/makkelijker gekund?

BitParity: Algoritmiek

- Conditional statements
 - `not facingNorth`, `not borderAhead`
- Boolean statements
 - `moreRowsToCheck`, `isEven`
- Bedenk Pre- en post condities:
 - `countEggsInRow`: gaat na afloop terug naar begin van rij

Bitparity: Abstractie

- Algoritmes op hoog-niveau



BitParity: Decompositie

Methodes en submethods:

- `fixIncorrectBit()`
 - Check of ei gelegd of weggehaald moet worden en repareert deze
- `countEggsInRow()`
 - Tel aantal eieren in rij

BitParity: Generalisatie

- Herken mogelijkheden voor hergebruik
- Voordelen en nadelen van hergebruik afwegen
- Minder afhankelijk van details:
 - Wereld grootte: gebruik van constantes
of `borderAhead()`
 - Methodes met parameters: `count eggs` in WELK rij
 - Richtingsgevoel:
 - Maakt niet uit: rij/kolom of initiele kijkrichting
 - ... Je loopt gewoon door tot einde wereld bereikt is

BitParity: Generalisatie

- Mogelijkheden voor hergebruik: `countEggsInRow` voor kolommen
- Hoe generiek is `GoToInitialPosition`?
Keuze voor relatieve implementatie maakt oplossing eenvoudiger, bv `loopNaarHoekLinksAchterJe`:
 - Geen noodzaak richtingsgevoel
 - Hergebruik `getIncorrectRowNr` ook voor kolom
 - Keuze is bepalend voor verdere keuzes die je maakt, bv hoe kolommen aflopen en `gotoIncorrectBit`

BitParity: Evaluatie

- Correctheid?
 - Werkt ook bij 0 of 2 incorrect bits?
 - Als ei wordt toegevoegd (ipv kwijt)?
- Herbruikbaar?
 - Submethodes, JavaDoc, Pre-post condities
- Mogelijke verbeteringen:
 - Oplossing
 - Aanpak

BitParity Samenvattend

The big five:

- Algoritmie: denken in regels en opeenvolgingen (van deeltaken)
- Abstractie: weglaten van details
- Decompositie: opdelen in deelproblemen
- Generalisatie: patronen & generieke opl.
- Evaluatie: voldoet oplossing aan eisen?
had het beter/makkelijker gekund?

Japanese Puzzle

				2										
				1		3			1	3				
				1	1	2		4	2	1				
			3	1	1	2	6	4	1	1	3	2		
1	1	1	1											
1	1	1	1											
			6											
	1	2	1											
1	2	1	2											
	1	4	2											
	2	2	1											
			5											
1	1	1	1											

1	1	1	2		■		■					■		■	■
---	---	---	---	--	---	--	---	--	--	--	--	---	--	---	---

Japanse Puzzel – waar gaat het om?

- Representatie van data
 - Plaatje als getallen
 - Geschikt voor computer (om te zetten naar binair)
 - Minder getallen nodig dan simpel binair representatie: sprake van compressie
 - Getallen als plaatje
 - QR
 - authenticatie codes



