



PROGRAMMEREN INFORMATIKA voor HAVO / VWO nieuwe stijl



Augustus 2003



Inhoudsopgave

Inhoudsopgave	1
1 Het oplossen van een probleem	2
§ 1.1. Wat is de eerste stap bij het oplossen van een probleem?	2
§ 1.2 Oplossingsmethode in de vorm van een algoritme	2
§ 1.3. Omzetting van een algoritme in een PSD.	2
§ 1.4. Samenvatting.....	7
2 De computer.....	8
§ 2.1. Inleiding	8
§ 2.2. Geheugenplaatsen.....	8
§ 2.3. Teksten en getallen	9
§ 2.4. Rekenkundige bewerkingen en prioriteitsregels.	10
§ 2.5. Uitvoeropdracht / Schrijfopdracht.....	11
§ 2.6. Invoeropdracht / Leesopdracht.....	11
§ 2.7. Samenvatting.....	13
3. De computer kiest	14
§ 3.1. Inleiding	14
§ 3.2. Relatietekens	14
§ 3.3. Analyse van een probleem	15
§ 3.4 Samenvatting.....	18
4 De computer op herhaling	19
§ 4.1. Inleiding	19
§ 4.2. Probleemstelling.....	19
§ 4.3. Tellende lussen.....	19
§ 4.4. De niet tellende lus.	22
§ 4.5. Samenvatting.....	24
5 Modulair programmeren.....	25
§ 5.1. Inleiding	25
§ 5.2. Voordelen modulaire programma opbouw	25
§ 5.3. Top-down benaderingswijze	25
§ 5.4. Modulaire opbouw van een PSD.....	26
§ 5.5. Samenvatting.....	28
6 Werken met Visual Basic.	29
§ 6.1. Inleiding.	29
§ 6.2. Visual Basic verkennen.	29
7 Events en Codes	32
§ 7.1. Inleiding	32
§ 7.2. Code toevoegen aan control events	32
§ 7.3. PSD's omzetten in Visual Basic code	33
§ 7.4. Benaming van Controls	36
§ 7.5. Properties en codes.	38
§ 7.6. Methoden	38



1 Het oplossen van een probleem

§ 1.1. Wat is de eerste stap bij het oplossen van een probleem?

Deze stap is het stellen van het probleem (= *probleemstelling*).

De beschrijving van het probleem en van het gewenste resultaat dient zo duidelijk en zo volledig mogelijk te zijn, ten einde misverstanden te voorkomen. Immers, als de probleemstelling hier niet aan voldoet kunnen de daarop volgende stappen heel gemakkelijk tot een niet bedoelde uitkomst leiden, wat in de praktijk inderdaad nogal eens gebeurt.

Een voorbeeld van een probleemstelling is:

'Iemand wil elke avond trainen, behalve op de avond dat hij/zij een vergadering heeft'.

§ 1.2 Oplossingsmethode in de vorm van een algoritme

Aan de hand van de probleemstelling wordt onderzocht (geanalyseerd) welke bewerkingen tot de oplossing van het probleem kunnen voeren.

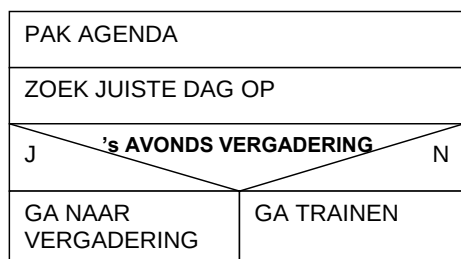
Een *algoritme* is een voorschrift dat stap voor stap aangeeft hoe een bepaald probleem opgelost kan worden.

VOORBEELD

1. Pak agenda.
2. Zoek juiste dag op.
3. 's Avonds vergadering?
4. Zo ja, ga naar vergadering.
5. Zo niet, ga trainen.

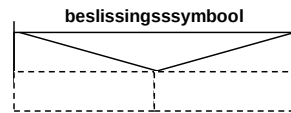
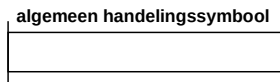
§ 1.3. Omzetting van een algoritme in een PSD.

Een programmastructuurdiagram of kortweg PSD is de presentatie van het algoritme in een overzichtelijk schema. Het heeft de vorm van een rechthoek. Het totale diagram is zelf opgebouwd uit een aantalsymbolen. Iedere symbool heeft een bepaalde betekenis. Daardoor is het mogelijk met slechts een summiere tekst in het symbool de gehele bedoeling ervan duidelijk te maken. Het gebruik maken van de computer vereist nu eenmaal dat de instructies kort en eenduidig worden omschreven. Voor het algoritme in § 1.2 ziet het programmastructuurdiagram er als volgt uit:





Als je naar dit programmastructuurdiagram kijkt, dan zie je de volgende symbolen:



Hoe hoog en breed elke rechthoek zal zijn, is afhankelijk van de hoeveelheid tekst die erin vermeld moet worden, en/of de grootte van de aangrenzende rechthoeken.

De belangrijkste uitgangspunten bij het samenstellen van een programmastructuurdiagram is de overzichtelijkheid en de duidelijkheid.

§ 1.3.1. Algemeen handelingssymbool



- Elk algemeen handelingssymbool heeft een *ingang* en een *uitgang*.
- Een algemeen handelingssymbool kom je binnen aan de bovenkant (ingang) en je verlaat dit symbool aan de onderkant (uitgang).
- Elk algemeen handelingssymbool bevat een uit te voeren opdracht.
- De opdrachten moet je uitvoeren in de volgorde waarin de symbolen van boven naar beneden in het programmastructuurdiagram staan.

1.1 Opdracht

Schrijf een programmastructuurdiagram (PSD) op voor iemand die zijn boeken en schriften in zijn schooltas moet doen en daarna naar school gaat, met behulp van de volgende opdrachten:

- ga naar school
- doe boeken in tas
- pak tas
- doe schriften in tas

Opmerking:

Let op de juiste volgorde van de opdrachten.

1.2 Opdracht

Maak een programmastructuurdiagram (PSD) voor iemand die zijn fiets pakt, daarna zijn schooltas op de fiets doet en vervolgens naar school rijdt.



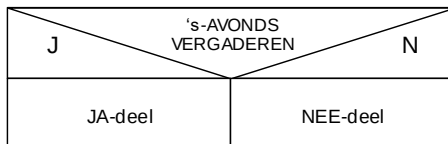
1.3 Opdracht

Welke opdracht is in het hieronder gegeven programmastructuurdiagram (PSD) niet juist?

DEUR IS DICHT
PAK EEN SLEUTEL
DOE DEUR OPEN
GA NAAR BINNEN

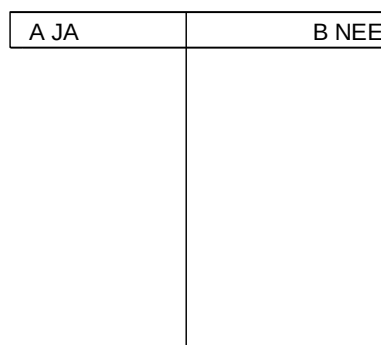
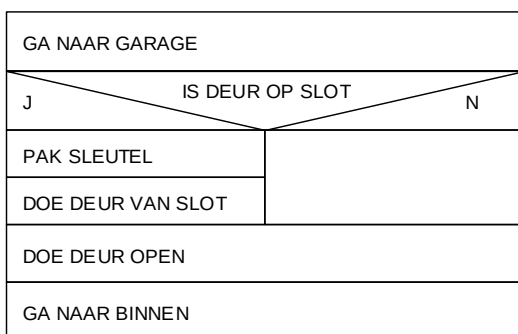
§1.3.2. Beslissingssymbool

- Het beslissingssymbool komt je binnen langs de bovenkant (ingang).
- Er zijn twee uitgangen, namelijk de ja- en de nee-uitgang, aangeduid met J en N.
- Elk beslissingssymbool bevat een vraag.
- De vraag moet zo gesteld worden dat het antwoord alleen ja of nee kan zijn.
- Indien het antwoord ja is, wordt verder gegaan met het JA-deel; indien het antwoord nee is wordt verder gegaan met het NEE-deel.
- Zowel het JA-deel als het NEE-deel kan opgebouwd zijn uit een of meer algemene handelingssymbolen en/of een of meer beslissingssymbolen of het kan leeg zijn.
- Aan het eind wordt de splitsing weer opgeheven.



