Les C-04 Programmeren

4.1 Programmeertalen

In les B-03 is uitgelegd hoe de processor van een moderne computer geprogrammeerd dient te worden: **programmacode** wordt omgezet in **assemblercode**, die vervolgens weer omgezet wordt in **machinecode**. Deze machinecode vertelt de processor precies welke handelingen er moeten worden verricht met de bitstroom die door de processor verwerkt wordt.

De gebruiker van de computer heeft meestal niets te maken met assemblercode en machinecode maar gebruikt programmeertalen die makkelijker te begrijpen zijn dan de instructies die tot op de registers nauwkeurig vertellen wat de processor moet doen. Net als een natuurlijke taal bestaat een programmeertaal uit een vastgelegde **semantiek** (betekenis) en **syntaxis** (wijze van gebruik, grammatica).

Een eenvoudig voorbeeld van een programmeertaal is de programmeertaal **BASIC** (**Beginners All-Purpose Symbolic Instruction Code**). Deze programmeertaal is ontstaan in 1970 en werd tot ver in de jaren 80 gebruikt. De taal was gemaakt voor gebruik onder het DOS besturingssysteem. Een voorbeeld van een BASIC programma, waar veel computergebruikers in de jaren 70 en 80 mee zijn begonnen, is:

```
10 PRINT    "HALLO DAVID"

20 GOTO     10
```

Bij dit programma wordt je op het beeldscherm van je computer getrakteerd met een alsmaar doorgaande reeks begroetingen. Het voorbeeld maakt ook duidelijk dat de taal BASIC een eigen semantiek en syntaxis heeft.

Een wat gecompliceerder voorbeeld berekent de tafel van …:

```
10 REM      Dit programma berekent vermenigvuldigingstafels.

20 PRINT    "Geef het gewenste getal waarvan u de tafel wilt berekenen:"

30 INPUT    Getal

40 LET      Teller = 1

50 LET      Uitkomst = Teller * Getal

60 PRINT    Teller, "*", Getal, "=", Uitkomst

70 LET      Teller = Teller + 1

80 IF       Teller = 11   THEN   GOTO 90   ELSE   GOTO 50

90 END
```

OPDRACHT

Opdracht 4.1
Wat doet het onderstaande BASIC programma ?

```
10 REM      Dit programma ……...

20 PRINT    Geef het gewenste getal:

30 INPUT    Getal

40 LET      Uitkomst = 1

50 LET      Teller = Getal

60 LET      Uitkomst = Uitkomst * Teller

70 LET      Teller = Teller - 1

80 IF       Teller = 1   THEN   GOTO 90   ELSE   GOTO 60

90 PRINT    Uitkomst

100 END
```

Karakteristieke eigenschappen van de programmeertaal BASIC zijn de regelnummering die achtereenvolgens moet worden doorlopen en de GOTO instructie waarmee naar een ander regelnummer kan worden “gesprongen”.

Naast de programmeertaal BASIC waren ook de programmeertalen **PASCAL** en **C** bekende talen die onder het besturingssysteem DOS werden gebruikt.

Het programma dat de tafel van … berekent ziet er in PASCAL als volgt uit:

```
program tafel;
{ afdrukken van een tafel naar keuze }
var       getal, teller, uitkomst: integer;
begin
   writeln(‘Van welk getal wil je de tafel afdrukken?’);
   readln(getal);
   teller := 1;
   while teller <= 10 do
      begin
         uitkomst := teller * getal;
         writeln(teller,’*’,getal,’=’,uitkomst);
         teller := teller + 1;
      end;
end.
```

Met de komst van de Grafische User Interface (GUI), in het bijzonder het besturingssysteem MS Windows als vervanger van MS DOS, ontstond de behoefte om ook op een meer grafische manier programma's te ontwerpen.

BASIC kreeg als opvolger **VISUAL BASIC**, PASCAL kreeg als opvolger **DELPHI** en C kreeg als opvolger **C++.** Ook de programmeertaal **JAVA** ontstond.

Met de komst van deze programmeertalen is ook de manier van programmeren veranderd. Programma's bestaan niet meer uit één groot geheel van instructies maar zijn opgebouwd uit **objecten, methoden en klassen**. We noemen deze programmeertalen dan ook **object georiënteerde programmeertalen**.