To Run or not to Run

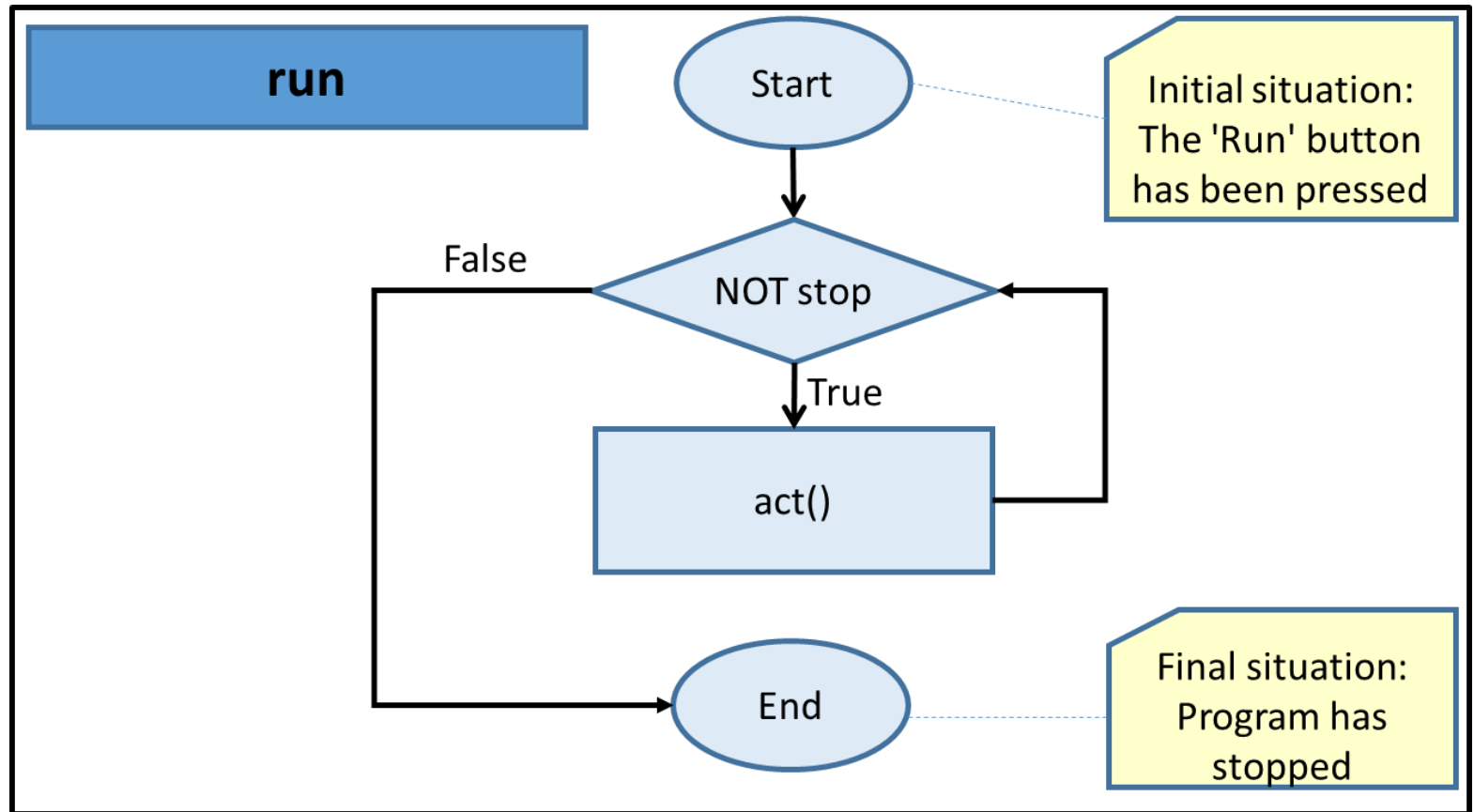
Eenmalige aanroep:

- met rechtermuisknop
- Methode in `act()` aanroepen
en dan knopje Act

Herhaaldelijk (als while)

- Methode in `act()` aanroepen
en dan knopje Run

Greenfoot Run



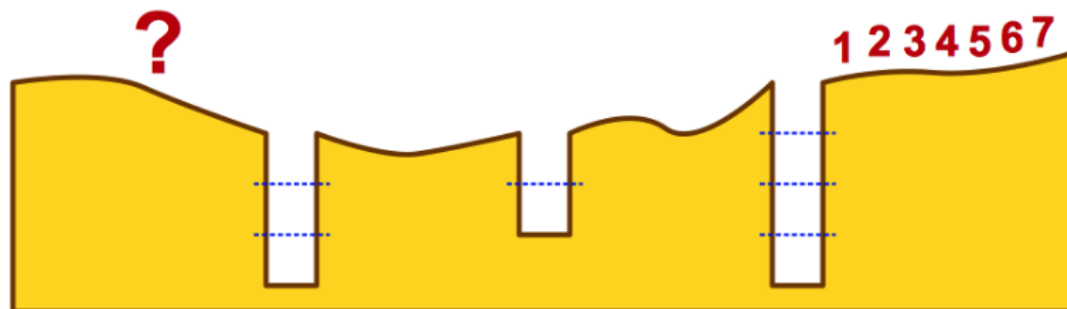
Run

- Geen while in code meer nodig
- Gebruiken bij spellen:
 - Gebruikersinteractie
- Komt in deel 2 van lesmateriaal aan bod