1.4 Opdracht

Welke opdrachten moeten achtereenvolgens worden uitgevoerd volgens het onderstaande programmastructuurdiagram (PSD) als het antwoord op de vraag in het beslissingssymbool: A JA is; B NEE is.





1.5. Opdracht

Geef aan welke vragen in onderstaande beslissings-symbolen goed of fout zijn en waarom.



1.6 Opdracht

Schrijf een programmastructuurdiagram met de volgende vraag en opdrachten:

- ga met de trein
- pak de brommer
- rijd met de brommer
- band brommer lek?

1.7 Opdracht

Maak een programmastructuurdiagram waarin staat hoe je een auto moet starten. Gebruik hierbij:

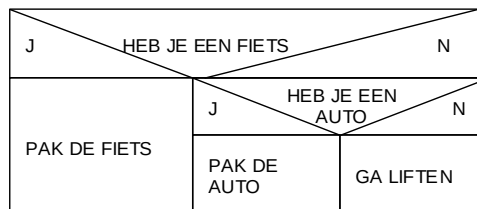
- start auto
- doe autodeur open
- doe sleutel in contact
- stap in auto
- doe autodeur van slot
- zit sleutel in contact?
- is autodeur op slot?

1.8 Opdracht

Welke *opdracht(en)* voer je in het onderstaande programmastructuurdiagram (PSD) uit als je in het bezit bent van:

- 1 een fiets;
- 2 een auto, maar niet een fiets;
- 3 een fiets en een auto?

Welke *opdracht(en)* voer je uit als je noch een fiets noch een auto hebt?



1.9 Opdracht

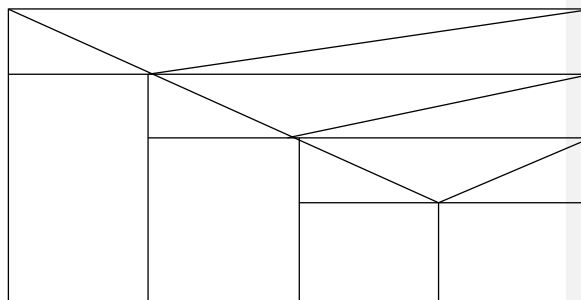
Een school hanteert voor de overgang van klas 2 naar klas 3 de volgende regels:

Een leerling wordt bevorderd bij minder dan 5 onvoldoendes en niet bevorderd bij meer dan 6 onvoldoendes.

Een leerling met 5 of 6 onvoldoendes wordt bevorderd als het totaal aantal punten op zijn rapport 60 of meer is.

Vul hiernaast in het PSD de volgende vragen en opdrachten in:

- bevorder leerling (2 keer gebruiken)
- bevorder leerling niet (2 keer gebruiken)
- minder dan 5 onvoldoendes?
- 60 punten of meer?
- meer dan 6 onvoldoendes?

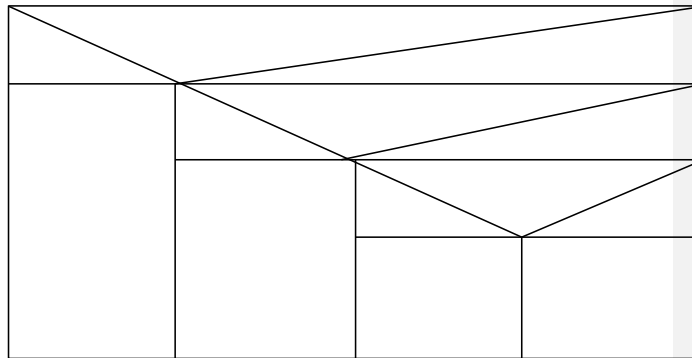




1.10 Opdracht

Een winkel hanteert tijdens de uitverkoop de volgende kortingsregels:

- 1 50 % korting op artikelen waarvan slechts 1 exemplaar in voorraad is;
- 2 40 % korting op artikelen waarvan 2 tot en met 5 exemplaren in voorraad zijn;
- 3 10 % korting op artikelen waarvan 6 of meer exemplaren in voorraad zijn, behalve wanneer ze langer dan een jaar in het assortiment zitten: in dat geval is de korting 30%.



Het is de bedoeling dat het bovenstaande PSD het juiste kortingspercentage voor een willekeurig artikel vaststelt. Vul hiervoor de vragen en opdrachten in. (De vragen en opdrachten moet je zelf formuleren.)

1.11 Opdracht

Een bedrijf hanteert bij een sollicitatieprocedure voor de functie van secretaresse de volgende selectiecriteria: leeftijd niet jonger dan 21 jaar; in het bezit van een m.e.a.o.- of Schoevers-diploma.

Schrijf van deze sollicitatieprocedure een PSD. Gebruik de volgende vragen en opdrachten:

- Schoeversdiploma?
- m.e.a.o.-diploma?
- jonger dan 21 jaar?
- neem aan (2 keer gebruiken)
- neem niet aan (2 keer gebruiken)

1.12 Opdracht

Je studeert elke avond, behalve op zaterdagavond. Dan ga je met je vriend(in) dansen, tenzij er een mooi programma op de t.v. is. Is je vriend(in) verhinderd en er is geen mooi t.v.-programma, dan ga je naar de bioscoop. Schrijf een PSD uitsluitend met behulp van de volgende vragen:

- mooi t.v. -programma?
- vriend(in) verhinderd?
- zaterdagavond?

(De opdrachten moet je zelf formuleren.)



§ 1.4. Samenvatting

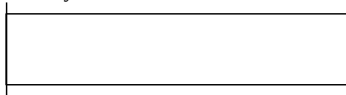
In dit hoofdstuk kwamen o.a. aan de orde:

- 1 de probleemstelling;
- 2 de oplossingsmethode in de vorm van een algoritme;
- 3 de omzetting van een algoritme in een programmastructuurdiagram.

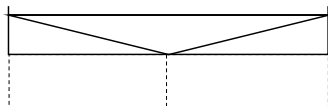
Een programmastructuurdiagram of kortweg PSD geeft ons de mogelijkheid beknopt en toch overzichtelijk aan te geven hoe een bepaald probleem moet worden opgelost. Deze heeft de vorm van een rechthoek, waarin de verschillende symbolen getekend worden.

In dit hoofdstuk hebben we twee symbolen leren gebruiken.

Dit is het symbool voor het laten uitvoeren van een opdracht. Het heeft één *ingang* en één *uitgang*.



Dit is het symbool voor vragenstellingen (testen). Het heeft één ingang en twee uitgangen, namelijk voor JA en NEE. We spreken af dat we de linkerkant gebruiken voor Ja en de rechterkant voor Nee!





2 De computer

§ 2.1. Inleiding

In het vorige hoofdstuk kwamen PSD's en algoritmen aan de orde. Echter om te programmeren moeten de opdrachten die de computer krijgt in een wiskundige vorm ingetoetst worden. In dit hoofdstuk gaan we hier verder op in.

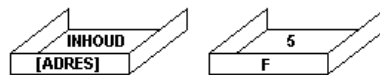
Het geheugen van de computer heeft zeer veel geheugenplaatsen, zodat we over veel variabelen kunnen beschikken. Voor elke variabele is een geheugenplaats beschikbaar. Om de computer de juiste opdrachten te geven is het belangrijk om te weten hoe deze geheugenplaatsen werken.

§ 2.2. Geheugenplaatsen

We kunnen een geheugenplaats vergelijken met een postvak. De geheugenruimte van een computer is dan te vergelijken met een ruimte met heleboel postvakjes.

