

Kan dit altijd zo?

COMPUTATIONAL THINKING IN ELKE REKENWISKUNDELES

Tot de 21^e eeuwse vaardigheden wordt ook computational thinking gerekend. Een term waar niet zomaar een eenvoudige vertaling voor gevonden wordt. De term suggereert dat het gaat om het gebruik van de computer, maar in het huidige onderwijs zijn veel voorbeelden van computational thinking te vinden waar geen computer aan te pas komt. Anna Hotze en Ronald Keijzer geven een aantal voorbeelden en laten zien hoe u deze kunt gebruiken in uw onderwijs.

Tekst

Anna Hotze &
Ronald Keijzer

Anna Hotze is lector
Wetenschap en
technologie aan
Hogeschool iPabo,
Amsterdam/Alkmaar
Ronald Keijzer is lector
rekenen-wiskunde
aan Hogeschool iPabo,
Amsterdam/Alkmaar

PLANBORD

Het is kwart voor negen in de ochtend. De kleuters van juf Sylvia zitten in de kring. Nadat kort met elkaar teruggeblikt is op wat iedereen gisteren gedaan heeft op school en thuis, richt Sylvia het gesprek op het kiezen voor het spelen van deze ochtend. Ze vertelt de kinderen dat ze weten dat ze zelf mogen kiezen, maar dat ze daarbij moeten bedenken dat ze iets moeten kiezen dat ze de afgelopen dagen niet hebben gedaan. Dat geldt overigens niet voor kleuters die met de juf hebben afgesproken dat ze iets mogen afmaken. Verder mag er alleen iets gekozen worden wat nog niet vol is. Sylvia wijst op het planbord, waarop straks magneetjes met voor ieder kind een herkenbare afbeelding aangeven wat welk kind gaat doen: 'In de bouwhoek passen maar vier kinderen, kijk maar.' Op het planbord is bij de bouwhoek inderdaad plek voor precies vier magneetjes. De magneetjes van de kinderen die nog iets mogen afmaken worden het eerst op het planbord geplaatst. Daarna wijst Sylvia één voor één aan waar de kinderen uit kunnen kiezen. Als veel kinderen een keuze gemaakt hebben en het planbord al aardig vol is, is Jaimy aan de beurt (afbeelding 1). Zij wil in de huishoek spelen, maar dat is niet mogelijk. Juf Sylvia: 'Kijk eens goed naar het bord. Daar spelen al twee kinderen.' Dat ziet Jaimy ook. Zij vindt dat jammer, maar wil ook wel een andere keuze maken. Ze gaat razendsnel met haar ogen het hele bord langs van links naar rechts en ze ziet direct bij de leeshoek een lege plek. Maar daar heeft ze geen zin in. De andere lege plek is bij de bouwhoek en dat lijkt Jaimy wel leuk.

Het planbord is een schematische weergave waarop wordt aangegeven welk kind wat gaat doen. Dit schema leidt tot redeneren over de keuze. Immers,



1. Het planbord in de kleuterklas (Foto: Brit Nederhoed)



Computational thinking is een specifieke manier van denken om tot de kern te komen van een probleem.

kinderen weten dat de keuze niet helemaal vrij is. Zo mag je bijvoorbeeld niet met te veel kinderen in de bouwhoek. Het planbord biedt een mogelijkheid om het kiezen te ondersteunen. Het bord snel langslopen is een systematische werkwijze om te bepalen waar er wel gespeeld kan worden. Wat Jaimy zo feitelijk doet is een model, het planbord, gebruiken om via een vaste werkwijze of algoritme een plekje voor haarzelf veilig te stellen.

21^e-EEUWSE VAARDIGHEID

In het basisonderwijs wordt momenteel regelmatig gesproken en nagedacht over computational thinking, een van de zogenaamde 21^e-eeuwse vaardigheden. Computational thinking roept de nodige vragen op. De term suggereert dat het gaat om het gebruik van de computer, maar er zijn tal van voorbeelden van computational thinking waar geen computer aan te pas komt. Bij computational thinking gaat het om het (her)formuleren van problemen zodat deze uiteindelijk met behulp van een computer kunnen worden opgelost. Dit vraagt om computers maar vooral ook om meer algemene vaardigheden. Computational thinking is een specifieke manier van denken. Bij computational thinking gaat het onder meer om het verzamelen, analyseren en visualiseren van gegevens, waarbij ook modelgebruik een rol kan spelen. Daarnaast gaat het om het opdelen van een taak in kleine overzichtelijke deeltaken en in het algemeen om het reduceren van de complexiteit van een probleem. Het gaat om het tot de kern komen van het probleem. Ook gaat het bij computational thinking om het gebruik maken van een serie geordende stappen om een probleem op te lossen. Dat is het toepassen van algoritmes en procedures. En tot slot is automatiseren, het door een computer laten uitvoeren van herhalende stappen of taken, onderdeel van computational thinking.