Onze bevers moeten vaak in groepen door het donkere bos. In het bos zijn de paden erg smal. Daarom lopen ze in een rij. Ze kunnen elkaar niet passeren. In het bos zijn veel diepe smalle kuilen. Bevers passen wel op elkaar, maar niet naast elkaar in zo'n kuil. Ze passeren zo'n kuil als volgt:

Eerst springen er net zo veel bevers in de kuil, totdat de kuil vol is. Daarna lopen de andere bevers uit de groep over de gevulde kuil. Tenslotte klimmen de bevers uit de kuil. Dan kan de groep zijn weg vervolgen. De plaatjes laten zien hoe 5 bevers één kuil passeren. In deze kuil passen 3 bevers.

Een groep van 7 bevers loopt door het bos. De bevers moeten over drie kuilen zien te komen. In de eerste kuil passen 4 bevers, in de tweede 2 bevers, en in de derde passen 3 bevers.



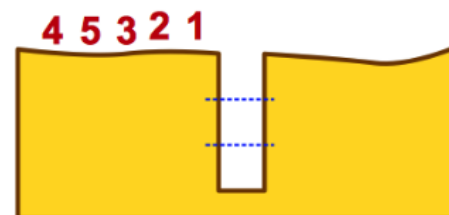
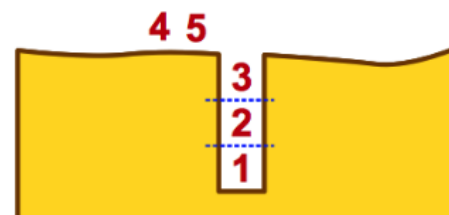
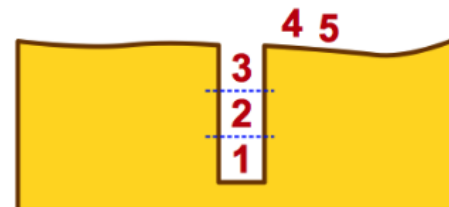
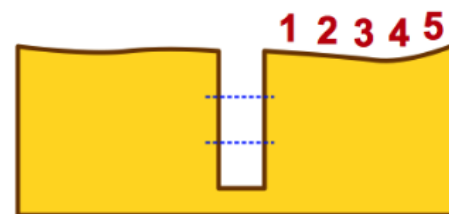
In welke volgorde lopen de bevers na de derde diepe kuil verder?