Aan de hand van een voorbeeld wordt uitgelegd wat object georiënteerd programmeren inhoudt. We bekijken het spel Yahtzee, waarvan het scoreformulier hieronder is weergegeven. Bij het spel Yahtzee krijgt elke speler steeds een beurt waarin hij 3 keer met 5 dobbelstenen mag gooien. Dat levert een score op die wordt ingevuld op het scoreformulier.

	1	2	3	4	5	6
Enen						
Tweeën						
Drieën						
Vieren						
Vijven						
Zessen						
totaal boven						
3 dezelfde (3 of a kind)						
4 dezelfde (carré)						
3 + 2 dezelfde (full house)						
kleine straat						
grote straat						
5 dezelfde (Yahtzee)						
totaal onder						
totaal boven						
bonus (boven >= 63)						
Totaal						

Bij object georiënteerd programmeren wordt een **object** omschreven door een aantal eigenschappen. In het dobbelspel "Yahtzee" is de "dobbelsteen" een object met als eigenschap "het aantal ogen". Ook een "worp" is een object met als eigenschap de ogenaantallen van vijf geworpen dobbelstenen.
Een **methode** doet iets met een object. De methode "gooi" kent een nieuw aantal ogen toe aan de dobbelsteen. Aangezien er in het dobbelspel "Yahtzee" met vijf dobbelstenen dient te worden gegooid worden er vijf objecten "dobbelsteen" gebruikt die allemaal de methode "gooi" kunnen ondergaan.
Een speler van het dobbelspel "Yahtzee" moet steeds beslissen op welke manier hij zijn worp wil laten tellen. Het is daarbij van belang dat het programma nagaat welke mogelijke scores een worp oplevert op het scoreformulier. Daarbij controleert het programma bijvoorbeeld hoeveel dobbelstenen uit een worp tot de **klasse** "5 ogen" behoren.

4.2 Generaties programmeertalen

De ontwikkeling van machinecode naar programmacode is parallel aan de ontwikkeling van de computer verlopen. De eerste computers moesten nog handmatig worden ingesteld. Een programmeur moest weten met welke binaire code de machine moest worden bediend. Machinecode wordt ook wel een **eerste generatie** programmeertaal genoemd.

Omdat het onthouden van instructies (ADD, MOVE) mensen nu eenmaal gemakkelijker afgaat dan het onthouden van binaire codes (01100010) is assemblercode ontwikkeld. Assemblercode wordt ook wel een **tweede generatie** programmeertaal genoemd.

Talen van de eerste en tweede generatie waren machineafhankelijk. Afhankelijk van het ontwerp van de processor moest deze worden geprogrammeerd. Deze talen worden daarom ook wel **machinegerichte programmeertalen** genoemd.

Naast de programmeertaal BASIC waren ook de programmeertalen PASCAL en C bekende talen die onder het besturingssysteem DOS werden gebruikt. Deze talen worden ook wel **derde generatie** programmeertalen genoemd. Compileerprogramma's ("**compiler**s") zetten deze programmeertalen om in assemblercode, die door de processoren kan worden verwerkt.

Vierde generatie programmeertalen zijn ontwikkeld vanuit de object georiënteerde talen. Voorbeelden zijn de instructies die in Micro Soft Office zijn te geven om adresgegevens uit een Excel werkblad te halen en in een Word document te plaatsen. Op gemakkelijke wijze is te programmeren dat je objecten (MS Word documenten) aanmaakt op basis van andere objecten (de records uit je MS Excel werkblad). In MS Office zorgt Visual Basic for Applications (VBA) voor deze mogelijkheden. Ook Structured Query Language (SQL) is een voorbeeld van een vierde generatie programmeertaal waarmee objecten (records uit MS Access) kunnen worden gemanipuleerd.

Bij programmeertalen van de derde en vierde generatie gaat het niet meer om het programmeren van de machine maar om het oplossen van een probleem. De programmeur hoeft niet na te denken over de wijze waarop de processor moet worden geprogrammeerd maar in welke stappen een probleemoplossing kan worden geprogrammeerd. We noemen programmeertalen van de derde en vierde generatie dan ook wel **functionele programmeertalen**.