Zo'n geheugenruimte heeft, net als een postvak een adres en een inhoud.

- ♦ Adres: in hogere programmeertalen kunnen adressen voorgesteld worden door symbolische waarden, bijv: A, B, f, som, totaal1, totaal2. Deze symbolische adressen kunnen bestaan uit letters e/o cijfers. De naam **moet** echter altijd met een letter beginnen. **Spaties** en **andere leestekens** in de naam zijn **niet** toegestaan.
- ♦ Inhoud: de inhoud kan een getal of een rij letters, cijfers of leestekens (bijv. een of meer woorden) zijn.



Geheugenplaats voorgesteld als postvak

Een andere manier om aan te geven wat de inhoud is van de adressen is door een zogenaamde geheugentabel (=schematische weergave van het geheugen van een computer). In het boek Fundament Informatica heet dat een ooggetuigeverslag. Hieronder staat een PSD met zijn bijbehorende geheugentabel.

<i>getal</i> := 7	GEHEUGENTABEL <table><tr><th><i>getal</i></th><th><i>getal1</i></th><th><i>getal2</i></th></tr><tr><td>7</td><td>-3</td><td rowspan="2">6.25</td></tr><tr><td></td><td></td></tr></table>	<i>getal</i>	<i>getal1</i>	<i>getal2</i>	7	-3	6.25		
<i>getal</i>		<i>getal1</i>	<i>getal2</i>						
7		-3	6.25						
<i>getal1</i> := -3									
<i>getal2</i> := 6.25									

Er zijn een aantal belangrijke zaken die in bovenstaand voorbeeld naar voren komen.

In het PSD:

Met een waardetoekenningsoopdracht is het mogelijk een getal in het geheugen te zetten. In het PSD zijn er drie waardetoekenningsoopdrachten gegeven. *Getal* := 7 is de eerste. Het teken: = spreek je uit als 'wordt'. De eerste waardetoekenningsoopdracht van het PSD luidt 'getal wordt 7. Dit betekent voor de computer zet het getal 7 in het geheugen met als symbolisch adres *getal*.

In de geheugen tabel:

De waarden die de adressen aannemen worden niet naast elkaar geschreven. Dit komt omdat in de geheugentabel is aangegeven welk adres als eerste zijn waarde heeft gekregen. *Getal1* heeft zijn waarde eerder gekregen dan *getal2*, daarom is -3 hoger geplaatst als 6.25.

Bekijk de geheugentabel aan de rechterkant. Wat gebeurt er met de waarden van *getal*, *getal1*, *getal2*?

<i>tot</i> := <i>getal</i> + <i>getal2</i>
--

GEHEUGENTABEL		
<i>getal</i>	<i>getal1</i>	<i>getal2</i>
7	-3	6.25
	27.3	-57
36		

De inhoud van twee adressen kan ook bij elkaar worden opgeteld. Dus opdrachten als *tot* := *getal* + *getal1* zijn ook mogelijk. Als deze opdracht wordt uitgevoerd krijgt *tot* als inhoud de waarde $36 + 27.3 = 63.3$. Hiervoor kijken we naar de waarden van *getal* en *getal1* in de tweede geheugentabel.



2.1 opdracht

Gegeven zijn de getallen 1, 2 en 3. Schrijf een PSD waarin:

- achtereenvolgens A, B en C de waarden 1, 2 en 3 krijgen toegekend.
 - A, B en C worden opgeteld en onder de naam SOM in het geheugen wordt opgeborgen.
 - A, B en C worden vermenigvuldigd en onder de naam PROD in het geheugen wordt opgeborgen.
 - $PROD - SOM$ wordt uitgerekend en onder de naam A in het geheugen wordt opgeborgen.
- Geef ook de geheugentabel van dit PSD.

2.2 Opdracht

Vul de geheugentabel hieronder in aan de hand van het volgende PSD.

C1 := 6
C5 := C1 + 3
C1 := C1 + C5
C5 := C5 + C1
P := C1 + C5

GEHEUGENTABEL					

§ 2.3. Teksten en getallen

Gegevens zijn te onderscheiden in constanten en variabelen. Deze kunnen op hun beurt alfanumeriek of numeriek zijn.

§ 2.3.1. Alfanumerieke en numerieke gegevens.

Numerieke gegevens zijn gegevens waarmee je kunt rekenen en bestaan uit cijfers.

Bijv: de getallen 3 en 987.

Alfanumerieke gegevens zijn gegevens waarmee je *niet* kan rekenen en bestaan uit cijfers en/of letters.

Bijv: het telefoonnummer 569-3333 en de naam Janchi.

Alfanumerieke gegevens moeten tussen aanhalingstekens worden geplaatst.

Bijvoorbeeld: "computer" en "640-9999"

§ 2.3.2. Constanten en variabelen.

Een constante is een grootte met een vaste waarde, die tijdens de uitvoering van een programma niet verandert. Men onderscheidt

numerieke constanten (getallen) bijv:

het O.B. percentage: 5

de temperatuur: -6

aantal leerlingen: 28

alfanumerieke constanten (teksten) bijv:

de naam: Buchi Wan

het adres: Santa Rosa 123

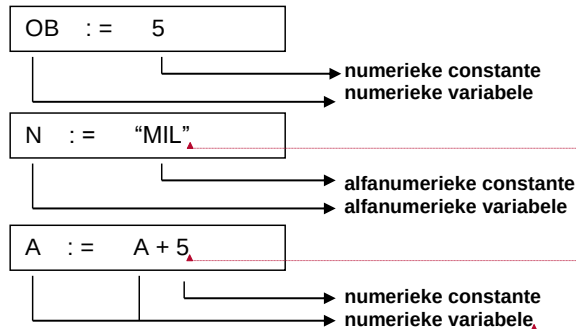
het telefoonnummer: 767-1234

Een variabele is grootte waarvan de waarde tijdens de uitvoer van een programma kan veranderen. Men onderscheidt:

-numerieke variabelen en alfanumerieke variabelen



Onderscheidt tussen constante en variabele.



Formatted

Formatted

Formatted

Let op:

Een numerieke variabele kan alleen een numerieke waarde bevatten.

Een alfanumerieke variabele kan alleen een alfanumerieke waarde bevatten.

§ 2.4. Rekenkundige bewerkingen en prioriteitsregels.

Het rekenorgaan kan onder meer de volgende rekenkundige bewerkingen uitvoeren:

Optellen (+)	som := $G1 + G2$; de uitkomst van een optelsom noemt men de SOM.
Aftrekken (-)	ver := $G4 - 5$; de uitkomst van een aftreksom noemt men het VERSCHIL.
Vermenigvuldigen (*)	prod := $15 * G3$; de uitkomst van een maalsom noemt men het PRODUKT.
Delen (/)	quot := $G7 / G8$; de uitkomst van een deelsom noemt men de QUOTIENT
Machtsverheffen (^)	macht := $G \wedge E$; de uitkomst van een machtsom noemt men de MACHT.

Deze rekenkundige bewerkingen hebben de volgende prioriteit.

prioriteit	Rekenkundige bewerkingen
van hoog	Haakjes () Machtsverheffen (^) Vermenigvuldigen (*) en delen (/) Optellen (+) en aftrekken (-)
naar laag	

Uit dit schema blijkt dat vermenigvuldigen en delen gelijke prioriteit hebben. Dit geldt ook voor optellen en aftrekken. De bewerkingen met gelijke prioriteit worden uitgevoerd in de volgorde waarin ze van links naar rechts voorkomen.

2.3 Opdracht

Geef de waarden van de geheugenadressen A, B, C, en D.

$$A = 9 - 6 / 3 \quad B = 9 - (6/3) \quad C = 5 (18 - 4^2) \quad D = 5 (18 - 4)^2$$

2.4 Opdracht

Geef de waarden van de geheugenadressen E, F, G, H, I en J.

$$E = 3 + 8 - 6 / 2 * 6 \wedge 2 \quad G = 3 + (8 - 6) / 2 * 6 \wedge 2 \quad I = 1 * (2 + 3) * 8 - 6$$

$$F = (3 + 8) + (6 / 3 * 6) \wedge 2 \quad H = 1 * 2 + 3 * 8 - 6 \quad J = 1 * (2 + 3) * (8 - 6)$$



§ 2.5. Uitvoeropdracht / Schrijfopdracht

2.5. Opdracht.

Schrijf een PSD dat het product berekent van de getallen 5, 3 en 7 en de uitkomst onder de naam uitk in het geheugen opbergt.

Het resultaat van de bewerking uit opdracht 2.5 staat in het geheugenadres uitk. Het is echter niet gemakkelijk om in het geheugen van de computer te kijken. Om het resultaat zichtbaar te maken, is de computer uitgerust met een uitvoerorgaan, zoals onder andere een printer en/of een beeldscherm. Met de opdracht **SCHRIJF uitk** geven we de computer opdracht de inhoud van het geheugenadres uitk via het uitvoerorgaan zichtbaar te maken. Dit heet de uitvoeropdracht of schrijfopdracht.

a : = 5
b : = 12
som : = a + b
SCHRIJF som

UITVOER
17

NB: Het uitvoersymbool is geen symbool van een PSD

Om aan te geven wat de uitvoer is wordt het UITVOERSymbool getekend.

2.6. Opdracht.

Schrijf een PSD dat de waarde van *RENTE* berekent met onderstaande formule en daarna afdrukt.

$$RENTE = \frac{KAP * PROC * MND}{1200}, \text{ waarbij } KAP = 750, PROC = 6 \text{ en } MND = 3.$$

§ 2.6. Invoeropdracht / Leesopdracht

In opdracht 2.6. heb je gezien dat je de rente berekent waarbij het kapitaal, het rentepercentage en het aantal maanden vaststaat. Stel je voor dat het rentepercentage om de drie maanden door de centrale bank wordt gewijzigd. Het gevolg is dat je om de drie maanden je programma moet wijzigen om het juiste percentage in te voeren. Om maar te zwijgen om de rente uit te rekenen van verschillende mensen die verschillende kapitalen op de bank hebben.

Om bovenstaande problemen op te lossen is er een andere manier om een getal in het geheugen van de computer te zetten, namelijk de leesopdracht.

Met **LEES g** geven we de computer opdracht om gegevens in te lezen onder andere door middel van het toetsenbord. Nadat het getal is ingetoetst, wordt het in het geheugen gezet onder de naam g.



2.7 Opdracht

Vul geheugentabel en uitvoerblok in voor onderstaande PSD, indien voor G1 het getal 9 en voor G2 het getal 3 ingelezen wordt.

LEES g1
LEES g2
$s := g1 + g2$
$v := g1 - g2$
SCHRIJF s, v

GEHEUGENTABEL			

UITVOER

Let op:

- Achter een LEES-opdracht mogen verschillende namen van geheugenadressen vermeld worden om een aantal gegevens ná elkaar via het toetsenbord in te lezen.
- Achter een SCHRIJF-opdracht mogen eveneens verschillende namen van geheugenadressen vermeld worden. De inhoud van elk geheugenadres wordt in dit geval op dezelfde regel afgedrukt.
- De namen van de verschillende geheugenadressen dienen zowel bij de LEES-opdracht als bij de SCHRIJF-opdracht van elkaar gescheiden te worden door een komma.

2.8 Opdracht

Schrijf een PSD dat van een bepaalde order het bruto-orderbedrag en het netto-orderbedrag inleest, en het verleende kortingspercentage afdrukt.

Ga ervan uit dat de korting berekend wordt over het bruto-orderbedrag.

2.9 Opdracht

Het waterbedrijf AWADUSHI N.V. bepaalt het bedrag dat haar klanten moeten betalen als volgt:

- meterverhuur: een vast tarief van f 35,-
- waterverbruik: f 8,- per m^3 .

Schrijf een PSD dat aan de hand van het ingelezen aantal verbruikte m^3 het te betalen bedrag inclusief de 5% OB afdrukt.

2.10 Opdracht

Schrijf een PSD dat eerst de getallen X en Y inleest, waarna de inhoud van X en Y verwisseld wordt. Maak een geheugentabel ter controle.



§ 2.7. Samenvatting

In dit hoofdstuk kwam o.a. aan orde:

De werking van het interne geheugen uitgelegd aan de hand van PSD's. Door een waardetoekenings-opdracht kan een getal in het geheugen van een computer worden opgeborgen.

De inhoud van het geheugen kan worden bijgehouden door een geheugentabel.

De naam van symbolische adressen mogen alleen uit letters en cijfers bestaan en moeten met een letter beginnen.

Gegevens zijn te onderscheiden in:

-constanten

-variabelen

Constanten zijn te onderscheiden in:

-numerieke constanten (getallen)

-alfanumerieke constanten

Variabelen zijn te onderscheiden in:

-numerieke variabelen

-alfanumerieke variabelen

Het rekenorgaan kan onder meer de volgende rekenkundige bewerkingen uitvoeren:

Optellen (+);

Aftrekken (-)

Vermenigvuldigen (*)

Delen (/)

Machtsverheffen (^)

Deze rekenkundige bewerkingen hebben de volgende prioriteit.

<i>prioriteit</i>	<i>Rekenkundige bewerkingen</i>
van hoog	Haakjes ()
↓	Machtsverheffen (^)
↓	Vermenigvuldigen (*) en delen (/)
naar laag	Optellen (+) en aftrekken (-)

Men kan op de volgende manieren met behulp van de leesopdracht een getal laten inlezen.

LEES g	LEES g2, g4
--------	-------------

Men kan op de volgende manieren met behulp van een schrijfoopdracht een getal laten afdrucken.

SCHRIJF som	SCHRIJF s, v
-------------	--------------

Met een geheugentabel wordt de inhoud van het geheugen bijgehouden.

GEHEUGENTABEL				

Een geheugentabel is geen symbool van een PSD.

Formatted

Met de uitvoersymbool wordt aangegeven wat de waarde is die via het uitvoerapparaat wordt afgedrukt.

UITVOER

Het uitvoersymbool is geen symbool van een PSD.



3. De computer kiest

§ 3.1. Inleiding

In hoofdstuk 2 hebben we alleen opdrachten besproken die de computer één voor één in de aangegeven vaste volgorde moet uitvoeren. Het is in een aantal gevallen wenselijk van deze vaste volgorde af te wijken, waarbij bepaalde opdrachten wél en andere opdrachten niet uitgevoerd moeten worden. Het wel of niet uitvoeren van bepaalde opdrachten wordt in een schema aangegeven in het beslissingssymbool. Zie hoofdstuk 1.

Op grond van de vergelijking ofwel de conditie in dit symbool, waarvan de uitkomst waar (JA) of nietwaar (NEE) is, kiest de computer met welke opdracht verder gegaan moet worden.

§ 3.2. Relatietekens

Een relatieteken is een teken dat in een beslissingssymbool wordt gebruikt om de relatie tussen twee grootheden (conditie) te testen. De volgende relatietekens gebruiken we in een beslissingssymbool :

=	A = Z	A gelijk aan Z
<	Z < 0	Z kleiner dan 0
<=	5 ≤ H	5 kleiner dan of gelijk aan H
>	K > B	K groter dan B
>=	X ≥ 3	X groter dan of gelijk aan 3
<>	M ≠ N	M ongelijk aan N

De uitslag van de test kan uitsluitend waar (JA) of niet waar (NEE) zijn. Bijvoorbeeld:



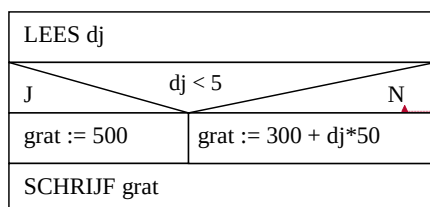
Hieronder volgt een voorbeeld van een opgave.