4 7 5 6 1 2 3

5 7 6 1 4 3 2

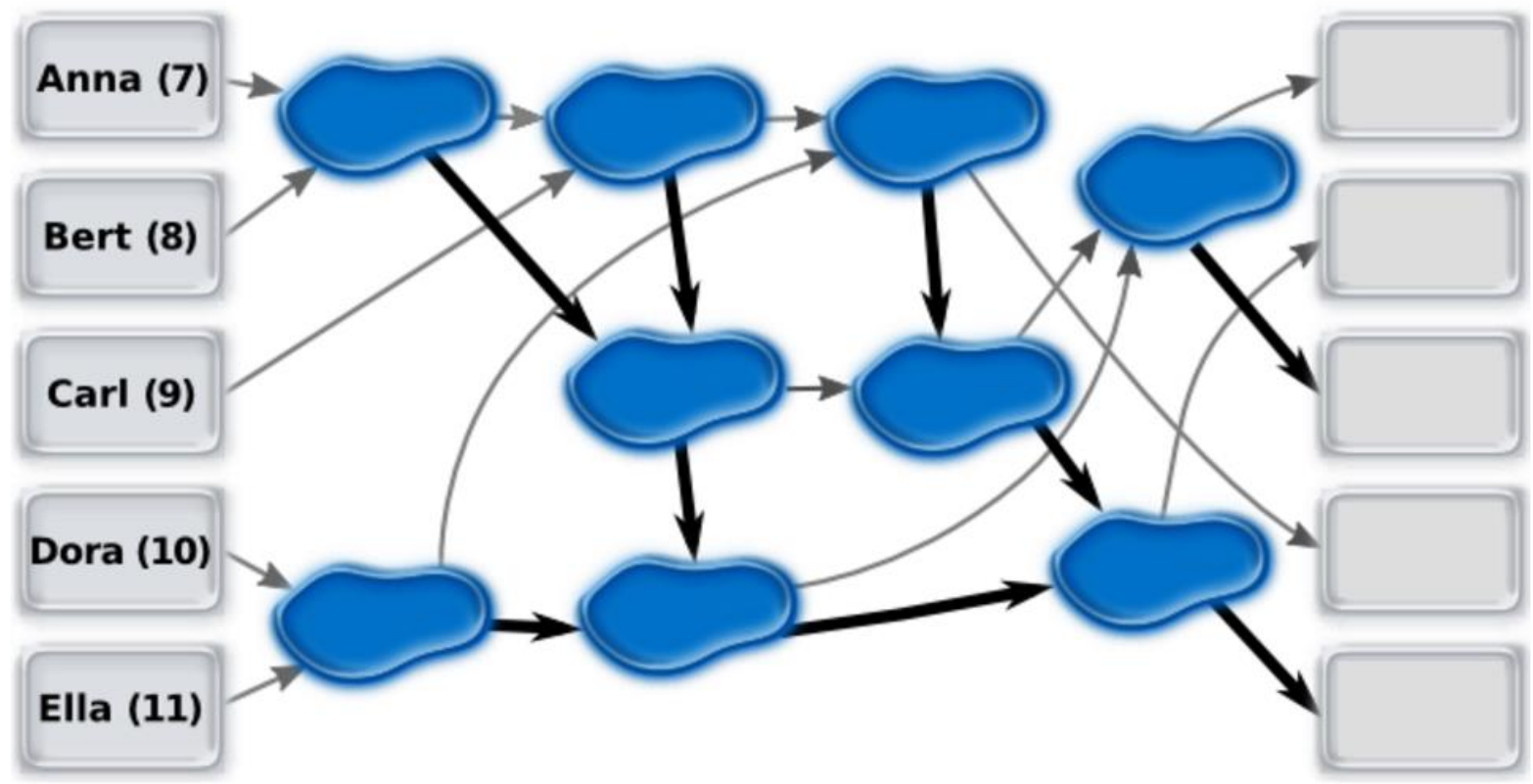
6 5 7 4 3 2 1

2 1 6 5 3 4 7



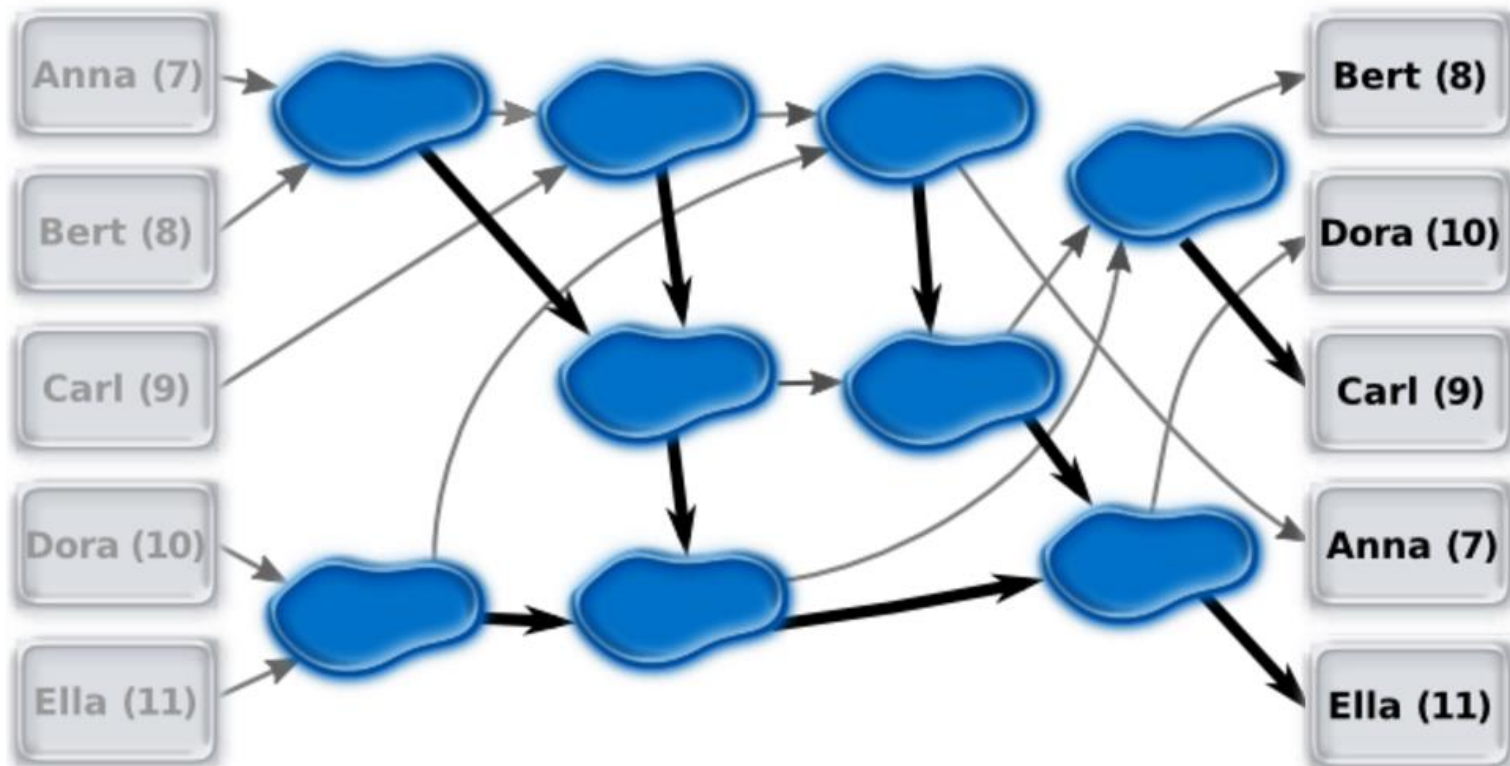
Anna (7 jaar), Bert (8 jaar), Carl (9 jaar), Dora (10 jaar) und Ella (11 jaar) spelen een spel, waarbij ze van plas naar plas springen. Daartoe hebben ze pijlen op de grond getekend. Aan het begin staan de kinderen op de plekken links en springen de pijlen volgend naar een volgende plas. Een kind, dat als eerste in een plas staat, moet wachten totdat er een tweede bij komt. Het oudste kind springt dan verder en moet de dikke pijl volgen, terwijl de jongste de dunne pijl moet volgen.

Sleep elk van de namen naar het veld rechts, waarop het kind aankomt.



Anna (7 jaar), Bert (8 jaar), Carl (9 jaar), Dora (10 jaar) und Ella (11 jaar) spelen een spel, waarbij ze van plas naar plas springen. Daartoe hebben ze pijlen op de grond getekend. Aan het begin staan de kinderen op de plekken links en springen de pijlen volgend naar een volgende plas. Een kind, dat als eerste in een plas staat, moet wachten totdat er een tweede bij komt. Het oudste kind springt dan verder en moet de dikke pijl volgen, terwijl de jongste de dunne pijl moet volgen.

