



Programma Structuur Diagram - PSD

Doordat er verschillende programmeertalen zijn is de behoefte ontstaan naar een algemene manier om algoritmen/programma's weer te geven. Hiervoor wordt bijvoorbeeld het **Programma Structuur Diagram (Nassie-Shneiderman diagram)** gebruikt. Er bestaan nog meer schema technieken, zoals **Flow chart** die ook algoritmen schematisch weergeven.

De **PSD** schema's bestaan uit:

- Rechthoeken
- Driehoeken
- Tekst

Doordat schematechnieken als PSD onafhankelijk zijn van een programmeertaal, kunnen de schema's vertaald worden naar elke programmeertaal.

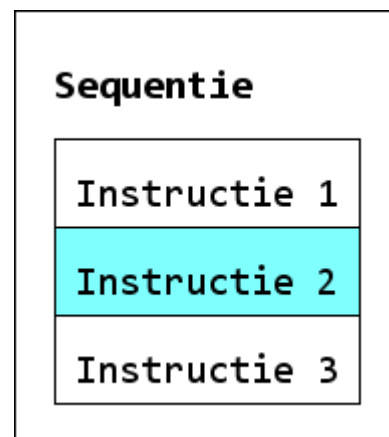


PSD structuren

Er zijn drie basisstructuren waar alle programma's uit bestaan. Deze structuren zijn:

1. Sequentie

In elke recht hoek staat één instructie. Instructies worden in volgorde, van boven naar beneden, achter elkaar uitgevoerd.

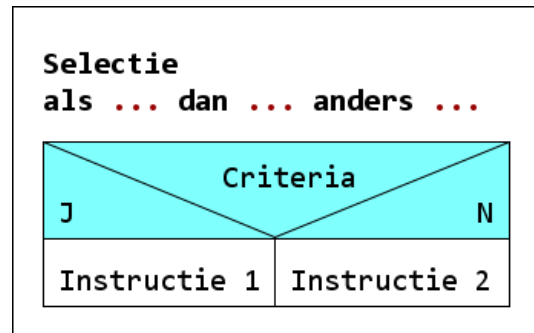




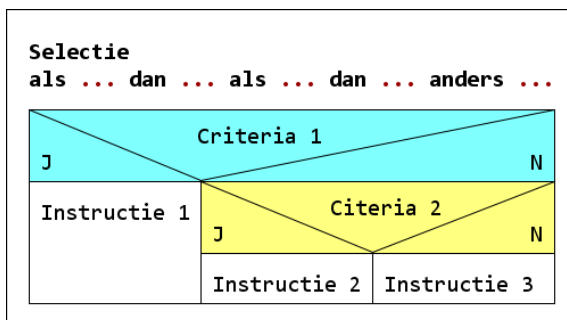
Programma Structuur Diagram - PSD

2. Selectie

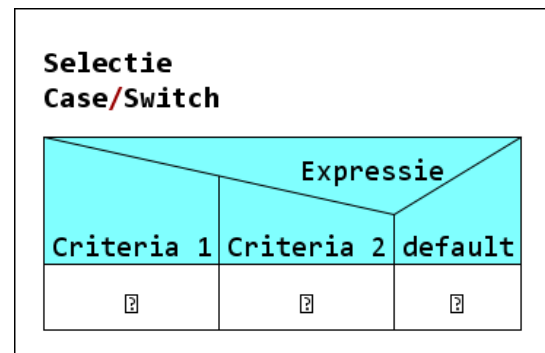
Selectie-structuren ook wel keuze structuren zoals *als ... dan ... anders ...* en de *case/switch* worden gebruikt om het algoritme een bepaalde weg te laten volgen.



Er kunnen meerdere als ... dan ... anders ... structuren in elkaar gezet worden. Dit wordt nesting genoemd.



Wanneer er meerder selectie structuren achterelkaar of genest gebruikt moeten worden, gebruikt men meestal een case ofwel switch structuur. Deze maken gebruik van meerdere opeenvolgende selectie structuren waardoor de code overzichtelijker en beter leesbaar wordt.



3. Iteratie

De iteratie-structuur ook wel herhaal structuur, wordt gebruikt om een serie opdrachten te herhalen tot dat een bepaalde voorwaarde niet meer wordt voldaan.

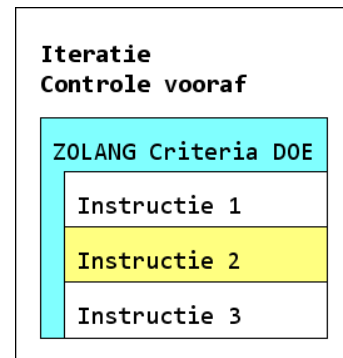


Programma Structuur Diagram - PSD

De drie iteratie-structuren zijn:

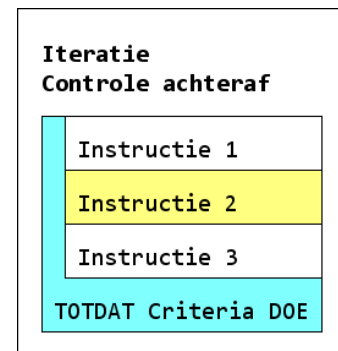
1. Met controle vooraf.