Wat is 'Computational thinking'?

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.

*Overgenomen uit: curriculum van de toekomst/SLO**

In het hierboven beschreven voorbeeld van het planbord laten we zien hoe Jaimy de gegevens op het planbord analyseert, waarbij het planbord een model is om de werkelijkheid van het spelen in de kleuterklas weer te geven. Ook betekent het gebruik van het planbord dat Jaimy een algoritme of procedure moet inzetten. Ze loopt het planbord razendsnel, maar systematisch, van links naar rechts af, om zo na te gaan wat er nog leeg is. Ze overweegt zo bij iedere stap of daar een bordje kan worden neergehangen, totdat ze een plekje tegenkomt waar het bordje past. Dit herhaalt ze tot ze een activiteit heeft gevonden die ze wil kiezen. Dit voorbeeld van het planbord laat zien dat het planbord verschillende activiteiten uitlokt die ook passen bij computational thinking.

DE GETALLENRIJ

Het volgende voorbeeld komt uit de middenbouw. In groep 3 van juf Carla is al regelmatig aandacht besteed aan het tellen aan de hand van de kralenketting. Daarbij heeft de juf de kinderen nadrukkelijk gewezen op de groepjes van 10, de kleurwisselingen van de ketting bij de tientallen en de regelmaat in de namen van de getallen bij de wat grotere getallen. Bij het opschuiven van kralen op de kralenketting leren de kinderen dat het aantal kralen dat opgezet wordt een totaal aantal kralen vertegenwoordigt. Als zij 23 moeten opschuiven, gaat dat op een gegeven moment in 2 groepjes van 10 en 3 'losse' kralen. Ze weten dat '23' gaat over alle kralen die ze hebben opgezet. Dan gaan de kinderen met een opdracht uit de methode aan de slag (afbeelding 2). Daarin zien de kinderen een deel van de kralenketting. Zij krijgen de opdracht om het *rangnummer* van de kraal te noemen bij de pijl.

3 welk getal is het?

a 18 ... 20 21 22 b 21 22 ... 24 25 c 9 10 11 ... 13
 d 24 ... 28 e 20 ... 24 25 f 19 ... 21 22

Carla kijkt met de kinderen naar deze opgave, voordat ze er zelfstandig mee aan het werk gaan. Micky mag vertellen hoe je kunt weten welk getal er moet worden ingevuld: 'Gewoon tellen, tot je de stippeltjes tegenkomt.' Hij mag vertellen hoe hij dat

2. Alles telt 3a, werkboek, p.18



Belangrijke aspecten van computational thinking zitten in vrijwel iedere reken-wiskundeles: gegevens verzamelen, analyseren en visualiseren.

bij de tweede opgave doet. Micky: 'Eenentwintig, tweeëntwintig, drieëntwintig, Je moet drieëntwintig invullen, een twee en een drie.' Ahmed zegt dat het ook sneller kan. Je kijkt gewoon naar het getal ervoor: 'Na tweeëntwintig is drieëntwintig.' Daar is Micky het wel mee eens, maar hij had al bedacht dat zijn manier ook werkt bij de onderste drie opgaven.

De stukken van de kralenketting in deze opgave zijn een voorbode van de getallenlijn. Ook hier ondersteunt de gegeven schematisering het bepalen van het antwoord en ook hier leidt het gebruiken van de schematisering tot een systematische aanpak die altijd gebruikt kan worden, een algoritme. Feitelijk geven zowel Micky als Ahmed een dergelijk algoritme, al is dat van Ahmed niet in alle gevallen bruikbaar.

In het voorbeeld uit groep 3 zien we hoe de kralenketting als model wordt gebruikt om de telrij te visualiseren. Leerlingen moeten om het getal bij het pijltje te vinden een serie geordende stappen vinden en uitvoeren. Het gaat hier weer om het bedenken van een procedure. De kinderen zeggen de telrij op of kijken naar het getal ervoor. In feite gebruiken de kinderen een algoritme en bewegen zich zo op het pad van computational thinking.

SPAREN

In groep 7 van juf Stien maken de leerlingen een opgave over het door Ilse gespaarde bedrag (afbeelding 3). In de opgave bepalen de kinderen voor iedere maand welk bedrag Ilse die maand gespaard heeft. Ze gebruiken daarvoor de bedragen die in de tabel staan. Het verschil tussen twee maanden kan berekend worden door aan te vullen. Het schema waarin de getallen gegeven zijn helpt de kinderen hierbij, want het gaat telkens

om het aantal euro's dat van maand tot maand meer op de rekening staat. Bijvoorbeeld: in maart wordt er door Ilse €44 gespaard, want dat bedrag moet je bij €92 doen om €136 te krijgen.

Het schema in het werkboek van de kinderen biedt zelfs ruimte om een tussenstap te maken. Als we het spaarbedrag van maart willen bepalen, kunnen we eerst een stap van €8 maken om bij €100 uit te komen, om er vervolgens nog €36 bij te doen om bij de €44 te komen.

Het schema stimuleert de leerlingen zelf een algoritme te ontwikkelen om het spaarbedrag van de volgende maand te bepalen. Je begint dan bij het bedrag dat al op de rekening stond. Je vult aan tot het hele honderdtal en maakt daarna de stap naar het bedrag dat bij de volgende maand op de rekening staat.

In het voorbeeld van groep 7, waar de gegevens in een tabel worden gevisualiseerd, moeten de leerlingen eerst de gegevens in de tabel analyseren. Met het visualiseren en analyseren laten ze zien dat ze met aspecten van computational thinking kunnen werken. Ook moeten de leerlingen van groep zeven in dit voorbeeld een serie van opeenvolgende stappen bedenken om het probleem op te lossen, ze gebruiken algoritmes en procedures. Ook die zijn typerend voor computational thinking.

TRADITIONELE ALGORITMEN

In bovengenoemde voorbeelden lieten we zien dat leerlingen in nogal wat gevallen bezig zijn met algoritmen waar je dat normaal gesproken niet als zodanig zou benoemen. Ze zijn natuurlijk ook bezig met algoritmen als die algoritmen aan de orde zijn die wel vaak als zodanig worden aangeduid. Het gaat dan om het cijferen of onder elkaar uitrekenen; het cijferend optellen, cijferend

1

Hoeveel spaart Ilse elke maand?

maand	jan	feb	mrt	april	mei	juni	juli	aug	sept	okt	nov	dec
bedrag op de rekening	€ 73	€ 92	€ 136	€ 200	€ 248	€ 291	€ 343	€ 392	€ 457	€ 501	€ 646	€ 829
gespaard	...	...	...	...	...	...	...	...	...	...	...	...



afrekken, cijferend vermenigvuldigen en cijferend delen. Leerlingen leren in de loop van de bovenbouw van de basisschool de cijferprocedures kennen en kunnen die uiteindelijk in een meer of minder verkorte vorm uitvoeren.

Het cijferen heet zo omdat -in de meest verkorte procedure- cijfer voor cijfer gerekend wordt. Bijvoorbeeld bij het onder elkaar uitrekenen van 26×7 :

- 6 keer 7 is 42, de '2' wordt genoteerd en de '4' onthouden,
- 7 keer 2 is 14, daar komt de '4' bij,
- '18' wordt voor de genoteerde '2' geplaatst en het antwoord is 182.

We mengen ons hier niet in de discussie over de noodzaak van het memoriseerd kunnen

toepassen van deze rekenwijze. We kiezen ervoor om de werkwijze te bekijken door de bril van computational thinking. Denkend aan algoritmen bij computational thinking gaat het om het doordenken van de procedure en ligt de vraag voor de hand om te verklaren waarom je de gemaakte stappen moet of mag zetten. Dat betekent dat het in het gesprek met de leerlingen om de volgende vragen gaat:

- Wat is de betekenis van de '2' die als eerste genoteerd wordt?
- Waarom moet je eigenlijk $14 + 4$ uitrekenen?
- Wat betekent het antwoord '18'?