Sleep elk van de namen naar het veld rechts, waarop het kind aankomt.



Nascholing AD

Samenvattend....

Doelstelling materiaal

- Kennismaken met de *kunst* van het oplossen van computationele problemen
- Leuk
 - Puzzelen
 - Creatief
 - Zoeken naar oplossingen
- Serieus
 - Creëren van oplossingen



Lesmateriaal

- Focus: algoritmisch denken
- Itereren per subprobleem:
 - think-flow-code-evaluate
- Uitdagingen worden steeds complexer
- Ontwikkelen mbv flowcharts
 - Doel: Hoog niveau beschrijving
 - Opdelen in kleinere problemen

Welk ander material?

- CS unplugged

www.csunplugged.org

- Teaching London Computing

www.teachinglondoncomputing.org

- Computer Science For Fun

www.cs4fn.org

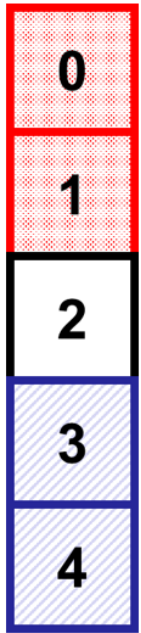
- Bever wedstrijd / Bebras

www.beverwedstrijd.nl / www.bebas.org

- Informatica olympiade / Int. Olympiad Informatics

www.informaticaolympiade.nl / www.ioinformatics.org

Unplugged



- Voordelen:
 - Leuk & uitdagend
 - Activatie leerling
 - Laag drempelig, iedereen kan begin maken
 - Samen leren (van elkaar, verschillende aanpakken/oplossingen), communiceren
- Nadelen
 - Voorbereiding docent
 - Mogelijke misconcepties / versimpeling

Alg Denk-Greenfoot lessen bestaan uit 2 delen

1. In de les: activiteit in de hoofd van de leerlingen
 - *Nadenken* zonder pc
 - Growth mindset
2. Opdrachten: oplossingen vertalen zodat computer er wat mee kan
 - Niet per se gerelateerd aan unplugged
 - Elk programmeertaal zou volstaan

Lesmateriaal deel 2

Status:

- Concept wordt nu getest

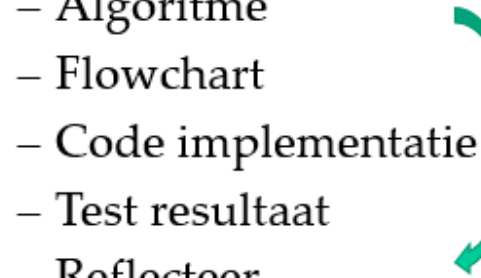
Inhoud:

- Run – dus ook spellen, gebruikersinteractie, visualisaties (molecule botsen, zonnestelsel)
- OO
- Recursie (L-systemen)
- Complexiteit
- (andere programmeeromgeving)

Verwachtingen voldaan?

- Unplugged didactiek
 - Samen leren / puzzelen / werken
- Greenfoot opdrachten met 'Bever'achtige puzzels
- Stappenplan

Per deelprobleem:

- 
- Algoritme
 - Flowchart
 - Code implementatie
 - Test resultaat
 - Reflecteer

Inf4All

The screenshot shows a web browser window with three tabs: 'Nascholing Alg Denken', 'SwissEduc - Computer S...', and 'Beta4all'. The address bar displays 'www.beta4all.nl/prog_inf4all.htm'. The page content is as follows:

- Navigation Menu:**
 - > Home
 - > Programmainformatie
 - > Informatica (inf4all)
 - > Natuurkunde (natk4all)>>
 - > Wiskunde-ondersteuning
 - > Scheikunde (chem4all)
 - > Onderwijsvorm en locatie
 - > Toetsing
 - > Inschrijven en kosten
 - > Contact
- Programma Inf4all:**

Het aanbod voor informatica bestaat uit acht cursussen van ieder 6 ec: twee cursussen op academisch basisoniveau voor kandidaten met een hbo-vooropleiding, en zes cursussen op gevorderd niveau.

	Semester 1	Semester 2
Academisch basisoniveau		
Grondslagen	x	
Algoritmiek		x
Gevorderd algemeen		
Advanced Object Oriented Programming	x	
Netwerken	x	
Databases en Information Retrieval		x
Gevorderd specialisatieonderwerpen		
Media, games, user experience	x	
Kunstmatige Intelligentie		x
Internet of things		x
- Jaarrooster:**

Alle cursusbijeenkomsten vinden plaats op vrijdagen. [Download jaarrooster 2016-2017](#)

De bijeenkomsten in het eerste semester zijn gepland van 11.00 tot 16.00 uur (inclusief middagpauze), beginnend in week 36 en doorlopend t/m week 5. Het tweede semester begint in week 7 en loopt door t/m week 27, cursustijden worden in de loop van september bekend.

Vakantie: de periode tussen kerst en oud & nieuw en de goede vrijdag zijn onderwijsvrij. In de overige vakanties gaan de bijeenkomsten door.
- Testimonials (Cursisten over natk4all):**
 - "Heerlijke colleges met goede enthousiaste docenten!"
 - "Top. Inhoudelijk goed en prettig overgebracht."
 - "Grote kracht is de goede docenten die weten waar ze het over hebben, geweldig!"
 - "Leuk aan de colleges en de stof vond ik de toepassingen en het filosofische karakter."
- Other elements:**
 - A sidebar with a navigation menu.
 - A 'Moodle' logo with a 'Log in op de ELO' link.
 - A 'Meteen inschrijven?' link.

http://www.beta4all.nl/prog_inf4all.htm

Afsluiting / Evaluatie

Vragen? Opmerkingen?