De personeelsleden van een bedrijf krijgen op het einde van een jaar een gratificatie. Personeelsleden die korter dan 5 jaar bij dit bedrijf in dienst zijn, ontvangen een bedrag van f 500,-. De overige personeelsleden ontvangen f 300,- plus f 50,- voor elk dienstjaar. Geef in een PSD aan, op welke manier de gratificatie van een willekeurig personeelslid wordt vastgesteld en afgedrukt.

Algoritme:

1. Lees het aantal dienstjaren (dj) van dit personeelslid in
2. Is $dj < 5$?
 - Zo ja, geef *grat* waarde 500
 - Zo niet, geef *grat* de waarde $300 + dj * 50$
3. Druk de inhoud van *grat* af.

Het PSD:



Formatted



§ 3.3. Analyse van een probleem

De probleemanalyse bestudeert de wegen die leiden tot de oplossing van het probleem. Dit houdt in dat onderzocht moet worden welke gegevens er ingevoerd moeten worden, welke bijkomende gegevens door middel van berekeningen, formules enz. nog meer nodig zijn, welke mogelijkheden zich kunnen voordoen en waar de moeilijkheden zitten.

Veel beginners maken de fout door meteen het PSD te maken zonder tevoren een helder idee te hebben over de aanpak die leidt tot de oplossing van het probleem of vraagstuk. Het gevolg is dat men na een aantal wijzigingen en probeersels door de bomen het bos niet meer ziet en er niet meer uitkomt.

Oplossingen verlopen volgens het volgende grondpatroon:



Bij het zoeken naar een oplossing is de volgende werkwijze aan te bevelen:

Fase 1: analyse van het probleem

1. Welke informatie wordt verlangd?
2. Welke gegevens zijn nodig om de informatie te kunnen verstrekken?
3. Welke van die gegevens zijn al aanwezig en kunnen ingelezen worden?
4. Welke gegevens moeten dan nog berekend worden?
5. Hoe moeten deze gegevens berekend worden?
6. Hoe moet het resultaat gepresenteerd worden?
7. Welke volgorde moet er gehanteerd worden om het beoogde resultaat te bereiken?

Fase 2: uitwerking en controle van de analyse

Indien alle vragen van fase 1 volledig zijn afgewerkt begint fase 2:

1. Schrijf aan de hand van de bovenstaande analyse het PSD
2. Kies zodanige testgegevens dat alle delen van het PSD getest worden.
3. Controleer met behulp van de zelfgekozen testgegevens of het PSD tot het beoogde resultaat leidt.
4. Indien het beoogde resultaat niet bereikt wordt, begin dan opnieuw met fase 1 van het probleem en ga na van punt 1 t/m 7 waar de denkfout zit.

3.1 Opdracht

Schrijf een PSD dat van twee willekeurig ingelezen getallen het kleinste getal afdruckt.

3.2 Opdracht

Het lesgeld van een muziekschool is afhankelijk van het aantal kinderen uit één gezin dat op de muziekschool zit. Het lesgeld in guldens bedraagt bij:

- 1 kind f 300,-
- 2 en 3 kinderen f 500,-
- meer dan 3 kinderen f 650,-

Vul onderstaande vragen en opdrachten in in het PSD hiernaast, zodat aan de hand van het aantal ingelezen kinderen (AK) het bedrag van het verschuldigde lesgeld (LG) wordt afgedrukt.

$ak < 2$

$ak > 3$

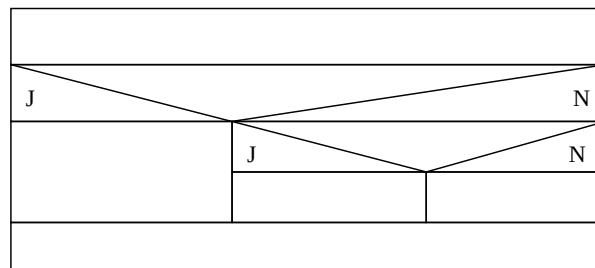
$lg := 300$

$lg := 500$

$lg := 650$

LEES ak

SCHRIJF lg



Mag de volgende opdracht op de volgende manier geschreven worden?

$lg := f$ 300,-



3.3 Opdracht

Men ontvangt alleen bonus van f 500,- als het jaarsalaris f 20000,- of minder bedraagt. Schrijf een PSD dat een jaarsalaris (s) inleest en het bonusbedrag (b) afdruckt.

3.4 Opdracht

De kerstbonus van een bedrijf bedraagt voor elke werknemer 5% van zijn jaarsalaris. De kerstbonus (in guldens) is maximaal f 2500,- en minimaal f 750,-. Schrijf een PSD dat van een willekeurig ingelezen jaarsalaris het van de kerstbonus vaststelt en afdruckt.

3.5 Opdracht

In een videotheek wordt voor het te laat terugbrengen van videocassettes de boete als volgt berekend:
 - f 2,50 indien de videocassette 1 dag te laat wordt teruggebracht;
 - f 3,50 indien de videocassette 2 of 3 dagen te laat wordt teruggebracht;
 - f 5,00 indien de videocassette meer dan 3 dagen te laat wordt teruggebracht.
 Schrijf een PSD dat inleest hoeveel dagen een videocassette te laat is teruggebracht en tevens de boete inclusief 5% OB vaststelt en afdruckt.

3.6 Opdracht

De vertegenwoordigers van een handelsonderneming ontvangen 2% provisie (=komishon) over de door hen behaalde omzet (=benta). De maximale provisie per vertegenwoordiger is echter vastgesteld op f 5000,-. Schrijf een PSD dat de behaalde omzet van een vertegenwoordiger inleest en de provisie afdruckt. Test het schema met zelfgekozen testgegevens.

3.7 Opdracht

In verband met het 100-jarig bestaan van bedrijf 'PORTA BLANKU' is besloten elk personeelslid een jubileumgratificatie te geven. Deze gratificatie bedraagt per personeelslid 5% van zijn jaarsalaris, verhoogd met $\frac{1}{2}\%$ per dienstjaar. De gratificatie bedraagt maximaal 20% van het jaarsalaris. Schrijf een PSD dat van een werknemer het jaarsalaris (sal), en het aantal dienstjaren (jar) inleest, en vervolgens het bedrag voor gratificatie afdruckt. Test het schema met zelfgekozen gegevens.

3.8 Opdracht

De verzekeringsmaatschappij "SEGURO SIGUR" berekent voor zijn verzekeringsagenten de provisie op de volgende manier:
 De provisie bedraagt 10% van de omzet in guldens, vermeerderd met f 1,50 per verkochte eenheid. Indien het aantal verkochte eenheden meer bedraagt dan 1500, ontvangt men 2% extra provisie over de omzet in geld. Schrijf een PSD dat van een verzekeringsagent de omzet in guldens en het aantal verkochte eenheden inleest en de provisie afdruckt. Test het schema met zelfgekozen gegevens.

3.9 Opdracht

In glasfabriek 'MASGLAS' worden 25 soorten glas vervaardigd. Ten behoeve van de afdeling administratie heeft elke glassoort een code. De prijs exclusief OB van elke glassoort wordt op basis van de code als volgt vastgesteld:
 -code 1 t/m 10: f 10,- per m^2 glas + f 15,- snijkosten per bestelling;
 -code 11 t/m 20: f 17,50 per m^2 glas + f 35,- snijkosten per bestelling;
 -code 21 t/m 25: f 36,- per m^2 glas + f 53,- snijkosten per bestelling;
 Schrijf een PSD dat, uitgaande van het ingelezen aantal m^2 glas en de glascode, afdruckt hoeveel men voor een bepaalde bestelling moet betalen. De ob bedraagt 5%. Test het schema met zelfgekozen gegevens.