Voor dat het programma de herhaal lus in gaat wordt eerst gecontroleerd of de Criteria (voorwaarde) voldoet. Als dat het geval is worden de opdrachten in de lus uitgevoerd. Na de laatste opdracht wordt de voorwaarde opnieuw gecontroleerd. Zolang deze geldig blijft worden de opdrachten in de lus herhaalt. Zodra dit niet meer het geval is stopt de computer met de lus en gaat het met de volgende instructie na de herhaal lus verder indien deze er is.



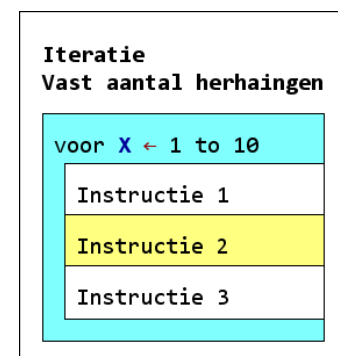
2. Met controle achteraf.

Bij controle achteraf worden de opdrachten van de herhaal lus minimaal één keer uitgevoerd. Pas na de eerste keer uitvoeren wordt er gecontroleerd of de voorwaarde nog geldig is en de instructies in de lus opnieuw herhaald kunnen worden.



3. Met een vast aantal herhalingen.

Herhaal lussen met vast aantal herhalingen heeft geen Criteria maar er wordt een teller gebruikt om het keren dat de herhaal lus wordt uitgevoerd te controleren.



De 2^{de} iteratie wordt eigenlijk afgeraden om te gebruiken daar het vaak voor problemen zorgt in de programma's.

Je gebruikt de iteratie, *met controle vooraf*, wanneer je niet weet hoeveel keren een lus moet worden herhaalt. Je weet wel wanneer het programma moet stoppen met de uitvoering van de lus.

Je gebruikt de iteratie, met vast aantal herhalingen, wanneer je precies weet hoeveel keren een lus moet worden herhaalt. Hou bij dit type wel rekening dat je de controle variabele niet veranderd of zomaar gebruikt binnen de code van de lus.



Programma Structuur Diagram - PSD

4. Subroutine

Een Subroutine kent bij PSD's twee schema's:

Eén voor de aanroep van de routine, *de eerste afbeelding*.

De andere voor de routine zelf, *de tweede afbeelding*. Subroutines zijn op zichzelf staande blokken met code die een bepaald doel of functie hebben binnen een programma. Programma's bestaan over het algemeen uit meerdere subroutines.

Het hoofdprogramma heeft rechte hoeken en subroutines hebben ronde hoeken bij een PSD.

Subroutine Aanroep

Subroutine (parameters)

Subroutine Code blok

ZOLANG Criteria DOE

Criteria

J

N

Instructie 1

Instructie 2

Instructie 3

Opdrachten

Bij het werken met PSD's wordt er gebruik gemaakt van een aantal basis instructies. Deze zijn:

- Variabelen
- LEES
- SCHRIJF
- ZOLANG ... DOE
- DOE ...TOTDAT
- VOOR ... TOT ... STAP ...

Variabelen

Dit zijn geheugenlocaties in het werkgeheugen van een computer waar informatie (data) opgeslagen wordt die voor het programma noodzakelijk is. Variabelen hebben een naam, in letters en cijfers aaneengesloten, die voor elke geheugenlocatie in de computer uniek is (net als een telefoonnummer).

```
var1 = 5  
var2 = 2  
var3 = var1 + var2
```

Je leest dit als var1 wordt vijf etc.

Het "=" teken is de *toekenningoperator* ook wel het toekenningsteken. In de instructies hierboven werd er namelijk een waarde toegekend aan een variablen. Bij sommige programmeertalen is het toekenningsteken uit ":=".



Programma Structuur Diagram - PSD

Invoer en uitvoer

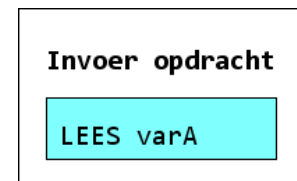
Alle programma's hebben invoer van gebruikers/omgeving nodig om te werk te kunnen doen. De programma's verwerken deze invoer en geven de gebruikers informatie terug, dit is de uitvoer.



Elke programmeertaal heeft instructies om gebruikers informatie te kunnen laten invoeren. Ook hebben ze instructies om informatie te kunnen tonen op bijvoorbeeld een beeldscherm.

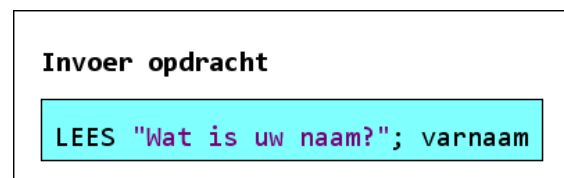
De opdracht **LEES**

De simpelste vorm van invoer is **LEES varA**. Hier wordt aan een gebruiker van het programma gevraagd om een waarde in te voeren voor de variabele varA.