Leerlingen bedenken, als het goed is, dat je met dit algoritme 26×7 uitrekent als $20 \times 7 + 6 \times 7$. Ze zien dat ze daarbij beginnen met het uitrekenen

van 6×7 . Dan pas ga je verder met 20×7 . Daar kun je de laatste nul weglaten, omdat op de plek van de énen al een cijfer staat. Er staat verder eigenlijk ook al wat op de plek van de tien, namelijk een '4' en die mag daar niet zomaar wegvallen. Door 14 en 4 bij elkaar op te tellen, zorg je ervoor dat op de positie van de tien en die van de honderden het goede cijfer komt te staan.

Belang van computational thinking voor het onderwijs

Veel van de huidige maatschappelijke en wetenschappelijke vraagstukken zijn dermate complex dat zij niet zonder de hulp van computertechnologie opgelost kunnen worden. Bij deze vraagstukken is de rekenkracht van de computer nodig om tot een oplossing te komen.

Computational thinking richt zich op de vaardigheden die essentieel zijn om problemen op te lossen waarbij veel informatie, variabelen en rekenkracht nodig zijn. Het is daarbij belangrijk om te begrijpen hoe informatie tot stand komt zodat je computersystemen kan benutten voor probleemoplossen, voor het denken in stappen en daarmee in voorwaardelijkheden voor volgorde van de benodigde gegevens. Computertechnologie gebruiken bij het zoeken naar oplossingen betekent inzicht krijgen in algoritmes (een reeks instructies om vanaf een beginpunt een bepaald doel te bereiken) en procedures (een verzameling activiteiten die in een bepaalde volgorde moet worden uitgevoerd)

Overgenomen uit: curriculum van de toekomst/SLO

DISCUSSIE EN AFSLUITING

De voorbeelden maken duidelijk dat aspecten van computational thinking bij rekenen-wiskunde vaak op de stip liggen. Als je je daar bewust van bent, is het makkelijker om computational thinking een plek te geven in het onderwijs. Je hoeft daarvoor geen programmeeronderwijs te

geven. Belangrijke aspecten van computational thinking zitten in vrijwel iedere reken-wiskundeles. Immers in die lessen vragen we leerlingen gegevens te verzamelen, te analyseren en te visualiseren. Het interpreteren van een context betekent vaak dat kinderen de complexiteit van een probleem moeten reduceren. Als kinderen een manier bedenken om iets volgens een standaardwerkwijze aan te pakken gebruiken ze ook algoritmes. Leraren zouden die aspecten van computational thinking nog verder kunnen uitlokken door vragen te stellen die het verzamelen van gegevens, het analyseren, het reduceren van het probleem en het vinden van algoritmes uitlokken. Vragen als: 'kan het altijd zo?', 'hoe kan het efficiënter?' of 'waarom mag dat op deze manier?' kunnen deze manier van denken bij leerlingen stimuleren.

Maar is het dan zo eenvoudig dat we bij rekenen-wiskunde alle onderdelen van computational thinking aan bod kunnen laten komen? Uiteindelijk is het automatiseren (we bedoelen hier: het door een computer laten uitvoeren van een zichzelf herhalende taak) ook onderdeel van computational thinking, en zijn er uiteindelijk dus computers nodig. Hierbij komt het échte programmeren om de hoek kijken. Waarbij overigens het programmeren zelf, het gebruik van codes, instructies, ook nogal wat rekenen-wiskunde vraagt. Als bijvoorbeeld een figuur getekend moet worden door gebruik te maken van een programmeertaal, moet er nagedacht worden over hoeken, stappen, algoritmes. Hierbij kunnen leerlingen kennis en vaardigheden, die zij bij rekenen-wiskunde hebben opgedaan, inzetten.

Kortom, de voorbeelden in dit artikel laten zien dat het alledaagse reken-wiskundeonderwijs voldoende aanknopingspunten biedt om diverse aspecten van computational thinking aan bod te laten komen. Natuurlijk is er meer te beleven aan computational thinking, maar dat ligt na een goede voorbereiding voor het oprapen.

* Op <http://curriculumvandetoekomst.slo.nl/21e-eeuwse-vaardigheden/digitale-geletterdheid/computational-thinking/> geeft de SLO een voorbeeld van een leerplan computational thinking