4.3 Scripting

De laatste jaren komt ook **scripting** op. Om in te gaan op wat scripts zijn is het handig om eerst duidelijk te maken wat programma's precies zijn. Programma's worden gekenmerkt door het feit dat ze zelfstandig door een computer uitgevoerd kunnen worden. De programma's worden geleverd in een taal die de computer direct begrijpt.
Scripts daarentegen worden geleverd in hun eigen programmeertaal (bij scripts ook wel scripttaal genoemd). Scripts zijn dan ook gewoon te lezen en te bewerken. Dit heeft als gevolg dat elke keer dan een script wordt uitgevoerd deze eerst weer moet worden omgezet in machinecode. Dit gebeurt bij scripts door de **interpreter**. Het voordeel is dat scripts heel snel kunnen worden aangepast en gedistribueerd. Ook hoeven er voor verschillende systemen geen aparte versies te worden gemaakt; enkel een interpreter.

OPDRACHT

Opdracht 4.2
In het kaartspel BLACKJACK gaat het erom om met meerdere kaarten 21 te halen. De bank deelt kaarten uit. Elke kaart heeft een waarde:
- de aas heeft waarde 1 of 11.
- de twee t/m tien hebben hun eigen waarde.
- de plaatjes boer, vrouw, heer hebben waarde 10.

Jochem maakt een object georiënteerd programma voor het blackjackspel.
Benoem de beschreven onderdelen van het spel als object, klasse of methode:
a) kaart
b) plaatje
c) geef kaart
d) hand met kaarten
e) bepaal totale waarde

4. 4 "Programmeertalen" op het web

De laatste jaren is de betekenis van internet in ons dagelijks leven enorm toegenomen. Allerlei functies zoals het regelen van bankzaken, doen van aankopen en natuurlijk het communiceren is mogelijk omdat ook via het internet allerlei gecodeerde informatie toegankelijk en te gebruiken is.

HTML

HTML (Hyper Text Markup Language) is een scripttaal waarin webpagina's kunnen worden opgemaakt. Het is dus geen programmeertaal maar een opmaaktaal ook wel structuurtaal. HTML heeft echter wel een aantal eigenschappen van programmeertalen, zoals een eigen semantiek (verzameling instructies) en een eigen syntaxis (grammatica):

```
<html>

        <head>
        <title>Mijn eerste pagina</title>
        </head>

        <body>
        <B>Test</B>
        </body>

</html>
```

HTML is geen programmeertaal omdat met de HTML-code geen dynamische, interactieve gebeurtenissen plaats kunnen vinden. Willen we wel actie op de website, dan zijn er echte programmeertalen nodig. Hieronder worden er enkele beschreven.

HTML is gebaseerd op SGML. Een andere taal die is gebaseerd op SGML is XML. XML is een taal voor het gestructeerd opslaan van informatie en kom je tegenwoordig steeds meer tegen, o.a. in Word documenten.

Javascript en VB Script

JavaScript (eigenlijk ECMAScript) heeft weinig te maken met de bekende programmeertaal Java De meeste overeenkomst tussen JavaScript en Java vinden we in de syntax, deze leken in het begin nog veel op elkaar. Javascript is ontwikkeld door Netscape en Sun Microsystems, de bedenkers van Java toen in de jaren 90 bleek dat de programmeertaal Java voor velen te lastig en niet geschikt voor het internet was.

Scripts als Javascript kunnen worden ingebed in HTML-code:

```
<html>

        <head>
        <title>JavaScript Voorbeeld 1</title>
        </head>

        <body>

        <script language="JavaScript">
        document.write("Dit is mijn eerst script!");
        </script>

        </body>

</html>
```

Er bestaat zowel **server-side** Javascript (waarbij de programmacode op de server wordt uitgevoerd) als **client-side** Javascript (waarbij de programmacode door de browser bij de client wordt uitgevoerd). In het geval van client-side Javascript is het wel vereist dat de client een javascript-engine (interpreter) heeft geïnstalleerd, dit zit standaard in webbrowser, die de Javascript kan uitvoeren.
Javascripts moeten niet worden verward met Java applets. Dit zijn kleine Java programmaatjes op internet.