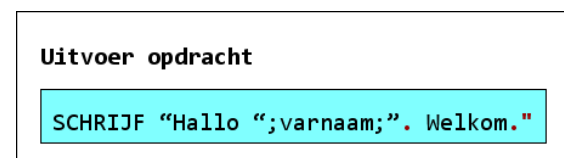
Het probleem is dat de gebruiker geen bericht op het scherm te zien krijgt dat het programma wacht op invoer van de gebruiker. Ook weet de gebruiker niet wat voor soort waarde hij/zij dient in te voeren. Dit is niet gebruikersvriendelijk.

Om een programma gebruikersvriendelijk te maken moet het programma de gebruiker uitleggen wat er van hem/haar verwacht wordt. In het voorbeeld hiernaast vraagt het programma aan de gebruiker wat het verwacht dat ingevoerd wordt. De ingevoerde waarde wordt dan opgeslagen in een variabele met de naam **varnaam** in het werkgeheugen van de computer.



De opdracht **SCHRIJF**

Met deze opdracht kan het programma tekst op het scherm van een computer laten zien aan de gebruiker. Er zijn verschillende variaties van deze opdracht mogelijk. In het voorbeeld hiernaast wordt de tekst **"Hallo "** gevolgd door de ingevulde naam en dan **". Welkom."** Op het scherm getoond. Het zou dus dit kunnen zijn als voor de naam **"Ingrid"** was ingevuld.



"Hallo Ingrid. Welkom."



Programma Structuur Diagram - PSD



De opdracht **ZOLANG ... DOE**

De **ZOLANG** opdracht wordt gebruikt in de iteratie structuur voor de herhaalstructuur met controle vooraf.

Dat gene wat na de **ZOLANG** opdracht staat is de criteria ook wel voorwaarde waaraan de lus moet voldoen om te blijven herhalen.

ZOLANG X <= 10 **DOE**

Iteratie Controle vooraf

ZOLANG X <= 10 **DOE**

SCHRIJF X

X = X + 1

De opdracht **VOOR TOT**

De **TOTDAT** opdracht wordt gebruikt in de iteratie structuur voor de herhaalstructuur met controle achteraf.

Dat gene wat na de **TOTDAT** opdracht staat is de criteria ook wel voorwaarde waaraan de lus moet voldoen om te blijven herhalen.

Iteratie Controle achteraf

SCHRIJF X

X = X + 1

TOTDAT X <= 10 **DOE**

De opdracht **VOOR TOT**

De **VOOR TOT** opdracht is een herhaal opdracht waarbij precies wordt aangegeven hoeveel keer een lus herhaald moet worden.

VOOR X = 1 **TOT** 10
SCHRIJF X

Iteratie Vast aantal herhalingen

VOOR X = 1 **TOT** 10

SCHRIJF X

Deze opdracht zal een lijst op het scherm afdrukken met de cijfers 1 t/m 10.



Programma Structuur Diagram - PSD

De opdracht **VOOR TOT STAP**

De **VOOR TOT STAP** heeft als verschil dat met deze opdracht je kunt aangeven in hoeveel stappen de *grenswaarde* bereikt moet worden.

VOOR X = 1 TOT 10 STAP 2
SCHRIJF X

Deze opdracht drukt de volgende reeks getallen op het scherm af: 1, 3, 5, 7 en 9.

Bij de het stap deel van de instructie kunnen ook negatieve getallen gebruikt worden, om bijvoorbeeld af te tellen van 10 naar 1. Dus de reeks: 10, 9, 8, ... , 2, 1.

De iteratie-structuur ook wel herhaal structuur, wordt gebruikt om een serie opdrachten te herhalen tot dat een bepaalde voorwaarde niet meer wordt voldaan.

Opdracht:

Maak een flowchart voor het programma Hoger-Lager en deze speelt met een gebruiker. Het programma moet een willekeurig getal kiezen dat de gebruiker moet proberen te raden binnen 10 beurten. Elke keer als de gebruiker raadt geeft het programma aan of het te raden getal hoger of lager is dan het getal dat de gebruiker heeft ingevoerd. Als de gebruiker het getal geraden heeft geeft het programma aan dat het getal geraden is in met het aantal beurten dat daar voor nodig was. Na 10 beurten geeft het programma het te raden getal met de melding dat de gebruiker het getal niet heeft kunnen raden in 10 beurten.

Iteratie
Vast aantal herhalingen
met positieve STAP

VOOR X = 1 TOT 10 STAP 2

SCHRIJF X

Iteratie
Vast aantal herhalingen
met negatieve STAP

VOOR X = 10 TOT 1 STAP -1

SCHRIJF X



Programma Structuur Diagram - PSD



Uitwerking opdracht:

