

Voorbeeld van een opdrachtenboek met “Computational-Thinking-opdrachten” aansluitend bij de methode Getal & Ruimte

Benny Aalders

Vincent Velthuisen

Versie: 4 juni 2018

In dit document staan voorbeeldopgaven die in een programmeerboek zouden kunnen staan dat wordt meegeleverd met een lesboek/een serie lesboeken. Bij wijze van voorbeeld hebben we voor de lesboekenserie Getal & Ruimte gekozen. Bij de opdrachten zijn niveaus aangegeven. Het meetniveau is ordinaal. Toewijzing van de niveaus is op dit moment enkel gebaseerd op onze eigen inschatting. De niveaus dienen louter om een vorm van structuur zichtbaar tijdens het ontwikkelingsproces.

Niveaus I–II Dit zijn opdrachten op instapniveau; i.e.: vertrouwd raken met de mogelijkheden en methodes. Dit zou voor vmbo het eindniveau kunnen zijn. Bij deze opdrachten is het programmeerniveau relatief laag of nul. Het draait hier meer om het gebruiken van de beschikbare technologie.

Niveau III Dit zijn opdrachten waar programmeren een grotere rol speelt. Dit is voor havo/VWO bedoeld. Bij het maken van deze opdrachten heeft de leerling alle basiskennis nodig over het bijbehorende hoofdstuk uit zijn lesboek.

Niveau IV Deze opdrachten gaan wat dieper in op de stof. Over het algemeen wordt hier getest wat een programma doet met ongebruikelijke invoer.

Een voorbeeld: In **P6(b)** wordt gevraagd wat er gebeurt wanneer je $\text{ggd}(-2, 4)$ met je programma probeert uit te rekenen. Een leerling moet dan nadenken over de precieze definitie van de ggd en of $\text{ggd}(-2, 4)$ wel betekenis heeft. Daarnaast moet hij nadenken over zijn code: “Krijg ik een foutmelding of een foute uitvoer?”; Waarom zegt mijn programma $\text{ggd}(-2, 4) = 2$, terwijl $\text{ggd}(-2, 4)$ niet gedefinieerd is?”

Een leerling moet zich bij deze opdrachten verdiepen in de betekenis van wiskundige begrippen en in de precieze werking van zijn code. Dit is waarschijnlijk alleen voor gevorderde leerlingen weggelegd.

Niveaus V–VI Diepergaande opdrachten over de geschreven code. Hierbij ligt de nadruk meer op het coderen dan op de wiskundige onderwerpen waarvoor de code gebruikt wordt.

Hoewel de opdrachten allemaal betrekking hebben op een bepaald hoofdstuk/bepaalde paragraaf, zou je ervoor kunnen kiezen om sommige opdrachten later in het jaar te behandelen.

Een voorbeeld: In opdracht **P6(b)** wordt gevraagd wat er gebeurt wanneer je $\text{ggd}(-2, 4)$ met je programma probeert uit te rekenen. Hier kan naar gekeken worden op het moment dat negatieve getallen behandeld worden. Dit hoeft niet per se tijdens het hoofdstuk over de ggd en het kgv te gebeuren.

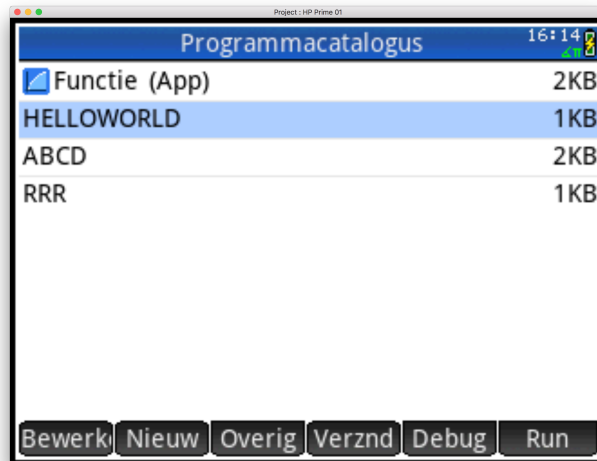
Legenda:

I Opdracht 1 uit het reguliere tekstboek.

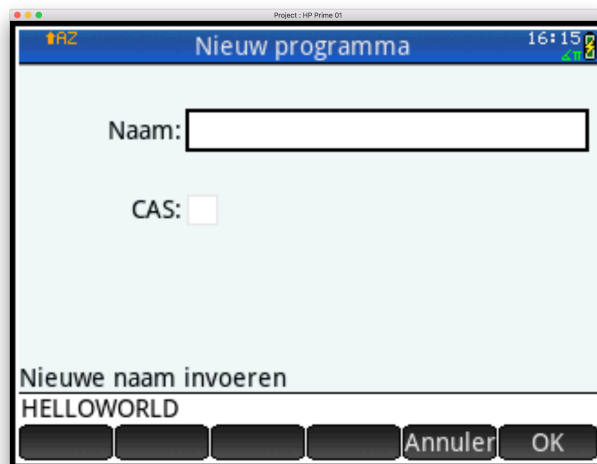
II P6 Opdracht 6 uit het programmeren-boek. Deze opdracht is van niveau 2.