VB Script is net als Javascript een uitgeklede vorm van Visual Basic en levert vergelijkbare prestaties. Een voordeel van VB Script boven Javascript is dat de taal makkelijker te leren is.

ASP

ASP (Active Service Pages) is eigenlijk geen taal op zich maar een verzameling functies en technologieën waarmee webpagina's interactief gemaakt kunnen worden. De daadwerkelijke taal achter ASP is naar keuze CScript of VBScript. Hiermee wordt bedoeld dat aan de hand van acties van de gebruiker verschillende gegevens weergeven en gewijzigd kunnen worden, bijvoorbeeld uit een database. Denk hierbij bijvoorbeeld aan een zoekmachine of een winkelwagentje van een webwinkel.

Zoals een 'gewone' website uit HTML pagina's bestaat, bestaat een ASP website (of ASP applicatie) uit ASP pagina's. Een ASP pagina lijkt op HTML, maar bevat ook scripts die op de webserver worden uitgevoerd voordat het resultaat naar de browser op de computer van de client gestuurd wordt. Met ASP scripts kunnen bijvoorbeeld resultaten uit een database op de server worden bepaald en bij de client worden weergegeven.

Omdat de bewerkingen op de database op de server plaatsvinden en niet bij de client, blijft het dataverkeer tussen server en client beperkt tot het versturen van het resultaat van de bewerking naar de client.

ASP is ontwikkeld door Microsoft.

PHP en MYSQL

PHP (Personal Home Page / Hypertext Processor) is net als ASP een taal waarmee webpagina's interactief kunnen samenwerken met databases. Een verschil met ASP is dat PHP open source taal is.

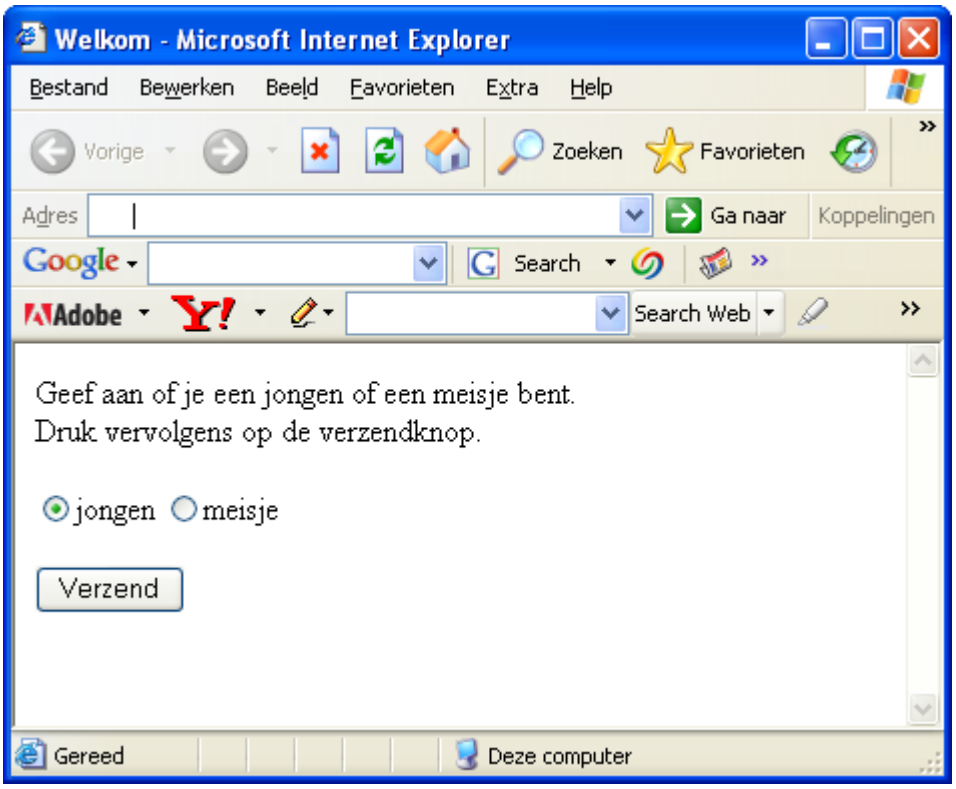
PHP werkt meestal samen met **MYSQL** databases en is eveneens een open source product. Aan MYSQL gegevensbestanden kan informatie worden onttrokken of toegevoegd met de vraagtaal SQL.

Net als andere scripts wordt PHP ingebed in de HTML code:

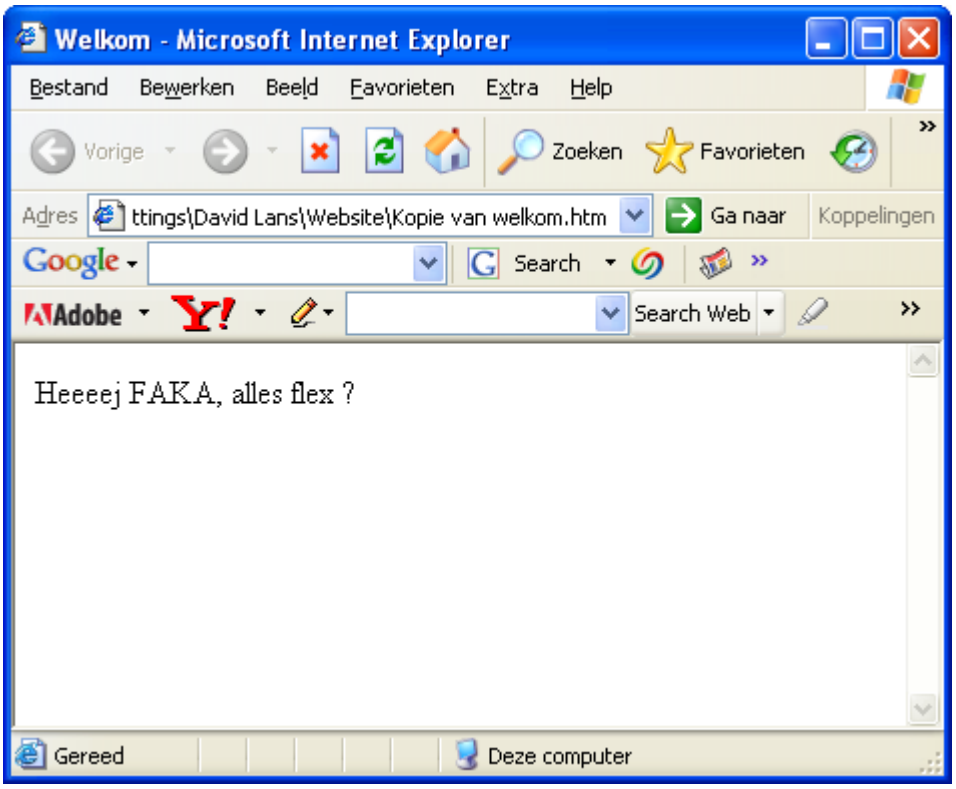
```
<html>
        <head>
        <title>PHP Voorbeeld 1</title>
        </head>
        <body>

        <?php
        print("Hoera, ook het PHP script werkt!");
        ?>

        </body>
</html>
```

Als je het bovenstaande script uit wilt proberen, moet je het bestand opslaan als "test.php" en op een server uploaden die PHP ondersteunt.

Het interessante van PHP is dat het websites interactief kan maken. Als je bijvoorbeeld in het onderstaande scherm invult of je een jongen of meisje bent en op de verzendknop drukt ….



verschijnt er een welkomstboodschap:



Het PHP script bij een interactieve webpagina wordt bij voorkeur niet in de webpagina zelf geplaatst. In kladblok of in Frontpage maak je een HTML-pagina aan:

```
<html>
        <head>
        <title>Welkom</title>
        </head>
        <body>
        Geef aan of je een jongen of een meisje bent.<br />
        Druk vervolgens op de verzendknop.<br />
        <br />

        <form action="welkom.php" method="post">
        <input type="radio" name="geslacht" value="jongen" checked="true" />jongen
        <input type="radio" name="geslacht" value="meisje" />meisje
        <br />
        <br />
        <input type="submit" value="Verzend" />
        </form>

        </body>
</html>
```

In deze pagina wordt een waarde (de waarde jongen of meisje die is gekozen door de gebruiker van de webpagina) verstuurd naar een PHP script. Het script zorgt voor het verschijnen van het antwoordformulier. Het PHP script ziet er als volgt uit:

```
<?php
$geslacht = $_POST["geslacht"];
if ($geslacht == "jongen")
{
print("Heeeej FAKA, alles flex ?");
}
if ($geslacht == "meisje")
{
print("Goedemiddag jongedame !");
}
?>
```

OPDRACHT

Opdracht 4.3
Kopieer de bovenstaande teksten naar kladblok.
Noem het html bestand “welkom.htm”.
Noem het php bestand “welkom.php”.
Upload de bestanden naar een provider die PHP ondersteunt.

Probeer het bestand welkom.htm uit.

De reden waarom de PHP scripts niet in de HTML code zelf staan is dat het veiliger is. In de PHP scripts staat immers informatie over de gegevensbestanden waaruit een website bestaat.

We besluiten met het voorbeeld waar we mee begonnen zijn: het afdrukken van een tafel. Hieronder staat de HTML code (tafel.htm) en het PHP script (tafel.php). In het HTML-bestand wordt een waarde gekozen, die naar het PHP script wordt verstuurd. Het script zorgt voor de verdere afhandeling en het afdrukken van de tafel.

```
<html>
	<head>
	<title>Tafel</title>
	</head>

	<body>
	Kies een geheel getal in tussen 1 en 10, en druk op de verzendknop:

	<form action="tafel.php" method="post">
	<select name="tafel" size="4">
	<option selected="selected">1</option>
	<option>2</option>
	<option>3</option>
	<option>4</option>
	<option>5</option>
	<option>6</option>
	<option>7</option>
	<option>8</option>
	<option>9</option>
	<option>10</option>
	</select>
	<br /><br />
	<input type="submit" value="Verzend" />
	</form>

	</body>

</html>
```

```
<?php
$tafel = $_POST["tafel"];
print("Dit is de tafel van $tafel:<br /><br /> \n");
for($i=1;$i<=10;$i++)
{
$uitkomst=$i*$tafel;
print("$i x $tafel = $uitkomst <br /> \n");
}
?>
```

4.5 Samenvatting

Met de ontwikkeling van de computer zijn ook de programmeertalen ontwikkeld. De programmeur van de eerste computers moesten nog zelf de processor instellen met machinegerichte instructies.
Bij de eerste generatie programmeertalen moest dat nog met binaire instructies (machinecode), bij de tweede generatie programmeertalen had de programmeur een verzameling instructies voor de processor (assemblercode).

Bij latere programmeertalen was kennis van de processor minder nodig. De programmeertalen waren erop gericht om stapsgewijs problemen op te lossen, werden steeds probleemgerichter. Derde generatie programmeertalen zijn onder andere:

- BASIC,
- PASCAL,
- C.

Een vertaalprogramma (compiler) vertaalt de programmacode van deze talen naar assemblercode. Met de komst van MS Windows werd ook de programmeeromgeving voor programmeertalen gebruiksvriendelijker en ontstonden er object georiënteerde programmeertalen. Dit zijn talen van de vierde generatie. Naast talen als:

- Visual Basic,
- Delphi,
- C++ en
- Java

ontstonden ook programmeertalen als:

- Visual Basic for Applications en
- SQL.

Een belangrijke eigenschap van object georiënteerde programmeertalen is dat programma's worden gezien als verzamelingen objecten, methoden en klassen.

Met de ontwikkeling van internet is een aantal scripttalen ontwikkeld. Deze talen lijken op programmeertalen maar zijn wezenlijk anders.

Van de volgende scripttalen moet je globaal weten waarvoor ze bestemd zijn:

- HTML
- JavaScript
- VB Script
- CGI
- PERL
- ASP
- PHP
- MYSQL

ANTWOORDEN

Opdracht 4.1
Bij een invoer 7 bepaalt het programma hoe groot:
7 x 6 x 5 x 4 x 3 x 2 x 1 is.
Dat is 7 !
Het programma bepaalt dus de faculteitwaarde van het ingevoerde getal.

Opdracht 4.2

a) kaart	= object
b) plaatje	= klasse
c) geef kaart	= methode
d) hand met kaarten	= object
e) bepaal totale waarde	= methode

Opdracht 4.3
-