

# Computational thinking

## Ideeën voor de reken-wiskundeles

*Vincent Jonker & Monica Wijers, Universiteit Utrecht: Freudenthal Instituut, Faculteit Bètawetenschappen / Onderwijsadvies & Training, Faculteit Sociale Wetenschappen*

### Inleiding

Geïnspireerd door de *Grote Rekendag 2016: Kijkje achter de code* en de ervaringen daarvan in de klas verkennen we in dit hoofdstuk de mogelijkheden om *computational thinking* een nadrukkelijker en regelmatig weerkerend plekje te geven in de rekenlessen. We willen een lans breken voor activiteiten in de rekenles die enerzijds zonder computer gedaan kunnen worden, maar anderzijds wel zicht geven op *automatische processen*, die vaak door computers worden afgewikkeld. Die automatische processen beïnvloeden ons leven op alle fronten, en het is zaak dat we met enig inzicht en kritisch met deze automatische processen kunnen omgaan. En omdat automatische processen volgens vaste (reken)regels verlopen is het goed in de rekenles hier ook aandacht aan te besteden. Dat is goed voor computational thinking en dat is goed voor de reken-wiskundeles.

### Achtergrond

Laten we helder zijn. Dit hoofdstuk is geen pleidooi voor een cursus programmeren voor iedereen. We zien dergelijke stemmen opgaan in bijvoorbeeld Engeland, waar het vak programmeren verplicht is gesteld, en ook in Nederland zijn er mensen die zeggen dat alle kinderen zouden moeten leren programmeren.

Wij denken dat dit te veel aandacht geeft aan slechts een beperkt en moeilijker onderdeel van computational thinking en dat we het breder en laagdrempeliger moeten zoeken. Laten we daarom eerst kijken naar een definitie:

Computational thinking is het procesmatig (her)formuleren van problemen op een zodanige manier dat het mogelijk wordt om met computertechnologie het probleem op te lossen. Het gaat daarbij om een verzameling van denkprocessen waarbij probleemformulering, gegevensorganisatie, -analyse en -representatie worden gebruikt voor het oplossen van problemen met behulp van ICT-technieken en -gereedschappen (<http://curriculumvandetoekomst.slo.nl/>).

Op deze wijze gedefinieerd is het een brede set van vaardigheden waar iedereen op een of andere manier in zijn of haar dagelijkse leven mee te maken heeft. Het gaat dan onder andere om:

- Zoeken naar regelmaat en patronen, bijvoorbeeld bij het analyseren van gegevens;
- Systematisch werken, bijvoorbeeld bij het oplossen van problemen;
- Maken en lezen van symbolische weergaven, bijvoorbeeld bij simuleren en modeleren;

- Interpreteren en zelf ontwerpen van codetaal, bijvoorbeeld bij het beschrijven van een proces.

Met deze omschrijvingen zijn we al veel dichterbij gekomen bij zaken die ook in de rekenles thuis horen en die in de beschrijving van 21e eeuwse vaardigheden worden genoemd (Voogt e.a., 2010). Het gaat dan ook in de rekenles niet zozeer om basisvaardigheden maar om hogere orde vaardigheden zoals probleemoplossen en kritisch denken (zie ook *Wiskunde voor Morgen*) en het vertoont ook overeenkomsten met onderzoekend leren (Van Graft, 2007).

Het wordt nu tijd om concreter te worden. Wat zijn dat dan voor activiteiten die je kunt uitvoeren op het gebied van *automatische processen* zonder dat je er een computer bij gebruikt, en die ook nog met rekenen te maken hebben?

## Activiteiten voor in de reken-wiskundeles

We geven enkele voorbeelden, die onder andere gebruikt zijn tijdens de *Grote Rekendag* van 2016 (Keijzer, 2016).

### Groep 1 & 2: Welk getal ben ik?

Dit spel voor groep 1 en 2 is een variant op het raadspelletje *Wie ben ik?* In dit spel krijgt een van de kinderen een getal (of aantal stippen) tussen één en twintig op zijn of haar muts (Afb. 1). Het kind met de muts op probeert te raden welk getal dat is. Een of meer van de andere kinderen geven aanwijzingen, bijvoorbeeld door gebaren te maken zonder te praten, of mét praten door het aantal te omschrijven, zónder daarbij getallen te noemen, bijvoorbeeld *evenveel als de poten van de tafel*.



Afb. 1. Welk getal ben ik?

*Wat heeft dit met rekenen te maken?*

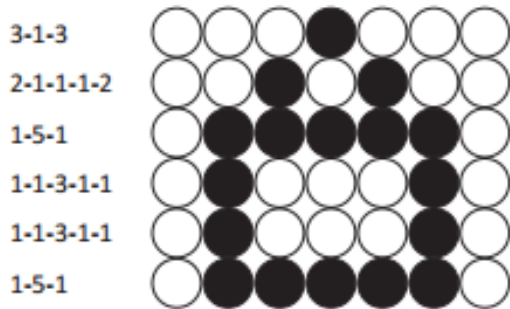
Speels en systematisch leren omgaan met getallen of aantallen en de 'kenmerken' daarvan.

### Robotpuzzel (groep 3 & 4) / Pixel voor pixel (groep 5 tot en met 8)

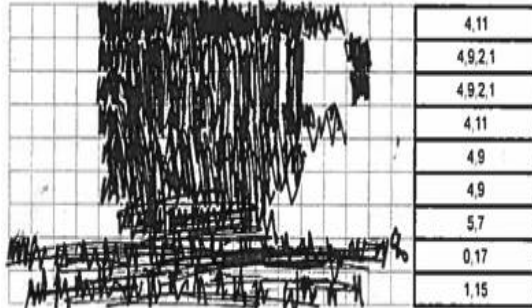
Voor puzzelaars geen onbekende. Van een tekening zijn alleen *aantallen* bekend zoals zichtbaar in de linker kolom van afbeelding 2 en de rechter kolom van afbeelding 3 (4,11 en 1,9,2,1). In feite zijn dit getallen die aangeven hoe lang er niets gewijzigd hoeft te worden per horizontale regel. De afspraak in deze voorbeelden is dat er steeds begonnen wordt met een wit vakje, dus 4,11 betekent dan: doe eerst vier witte *pixels* en dan elf zwarte (het bovenste randje van het kopje in de tekening).

In grafische software (zoals Photoshop en in video-opslag) wordt deze techniek gebruikt om met zo min mogelijk gegevens de definitie van een heel plaatje of van een video te kunnen geven.

Deze activiteit is bruikbaar vanaf groep 3 en 4, waar de wat eenvoudigere opdrachten gebruikt worden. In groep 7 en 8 kan het wat ingewikkelder zijn: daar wordt bijvoorbeeld gewerkt met een nul aan het begin van een regel om te zorgen dat ook het eerste vakje gekleurd wordt (omdat het eerste cijfer – nul – het aantal witte vakjes aangeeft). Dit blijkt lastig. Verder wordt de activiteit in groep 7 en 8 uitgebreid met het zelf bedenken van een code-systeem voor tekeningen met meerdere kleuren.



Afb. 2. Voorbeeld voor groep 3 en 4.



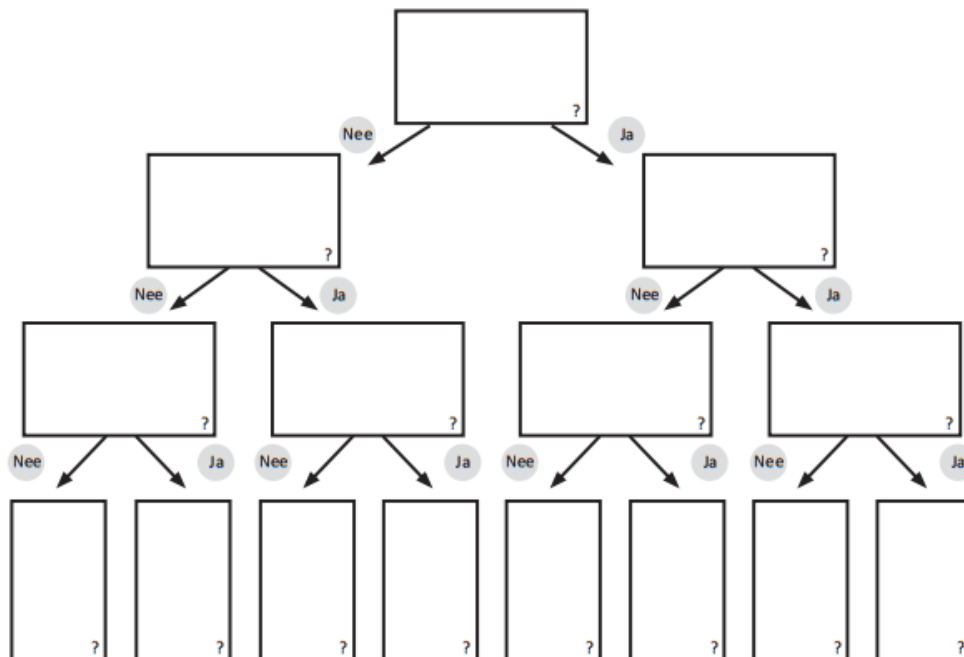
Afb. 3. Voorbeeld voor groep 7 en 8.

