

Inleiding programmeren natuurkundigen

Ragnar Groot Koerkamp

11 juni 2015

Contents

1	Overzicht	1
2	Visual studio	1
3	Basis	1
4	Console in- en uitvoer	2
5	Variabelen	3
6	For	3
7	If	4
8	While	5
9	Arrays [optioneel]	6

1 Overzicht

- Inleidend praatje ook hier?
- Voorbeeld code

2 Visual studio

Voor Manon

3 Basis

opmerkingen over

- puntkomma
- whitespace
- includes

- std
- main, return value

```
1 // to read and write from/to the console
2 #include <iostream>
3
4 // to acces functions in the standard library
5 using namespace std;
6
7 // the main function; the program starts here
8 int main(){
9     return 0; // no error
10 }
```

4 Console in- en uitvoer

Wanneer je het programma start, zie je steeds een zwart scherm verschijnen. Dat heet de *console* of *terminal*. Daar kunnen we tekst neerzetten of juist van lezen. Het neerzetten van tekst kan met het commando `cout`, oftewel, Console-out. Je kan achter de `cout` `<<` bijvoorbeeld een string of een getal zetten. Ook is er het speciale commando `endl`, dat voor *endline* staat. Hiermee beëindig je dus een regel. Het is handig om te onthouden dat je de tekst 'in de console stopt' en de pijltjes dus richting `cout` staan.

```
1 #include <iostream>
2 using namespace std;
3
4 int main(){
5     // print 'Hello world!' to the console
6     // together with a newline character
7     cout << "Hello world!" << endl;
8 }
```

Het is soms ook handig om juist invoer van de gebruiker te lezen. Daarvoor hebben we `cin`, console-in. Hieronder lezen we een getal in, wat we in de variabele (zie het volgende hoofdstuk) `leeftijd` stoppen. De pijltjes staan hier naar rechts, omdat het getal uit de console komt, en in de variabele wordt gestopt.

```
1 #include <iostream>
2 using namespace std;
3
4 int main(){
5     cout << "Hoe oud ben je?" << endl;
6     int leeftijd;
7     cin >> leeftijd;
```

```
8     cout << "Je bent " << leeftijd << " jaar oud" << endl;  
9 }
```

5 Variabelen

Er zijn meerdere soorten variabelen. De belangrijkste zijn getallen en strings. Een getal kan of geheel zijn, of reëel. In het eerste geval gebruiken we een `int`, wat staat voor *integer*. Als een getal niet altijd geheel is, hebben we een `double` nodig. Een `double` kan elke mogelijke waarde aannemen, dus bijvoorbeeld ook π .

Een `string` is een stuk tekst. In computertaal is dat gewoon een rij van tekens.

```
1 #include <iostream>  
2 #include <string> // nodig voor strings  
3 using namespace std;  
4  
5 int main(){  
6     /*  
7      * Weinig inspiratie hier  
8      * TODO:  
9      * + - * /  
10     * integer division  
11     * (%)  
12     */  
13     int a = 10;  
14     cout << "a = " << a << endl;  
15     cout << "2*a = " << 2*a << endl;  
16  
17     int b = 3.5; // een int is altijd geheel  
18     cout << "b = " << b << endl;  
19  
20     double c = 3.5; // een double kan elk getal zijn  
21     cout << "c = " << c << endl;  
22     cout << "c kwadraat = " << c*c << endl;  
23  
24     string s = "Hello world!";  
25     cout << s << endl;  
26  
27     return 0; // no error  
28 }
```

6 For

Als je een stukje code vaker dan één keer wilt uitvoeren, kan je hem natuurlijk een aantal keer onder elkaar kopiëren en plakken. Als je echter heel vaak (≥ 3 keer)

iets wilt doen, is dat niet meer handig. Daarvoor is de `for`-loop. Deze herhaalt iets een bepaald aantal keer. Hieronder staat een voorbeeldje.

```
1 #include <iostream>
2 using namespace std;
3
4 int main(){
5     cout << "drie hoeratjes:" << endl;
6     for(int i = 0; i < 3; i++)
7         cout << "hoera" << endl;
8
9     cout << "tellen tot 10:" << endl;
10    for(int getal = 0; getal <= 10; getal++)
11        cout << getal << endl;
12
13    return 0;
14 }
```

Als je dit intikt en uitvoert, zie je dat er eerst drie keer 'hoera' wordt geschreven, en daarna de getallen van 0 tot en met 10. De werking van een `for`-loop is als volgt:

```
1 for(initialisatie; conditie; verhoging){
2     code die herhaalt word
3 }
```

In de *initialisatie* wordt meestal een variabele geïnitieerd, zoals hier `int i=0`. In de *conditie* kijken we steeds of we moeten doorgaan met de volgende iteratie, of dat we moeten stoppen met herhalen. Zolang de conditie waar is gaan we door, en als hij niet meer waar is stoppen we met herhalen. Hier gaan we dus alleen maar door wanneer `i` kleiner of gelijk is dan 3. In de *verhoging* verhogen we de variabele waarover we lopen. Meestal verhogen we hem steeds met 1 tegelijk maar dat hoeft niet per se.

7 If

Tot nu toe lag het altijd precies vast welke code werd uitgevoerd en hoe vaak dat gebeurde. Het is ook mogelijk om iets alleen uit te voeren als aan een bepaalde voorwaarde is voldaan. Daarvoor gebruiken we het `if`-statement. De syntax is

```
1 if(conditie){
2     code wanneer conditie waar (true) is;
3 } else {
4     code wanneer conditie niet waar (false) is;
5 }
```

Dit kunnen we bijvoorbeeld gebruiken in het volgende voorbeeld, waarin de uitvoer afhangt van de invoer van de gebruiker.

```
1 #include <iostream>
2 using namespace std;
3
4 int main(){
5     cout << "voer je leeftijd in:" << endl;
6     int leeftijd;
7     cin >> leeftijd;
8     if(leefteijd >= 18){
9         cout << "Je bent volwassen!" << endl;
10    } else {
11        cout << "Je bent nog een kind!" << endl;
12    }
13
14    return 0;
15 }
```

8 While

Tot slot hebben we nog de `while`-loop. Deze lijkt een beetje op de `for`-loop, maar is toch anders. Bij een `for`-loop weten we meestal van tevoren al hoe vaak we de code willen herhalen. Als we dat niet precies weten, is een `while`-loop handig. We herhalen de code nu net zo lang tot conditie onwaar wordt in

```
1 while(conditie){
2     code die herhaald wordt;
3 }
```

In de code hieronder zie je een voorbeeld. We beginnen hier met $n = 0$, en vervolgens schrijven we steeds n^2 op, net zo lang tot n^2 groter wordt dan 1000.

```
1 #include <iostream>
2 using namespace std;
3
4 int main(){
5     cout << "alle kwadraten onder 1000" << endl;
6
7     int n = 0;
8     while(n * n <= 1000){
9         cout << (n * n) << endl;
10        n = n+1;
11    }
12
13    return 0;
14 }
```

9 Arrays [optioneel]

```
1  #include <iostream>
2  using namespace std;
3
4  int main(){
5
6      int kwadraten[10];
7
8      for(int i = 0; i < 10; i++){
9          kwadraten[i] = i*i;
10     }
11
12     int som = 0;
13     for(int i = 0; i < 10; i++){
14         som += kwadraten[i];
15     }
16
17     cout << "de som is " << som << endl;
18
19     return 0;
20 }
```