Hello world!

In deze sectie vind je een basale uitleg van het programmeren met de HP-Prime. We schrijven een “Hello world!” en geven een aantal gebruikerstips. Zet je apparaat aan en ga naar ‘Program’ ([shift] [1]). Je ziet nu je programmacatalogus met daarin de programma’s ISDELER en HELLOWORLD.



Door op ‘Nieuw’ te drukken (touchscreen) kun je een nieuw programma aanmaken. De invoer zie je onderaan het scherm en niet achter naam.



Elk programma begint met `EXPORT <NAAM>()`. Hier is <NAAM> de naam van het programma. Tussen de haakjes geef je de parameters van het programma (in dit geval geen).

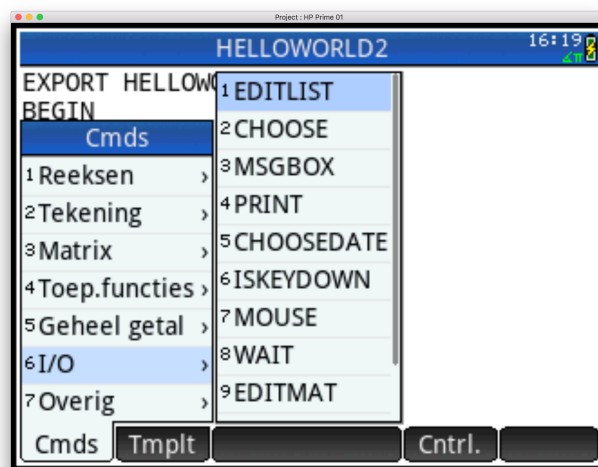
```

1  EXPORT HELLOWORLD()
2  BEGIN
3      LOCAL T := "Hello world!";
4      PRINT(T);
5  END;
6

```

Programma 1: *HELLOWORLD*

Zoals je ziet declareer je een variabele met `[:=]`. Maak je variabele altijd local. Aan T wordt de string "Hello world!" toegekend. Via Cmds (linksonder op je touchscreen) kom je bij I/O en PRINT.



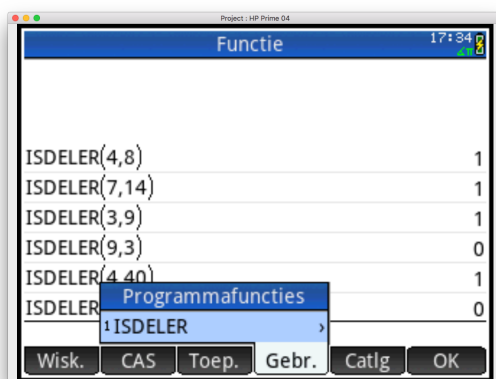
PRINT(T) zorgt ervoor dat het programma de waarde van T laat zien (in dit geval "Hello world!").

Getal & Ruimte

1 HAVO/VWO

Hoofdstuk 2.2 - De ggd en het kgv

Wanneer je een programma schrijft hoef je niet altijd alle problemen zelf op te lossen. Soms kun je bestaande programma's (her)gebruiken. Het is verstandig om zeker te weten wat je hulpprogramma precies doet, wat je moet invoeren en welke uitvoer je kunt verwachten.



Figuur 1: Open je eigen programma's door op "gereedschapskist", "Gebr." te drukken.

I P1

In je programmacatalogus staat het programma ISDELER.

a In Figuur 1 zie je hoe je programma's kan aanroepen. Beschrijf, op basis van wat je hier ziet, wat het programma ISDELER doet.

b Controleer met ISDELER of 2 een deler is van 6.

c Controleer met ISDELER of 2 een deler is van 7.

d Controleer met ISDELER je antwoord op vraag **14**.

Je gaat nu een programma schrijven dat gebruik maakt van het sleutelwoord IF. Je geeft een voorwaarde, die waar (true, 1) of onwaar (false, 0) moet zijn. Dit kan bijvoorbeeld door een vergelijking te schrijven. Gebruik hiervoor de symbolen onder Shift + 6. Wanneer de voorwaarde waar is dan wordt het THEN deel van de code uitgevoerd, wanneer het niet waar is het ELSE deel. Als je het ELSE deel niet nodig hebt, dan mag je het weglaten zoals in Programma 3.

II P2

Schrijf een programma dat

```

1 IF voorwaarde THEN
2   // Dit gebeurt als de voorwaarde TRUE is .
3 ELSE
4   // Dit gebeurt als de voorwaarde FALSE is .
5 END;

```

Programma 2: *IF-THEN-ELSE*

```

1 IF voorwaarde THEN
2   // Dit gebeurt als de voorwaarde TRUE is .
3 END;

```

Programma 3: *IF-THEN*

- vraagt om twee getallen a en b .