*Wat heeft dit met rekenen te maken?*

Systematisch tellen. Het gebruik van cijfers/getallen als symbolen om een instructie weer te geven.

### Sorteren volgens verschillende regels (groep 5 t/m 8)

Dit is een activiteit om leerlingen zicht te geven op hoe computers met vaste regels werken. In groep 5 en 6 wordt gesorteerd met de methode *herhaald verdelen* met behulp van een boomdiagram. De leerlingen sorteren een stapel kaartjes met getallen (op volgorde van grootte) of met woorden (op alfabetische volgorde) door steeds een verdeling te maken op basis van een handige *ja/nee-vraag*.



Afb. Boomdiagram voor groep 5 en 6.

De opdracht in groep 7 en 8 luidt: Gebruik de *bubble-sort-regel* om tien kaartjes met getallen te sorteren. De leerlingen werken in groepjes van vier en ze krijgen tien kaarten met getallen er op tussen 0 tot 100 (5, 16, 23, 37, 42, 58, 61, 79, 88 en 90). De kaartjes worden in willekeurige volgorde in een rij gelegd. De *bubble-sort-regel* die de leerlingen nu moeten hanteren luidt: De rij langsgaan en steeds twee kaarten omwisselen als de volgorde (van klein naar groot) (nog) niet klopt. Dit blijf je doen (steeds weer opnieuw beginnen, van links naar rechts) tot er geen kaartjes meer van plek ruilen. De gehele rij ligt dan op volgorde van klein naar groot.



Afb. 5. Bubble-sort voor groep 7 en 8.

*Wat heeft dit met rekenen te maken?*

Systematisch werken volgens een vast algoritme. Eenvoudige check van groter/kleiner, getallenrij.

### **Live Turtle (groep 7 & 8)**

In deze opdracht schrijven de leerlingen een instructie in stappen, in de vorm van een computerprogramma, voor het maken van een eenvoudige tekening. Ze doen dit zo dat iemand anders aan de hand van alleen die instructies de tekening kan maken. Dat kan in het klein op papier, maar ook in het groot op het schoolplein.

De leerlingen krijgen een paar commando's die ze mogen gebruiken om de instructies te schrijven, zoals: kleur ...; draai ... graden; herhaal ... keer; pen op; loop ... stappen; pen neer; ga naar het beginpunt. Samengevat krijg je dan de volgende opdracht:

- Verzin zelf een tekening (niet te moeilijk).
- Schrijf voor die tekening een programma op een papiertje.
- Geef vervolgens de commando's één voor één aan iemand anders uit jouw klas.
- Als jouw programma goed is (en er goed wordt geluisterd) ontstaat jouw tekening.

Deze opdracht kan gezien worden als een voorbereiding op het programmeren met bijvoorbeeld Scratch.

*Wat heeft dit met rekenen te maken?*

Speels leren omgaan met hoeken, afstanden, herhaling van stappen.



Afb. 6. Live turtle op het schoolplein.

Het moge duidelijk zijn dat dit overzicht van voorbeelden bewust zo gekozen is dat we kunnen laten zien dat de mogelijkheden voor *computational thinking* beginnen bij het jonge kind en door lopen naar groep 8 (en natuurlijk verder richting het voortgezet onderwijs).

## Leerlijn of losse activiteiten?

Op Daltonschool Rijnsweerd in Utrecht hebben we gewerkt aan een leerlijn voor computational thinking. Daaraan deden naast de W&T coördinator uit groep 6, ook leerkrachten mee uit de lagere groepen. Men zoekt nu naar een voor iedereen eenvoudig in te vullen en te onderhouden setje van lessen door het jaar heen, zodat leerlingen uitgedaagd blijven met opdrachten die steeds iets meer context krijgen en ook wat meer ruimte bieden voor onderzoekend leren.

Voor het aanbrengen van kennis op het gebied van computational thinking is er niet direct sprake van een leerlijn, hoewel het natuurlijk voor de hand ligt om in de lagere groepen met eenvoudiger opdrachten te werken die goed in de belevingswereld van de leerlingen passen. Zo gauw er sprake is van het aanleren van een programmeertaal met al die specifieke kennis die daarbij hoort, zoals het werken met vaste woorden en zinnestelsels, zodat de computer dit kan verwerken, zal er wel een leerlijn ingelegd moeten worden.

## Advies

### Losse activiteiten

Wij durven beweren dat de *losse activiteiten* op het gebied van computational thinking voor iedereen geschikt zijn en in principe ook als een set van losse activiteiten mag blijven bestaan. Misschien is dat zelfs beter, want dan is het ook makkelijker om deze elementen in reken-wiskundemethodes - en methodes voor andere vakken! - op te nemen.

Laat elke individuele leerkracht zich uitgedaagd voelen er sowieso in verschillende lessen iets mee te doen. Zo gauw het een leerlijn is zal er ook een methode zijn en worden veel vrijheidsgraden helaas ook weggenomen bij de leerkracht

### Leerlijn

Als het gaat om het aanleren van een *programmeertaal* is er meer nodig dan een setje losse activiteiten en zal er gewerkt moeten worden met een leerlijn. Maar dat is volgens ons dus een smalle benadering van computational thinking. Deze benadering zou naar onze mening beperkt moeten blijven tot enkele leerlingen die hier echt iets mee willen/kunnen, voor in de buitenschoolse/naschoolse setting of in vormen van gedifferentieerd onderwijs met individuele leerroutes. Bekijk ter inspiratie bijvoorbeeld eens hoe dit wordt uitgewerkt in [codestarter.nl](http://codestarter.nl).

### Meer weten en doen?

- [De Grote Rekendag 2016: Kijkje achter de code.](#)
- [Buble sort met 10 getallen.](#)
- [Live turtle.](#)
- [Codestarter.](#)
- [Codeuur.](#)
- [Csunplugged.](#)
- [Leerlijn programmeren.](#)

### Verder lezen?

- [Curriculum van de toekomst.](#)
- [Wiskunde voor Morgen.](#)

### Literatuur

- Jeuring, J., Corbalan, G., Van Es, N., & Van Montfort, J. (2016). *Leren programmeren in het PO, een literatuurreview*. Utrecht.
- Jonker, V., Wijers, M., Abels, M., & Keijzer, R. (2016). *Let's have a look behind the code. The Big Mathematics Day 2016 (Netherlands) about coding without computer*. Paper presented at the PATT, De Bilt, the Netherlands  
[http://www.fi.uu.nl/publicaties/literatuur/2016\\_jonker\\_wijers\\_abels\\_keijzer\\_patt.pdf](http://www.fi.uu.nl/publicaties/literatuur/2016_jonker_wijers_abels_keijzer_patt.pdf)
- Keijzer, R. (Ed.). (2016). *Kijkje achter de code. Grote Rekendag 2016*. 's-Hertogenbosch: Malmberg.
- Papert, S. (1980). *Mindstorms: Children, computers and powerful ideas*. Brighton: Harvester Press.
- Pijpers, R., Stiller, L., & Boeke, H. (2016). *Computing-onderwijs in de praktijk. Wat kunnen we leren van de britten?* Zoetermeer: Kennisnet.
- Schnabel, P. (2015). *Platform 2032. Hoofdlijn advies: een voorstel*. Den Haag.
- Van Galen, F. & De Moor, E. (1991). Midget-Logo. *Reken-wiskundeonderwijs: onderzoek, ontwikkeling, praktijk*, 10(1), 57-58.
- Van Graft, M., & Kemmers, P. (2007). *Onderzoekend & Ontwerpend Leren bij Natuur & Techniek. Basisdocument over de didactiek voor onderzoekend en ontwerpend leren in het primair onderwijs*. Den Haag: Stichting Platform Bèta Techniek.
- Voogt, J., & Pareja Roblin, N. (2010). *21st Century Skills. Discussienota*. Zoetermeer.

Jonker, V. & Wijers, M. (2017). Computational thinking. Ideeën voor de reken-wiskundeles. In: M. van Zanten (red.). *Rekenen-wiskunde in de 21<sup>e</sup> eeuw. Ideeën en achtergronden voor primair onderwijs* (pp. 137-142). Utrecht / Enschede: Panama, Universiteit Utrecht / NVORWO / SLO.