Als je deze getallen hebt ingevoerd, dan moet het programma zeggen

- “ a is een deler van b ”, **of**
- “ a is geen deler van b ”.

Voor de volgende opdracht vragen we de gebruiker slechts om één getal. Je gaat zoeken naar alle delers van dat getal. Dit kun je doen door gebruik te maken van een FOR-lus. De code in een FOR-lus wordt herhaald waarbij de variabele met de “stapper”-rol alle waarden in een bepaald bereik krijgt.

```

1 LOCAL deler ;
2 FOR deler FROM 1 TO 3 DO
3   PRINT deler ;
4 END;

```

Programma 4: *FOR-lus*

II P3 Schrijf een programma dat alle delers van een getal weergeeft.

III P4 Je gaat nu proberen om een programma te schrijven waarmee je de ggd van twee getallen kunt berekenen.

- Beschrijf alle stappen die je programma moet doorlopen.
- Geef per stap aan hoe je je rekenmachine dit zou laten doen.
- Probeer nu de stappen in de goede volgorde te zetten.
- Schrijf een programma dat de ggd van twee getallen geeft. Noem dit programma “GGD”.

III P5 Je gaat nu het programma dat je in opdracht **P4(d)** hebt geschreven, testen.

- Controleer je code door te kijken of je de juiste antwoorden krijgt bij opdrachten **15** en **16**.

- b** Controleer of $\text{ggd}(12, 30) = \text{ggd}(30, 12)$. Controleer op dezelfde manier vier opgaven van vraag **15** en **16**.
- c** Wat geeft jouw code voor $\text{ggd}(1, 8)$, $\text{ggd}(9, 9)$ en $\text{ggd}(1, 1)$?
- d** Kun je met jouw code vraag **18(f)** ook oplossen? Leg uit waarom wel of niet.
- e** Gebruik het testprogramma GGD-Test dat je van je docent hebt gekregen om je programma GGD te testen.

- IV P6** Deze vraag gaat over het programma dat je geschreven hebt in **P4(d)**. We kijken wat jouw programma geeft voor $\text{ggd}(a, b)$, voor verschillende waarden van a en b .
- a** Wat gebeurt er als $a = 0$ of $b = 0$? En wat als $a = 0$ en $b = 0$?
 - b** Wat gebeurt er als $a < 0$ of $b < 0$? En wat als $a < 0$ en $b < 0$?
 - c** Wat gebeurt er als a of b geen geheel getal is? En wat als ze dit allebei niet zijn?
 - d** Wat is volgens jouw programma $\text{ggd}(6, 9\frac{1}{5})$?

- II P7** Wat zou er gebeuren als je een programma alle veelvouden van twee getallen laat geven?

- III P8** Je gaat nu proberen om een programma te schrijven waarmee je het kgv van twee getallen kunt berekenen.
- a** Schrijf een programma dat lijst met veelvouden van de twee getallen weer-geeft. Denk na over een nuttig aantal veelvouden.
 - b** Schrijf een programma dat het kgv van twee getallen geeft.

- III P9** Je gaat nu het programma dat je in opdracht **P8(b)** hebt geschreven, testen.
- a** Controleer je code door te kijken of je de juiste antwoorden krijgt bij vraag **17** en **18(a,c)**.
 - b** Controleer of $\text{kgv}(8, 12) = \text{kgv}(12, 8)$. Controleer dit ook voor alle opgaven van vraag **17** en **18(a,c)**.
 - c** Wat geeft jouw code voor $\text{kgv}(1, 8)$, $\text{kgv}(9, 9)$ en $\text{kgv}(1, 1)$?
 - d** Kun je met jouw code vraag **18(e)** ook oplossen? Zo nee, leg uit waarom niet.

- IV P10** Deze vraag gaat over het programma dat je geschreven hebt in **P8(b)**. We kijken wat jouw programma geeft voor $\text{kgv}(a, b)$, voor verschillende waarden van a en b .
- a** Wat gebeurt er als $a = 0$ of $b = 0$? En wat als $a = b = 0$?
 - b** Wat gebeurt er als $a < 0$ of $b < 0$? En wat als $a < 0$ en $b < 0$?

c Wat gebeurt er als a of b geen geheel getal is? En wat als ze dit allebei niet zijn?

V P11

Je gaat nu kijken tussen de overeenkomsten van je programma van **P4(d)** en **P8(b)**. Wat zijn de overeenkomsten tussen de programma's?

VI P12

In deze opdracht ga je nadenken over de gebruiksvriendelijkheid van je programma's.

a Schrijf een programma dat de ggd **en** het kgv van twee getallen geeft.

b Denk na over de output van je programma. Snapt iemand die de code van jouw programma niet kent hoe de getallen ingevoerd moeten worden? Is duidelijk welke van de twee uitkomsten de ggd en welke het kgv is?