3.10 Opdracht

Schrijf een PSD dat van een willekeurig ingelezen brutoweekloon het nettoloan berekent en afdruckt, indien voor de inhoudingen het volgende geldt:

1. Als het weekloon minder dan f 150,- bedraagt, wordt niets ingehouden.
2. Als het weekloon niet minder is dan f 150,- en niet meer is dan f 250,-, wordt er 10% ingehouden.
3. Als het weekloon meer is dan f 250,- wordt er 30% ingehouden.

3.11 Opdracht

Een onderneming stelt op de volgende wijze de omzetbonus voor haar klanten vast:

1. Bij een omzet lager dan f 5000,- wordt geen omzetbonus uitgekeerd.
2. Bij een omzet van f 5000,- tot en met f 25000 bedraagt de omzetbonus 1% van de totale omzet van de afnemer.
3. Bij een omzet van meer dan f 25000,- bedraagt de omzetbonus 1½ % van de totale omzet van de afnemer.

Schrijf een PSD dat de omzet van een klant inleest en de omzetbonus afdruckt.

3.12 Opdracht

Een bedrijf berekent op de volgende manier voor zijn werknemers het brutoloan:

-Brutoloan = normaal loon + overwerkloon

-Het normale loon is aantal gewerkte uren * uurloon

-Het overwerkloon is aantal uren boven 38 * het overwerkuurloon.

-Het overwerkuurloon is 150% van het uurloon.

-Een werknemer kan minder dan 38 uur gewerkt hebben.

Schrijf een PSD dat van een werknemer het aantal gewerkte uren en het uurloon inleest en het brutoloan afdruckt.

Test het schema met zelfgekozen testgegevens.

3.13 Opdracht

De hoogte van de contributie van een sportvereniging is afhankelijk van de sport die men beoefent, de leeftijd en het aantal jaren dat men lid is van de vereniging. De gegevens voor de berekening van de contributie zijn:

-contributie voor sportcode 1 bedraagt f 150,- en voor sportcode 2 f 225,-;

-leden ouder dan 40 jaar krijgen f 25,- korting;

-leden die langer dan 10 jaar lid zijn, krijgen f 20,- korting.

Schrijf een PSD dat van een lid de sportcode (sp), de leeftijd (l) en het aantal jaren lidmaatschap (j) inleest en het bedrag van verschuldigde contributie afdruckt.

Test het schema met zelfgekozen gegevens.

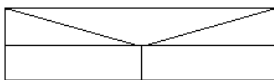


§ 3.4 Samenvatting

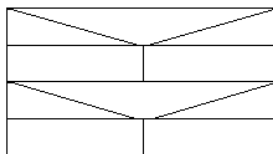
De relatietekens in het beslissingssymbool zijn:

teken	voorbeeld	conditie
=	$A = Z$	A gelijk aan Z
<	$Z < 0$	Z kleiner dan 0
<=	$5 \leq H$	5 kleiner dan of gelijk aan H
>	$K > B$	K groter dan B
>=	$X \geq 3$	X groter dan of gelijk aan 3
<>	$M \neq N$	M ongelijk aan N

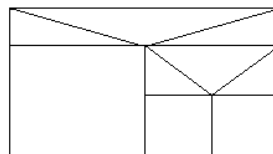
Het beslissingssymbool werd in dit hoofdstuk onder andere op de volgende manieren gebruikt:



één beslissingssymbool

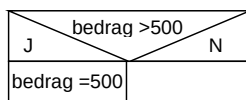


meerdere beslissingssymbolen
met verschillende variabelen.



meerdere beslissingssymbolen
bij verschillende voorwaarden
voor dezelfde variabele.

Het programmeerdeel dat bepaalt dat een bedrag maximaal 500 is heeft de volgende structuur:





4 De computer op herhaling

§ 4.1. Inleiding

In de voorafgaande hoofdstukken werd elke opdracht in het PSD slechts één keer uitgevoerd. Het komt echter vaak voor dat delen van een PSD een aantal keren worden herhaald. Op deze manier is het mogelijk met weinig instructies veel werk te laten doen. In het PSD dient wel duidelijk het einde van de herhaling aangegeven te worden.

Om een reeks opdrachten een aantal keren achter elkaar te laten uitvoeren maakt men gebruik van een cyclus of lus. In het engels wordt dit een loop genoemd.

§ 4.2. Probleemstelling

We bekijken de probleemstelling 'hoe je de 10 cijfers van het decimale stelsel laat afdrukken'.

Je kunt dit probleem met onderstaand PSD aanpakken.

$p := 0$
SCHRIJF p
$p := p + 1$
SCHRIJF p
$p := p + 1$
SCHRIJF p
$p := p + 1$
SCHRIJF p
$p := p + 1$
SCHRIJF p
$p := p + 1$
SCHRIJF p
$p := p + 1$
SCHRIJF p
$p := p + 1$
SCHRIJF p
$p := p + 1$
SCHRIJF p

De uitvoer van het PSD wordt dan:

UITVOER
0
1
2
3
4
5
6
7
8
9

Het PSD hiernaast kan veel korter. Als je het goed bekijkt zie je dat de volgende twee opdrachten een aantal malen worden herhaald.

SCHRIJF p
$p := p + 1$

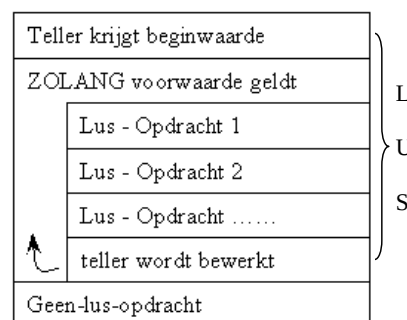
Opdat dezelfde uitvoer wordt verkregen als in het voorbeeld hierboven, moeten de twee opdrachten negen keer uitgevoerd worden. Het komt er dan als volgt uit te zien:

$p := 0$
ZOLANG $p \leq 9$
SCHRIJF p
$p := p + 1$

De herhaallus die hiernaast staat afgebeeld is een tellende lus hierna wordt verder ingegaan op tellende lussen.

§ 4.3. Tellende lussen

Men gebruikt een tellende lus als men van te voren het aantal herhaling weet. Een tellende lus heeft de volgende structuur.



Wat gebeurt er precies in de lus?

- 1) de teller krijgt een beginwaarde buiten de lus
- 2) Met de zolang opdracht wordt gekeken of er aan de gegeven voorwaarde wordt voldaan. De teller zit altijd in de voorwaarde.
- 3) Als aan de voorwaarde wordt voldaan worden de gegeven opdrachten uitgevoerd.
- 4) De teller wordt bewerkt en daarna wordt weer gekeken of aan de voorwaarde wordt voldaan.
- 5) Als er niet aan de voorwaarde wordt voldaan wordt verder gegaan met de opdrachten buiten de lus.

**4.1 Opdracht**

Maak de geheugentabel van het PSD hiernaast en vul het in, indien voor G de waarde 3 ingelezen wordt.
Geef ook aan wat de uitvoer wordt.

LEES G
T := 1
ZOLANG T <= 8
U := T * G
SCHRIJF U
T := T + 1

4.2 Opdracht

Schrijf een PSD waarin van de 25 in te lezen getallen de tweede macht wordt afgedrukt.

4.3 Opdracht

Maak een geheugentabel van het PSD aan de rechterkant en vul het in.
Geef ook de uitvoer.
Wat wordt er in het PSD berekend?
Waarom staat in het voorwaarde symbool geheugenadres T en niet geheugenadres SOM?

SOM := 0
T := 11
ZOLANG T <= 20
SOM := SOM + T
T := T + 1
SCHRIJF SOM

Opmerking:

Aan het begin van het PSD wordt aan geheugenadres SOM de waarde 0 toegekend. We mogen namelijk niet zonder meer aannemen dat de inhoud van geheugenadres SOM, rechts van het teken := bij waardetoekenningsoopdracht $SOM := SOM + T$, bij de eerste verwerking nul is.
Men dient aan elk geheugenadres dat achter := genoemd wordt, vooraf een waarde toe te kennen.

4.4 Opdracht

Schrijf een PSD waarin de som van de getallen 100 t/m 200 berekend en afgedrukt wordt.

4.5 Opdracht

Schrijf een PSD dat eerst één getal inleest en vervolgens van dit ingelezen getal de machten 0 t/m 16 afdrukt.

4.6 Opdracht

Schrijf een PSD dat de som bepaalt van getallen 100 t/m 250 en vervolgens de som en het gemiddelde van die getallen afdrukt.

4.7 Opdracht

Schrijf een PSD dat de eerste 10 uitkomsten van de tafel van 6 afdrukt.

4.8 Opdrachten

Maak de geheugentabel van het PSD hiernaast en vul het in, indien voor G de waarde 2 ingelezen wordt.
Geef ook aan wat de uitvoer wordt.

SOM := 0
LEES G
T := 75
ZOLANG T <= 100
U := T * G
SOM := SOM + U
T := T + 5
SCHRIJF SOM

4.9 Opdracht

Schrijf een PSD (zonder leesopdracht) dat de som van de volgende reeks van tien getallen bepaalt en afdrukt.

10 20 30 40 50 60 70 80 90 100

4.10 Opdracht

Schrijf een PSD (zonder leesopdracht) dat de som bepaalt en afdrukt van de volgende reeks getallen.

50 45 40 35 30 25 20 15 10 5 0 -5

**4.11 Opdracht**

Wat is de uitvoer van onderstaand PSD?

A := 7
B := 3
ZOLANG B <= A
SCHRIJF B
B := B + 1

4.12 Opdracht

Wat is de uitvoer van het onderstaande PSD?

S := 4
E := 16
B := 8
ZOLANG B <= E
SCHRIJF B
B := B + S

4.13 Opdracht

Maak een geheugentabel van onderstaand PSD en vul het in. Geef ook aan wat de uitvoer is.

SOM := 0
N := 5
TEL := 1
ZOLANG T <= N
U := TEL * TEL
J U > 10 N
SOM := SOM + U
TEL := TEL + 1
SCHRIJF SOM

4.14 Opdracht

Schrijf een PSD dat van twintig ingelezen getallen (ongelijk aan nul) bepaalt en afdruckt hoeveel van die getallen positief dan wel negatief zijn.

4.15 Opdracht

Een docent wenst met behulp van de computer een overzicht met het totaal aantal behaalde voldoende en het totaal aantal behaalde onvoldoendes voor een proefwerk.

Schrijf een PSD dat:

- het totaal aantal leerlingen inleest dat aan het proefwerk heeft deelgenomen;
- van elke leerling het behaalde cijfer inleest;
- de gewenste gegevens afdruckt.

Opmerking:

De proefwerk cijfers zijn afgerond op 1 decimaal.

4.16 Opdracht

Maak een geheugentabel van onderstaand PSD en vul het in. Geef ook aan wat de uitvoer is.

T1 := 2
ZOLANG T1 <= 3
T2 := 8
ZOLANG T2 <= 20
U := T1 * T2
T2 := T2 + 4
T1 := T1 + 1

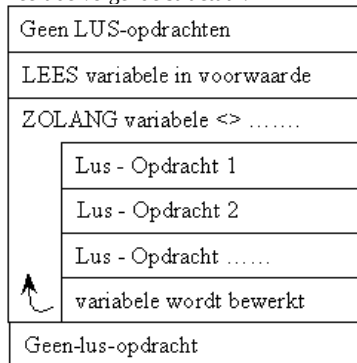
4.17 Opdracht

Schrijf een PSD dat de uitkomsten van de tafels 11 tot en met 20 onder elkaar afdruckt.



§4.4. De niet tellende lus.

Men gebruikt een niet-tellende lus als men het aantal herhaling van tevoren niet weet. Een niet-tellende lus heeft de volgende structuur.



Wat gebeurt er precies in de lus?

1. de variabele krijgt een beginwaarde buiten de lus
2. Met de zolang opdracht wordt gekeken of er aan de gegeven voorwaarde wordt voldaan. De variabele zit altijd in de voorwaarde. Als aan de voorwaarde wordt voldaan krijgt de variabele een onlogische waarde. Dit noemt men het sluitgegeven. Hiermee stopt de lus.
3. Als aan de voorwaarde wordt voldaan worden de gegeven opdrachten uitgevoerd.
4. De variabele wordt bewerkt en daarna wordt weer gekeken of aan de voorwaarde wordt voldaan.
5. Als er niet aan de voorwaarde wordt voldaan wordt verder gegaan met de opdrachten buiten de lus.

4.18 Opdracht

Schrijf een PSD dat van een groep leerlingen:

- de proefwerkcijfers inleest;
- berekent hoeveel leerlingen een cijfer lager dan 6 behaald hebben;
- het aantal leerlingen afdruckt met een cijfer lager dan 6.

Neem als sluitgegeven het getal 99.

4.19 Opdracht

Schrijf een PSD dat van een aantal ingevoerde positieve en negatieve getallen de som afdruckt van:

- de positieve getallen;
- de negatieve getallen.

Neem als sluitgegeven het getal 0.

4.20 Opdracht

1. Vul van het onderstaande PSD de geheugentabel in.

G := 1
ZOLANG G < 50
SCHRIJF G
G := G * 2

2. Welke drie symbolen vormen de basiselementen van deze niet-tellende lus?

Opmerking: De waarde waarmee in het beslissingssymbool het einde van de niet-tellende lus wordt aangegeven kan zijn:

- een sluitgetal of sluitgegeven;
- het resultaat van een berekening.

4.21 Opdracht

Schrijf een PSD dat een ingelezen getal alleen vermenigvuldigt met 3, de nieuwe waarde hiervan weer met 3 enz., zolang de waarde kleiner is dan 500. Tenslotte dienen de laatste waarde van het getal en aantal herhalingen afgedrukt te worden.



4.22 Opdracht

Schrijf een PSD dat een aantal getallen inleest en alleen van de getallen die groter zijn dan 100, de som bepaalt en afdruckt. Het sluitgetal is 0.

4.23 Opdracht

Schrijf een PSD dat een willekeurig aantal getallen inleest en hiervan het kleinste getal afdruckt. Neem als sluitgegeven het getal 0.

Aanwijzing: Neem hierbij voorlopig aan dat het eerste getal dat gelezen wordt het kleinste is en zet dit in geheugenadres *kleinste*.

4.24 Opdracht

Een onderneming voert een assortiment van 200 artikelen. Zij wenst een overzicht van alle artikelnummers (inclusief de voorraad in stuks) waarvan de voorraad minder dan 50 stuks bedraagt.

Schrijf een PSD dat:

- het artikelnummer en de voorraad in stuks van elk artikel inleest;
- de gewenste gegevens afdruckt.

4.25 Opdracht

Bij loononderhandelingen is de vakbond met een bedrijf overeengekomen dat alle werknemers met een maandsalaris lager dan f 2500,- en die langer dan drie jaar in dienst zijn, in aanmerking komen voor een éénmalige uitkering van f 450,-.

Schrijf een PSD dat:

- van elke werknemer het werknemersnummer, het bedrag van het maandsalaris en het aantal jaren dienstverband inleest;
- van de werknemers die in aanmerking komen voor de eenmalige uitkering, het werknemersnummer afdruckt.

4.26 Opdracht

Schrijf een PSD dat van honderd ingelezen getallen bepaalt en afdruckt hoeveel getallen er tussen 30 en 45 ingelezen worden.

4.27 Opdracht

Schrijf een PSD dat van elke verkoopfactuur:

1. het factuurnummer, het brutofactuurbedrag en het nettofactuurbedrag inleest;
2. bepaalt hoeveel korting van brutofactuurbedrag verleend is;
3. van een factuur waarvan het verleende kortingspercentage meer dan 15% bedraagt, het factuurnummer en kortingspercentage afdruckt.

Het sluitgegeven is factuurnummer 9999.

4.28 Opdracht

Schrijf een PSD dat van een bedrijf:

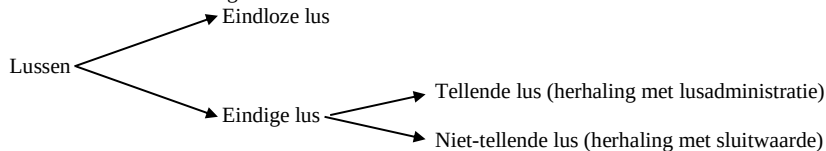
- van elke werknemer van het werknemersnummer, de leeftijd en het aantal dienstjaren inleest;
 - het gemiddeld aantal dienstjaren berekent en afdruckt van alle werknemers die ouder zijn dan 40 jaar.
- Het sluitgegeven is werknemersnummer 9999.



§4.5. Samenvatting

Een herhaling in een PSD noemen we een cyclus, lus of loop.

De lussen worden als volgt onderscheiden:



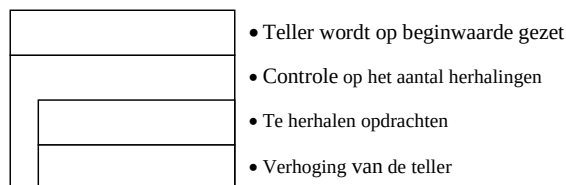
De eindloze lus is het gevolg van een logische fout. Hierbij is het einde:

- niet aangegeven
- onjuist aangegeven

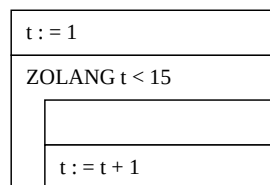
De eindige lussen kunnen weer onderverdeeld worden in:

- tellende lus, waarbij het aantal herhalingen vooraf bekend is;
- de niet-tellende lus, waarbij het aantal herhalingen vooraf niet bekend is.

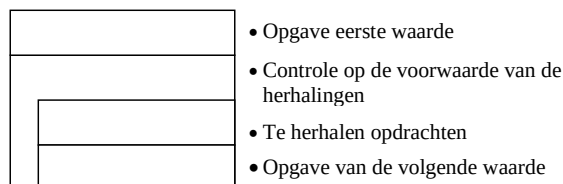
De basisvorm van een tellende lus:



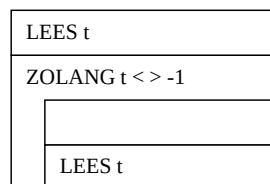
voorbeeld



De basisvorm van een niet-tellende lus:



voorbeeld



Sluitgetal of sluitgegeven:

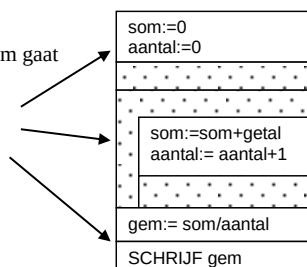
Een getal of gegeven dat gebruikt wordt om het einde van de gegevensinvoer / lus aan te geven. Dit getal of gegeven mag logischerwijze geen normaal voorkomend te verwerken gegeven zijn

De waarde waarmee in het beslissingssymbool het einde van de niet-tellende lus wordt aangegeven, kan zijn:

1. sluitgetal of sluitgegeven;
2. resultaat van een berekening

Het berekenen van een som, aantal, gemiddelde, maximum of minimum gaat in 3 stappen:

1. Voor de lus wordt de variabele geïnitieerd met een beginwaarde;
2. in de lus wordt de variabele bijgewerkt en
3. na de lus wordt de conclusie berekend (gemiddelde) en afgedrukt





5 Modulair programmeren

§ 5.1. Inleiding

Modulair programmeren is een programmeermethode waarbij het op te lossen probleem wordt onderverdeeld in een aantal logische deelproblemen (=submodulen), die elk duidelijk een eigen functie hebben. Een submoduul kan op zijn beurt ook onderverdeeld worden, tot het punt bereikt wordt waarop men de details wenst uit te werken.

De modulaire opbouw van een programma is globaal te vergelijken met een studieboek dat vanwege de overzichtelijkheid onderverdeeld is in een aantal hoofdstukken met paragrafen. De hoofdstukindeling is echter niet willekeurig gekozen, maar is gebaseerd op de samenhang van de min of meer afgeronde onderwerpen van de totale leerstof.

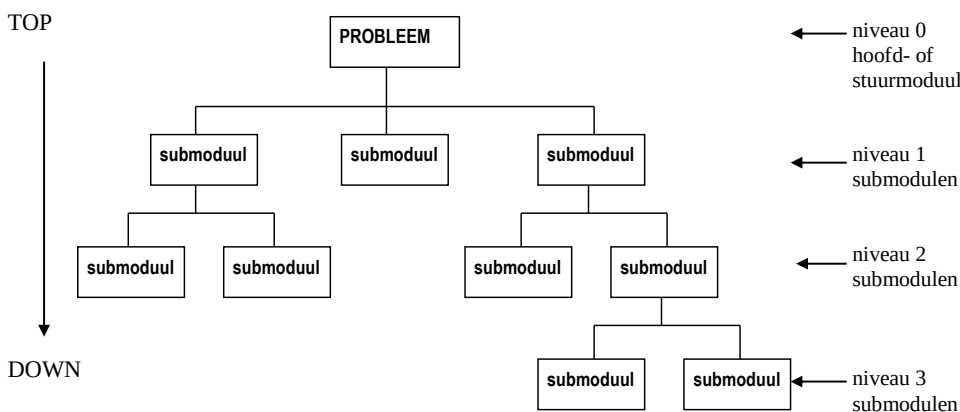
§ 5.2. Voordelen modulaire programma opbouw

Hier zijn een aantal voordelen van de modulaire programma opbouw:

- ❑ de leesbaarheid van het programma wordt beter, waardoor het gemakkelijker is eventuele fouten op te sporen te herstellen dan wel op een later tijdstip wijzigingen aan te brengen.
- ❑ Er ontstaat tijdswinst, doordat enkele programmeurs tegelijk aan hetzelfde programma kunnen werken
- ❑ Het oplossen van een complex probleem wordt eenvoudiger indien men het in deelproblemen splitst. Men kan zich nu concentreren op een onderdeel van het totale probleem, waarbij vermeden wordt dat men door de bomen het bos niet meer ziet.
- ❑ Wanneer een bepaalde reeks instructies op verschillende plaatsen in een programma terugkomt, is het niet nodig dat deze reeks telkens opnieuw geschreven wordt. Op een eenvoudige manier is het mogelijk steeds van hetzelfde submoduul gebruik te maken.

§ 5.3. Top-down benaderingswijze

Over het algemeen is een goed gestructureerd programma modulair opgebouwd. Om tot modulaire opbouw te komen, kan men gebruik maken van het boomdiagram ofwel top-down benaderingswijze. Hierbij wordt het op te lossen probleem eerst gesplitst in een aantal grote, herkenbare submodulen, ieder met een duidelijke specifieke functie. Vervolgens wordt elk submoduul op dezelfde wijze in kleinere submodulen gesplitst. De oplossing wordt vanuit een aantal niveaus opgebouwd, waarbij elk niveau een verdere uitwerking is van het voorgaande. Schematisch kan het bovenstaande worden weergegeven in de vorm van een boomdiagram.





§ 5.4. Modulaire opbouw van een PSD

Bij de modulaire opbouw bestaat het PSD uit de volgende onderdelen of modules:

1. een hoofddiagram (hoofdmodule of stuurmodule), dat in een overzichtelijke vorm de oplossing van het deelprobleem in grote lijnen weergeeft onder verwijzing naar diverse detaildiagrammen (submodulen).
2. een of meer submodulen, waarin men de deelproblemen apart uitwerkt en die op hun beurt naar een of meer detaildiagrammen kunnen verwijzen, die ook weer nader uitgewerkt dienen te worden.

VOORBEELD

Probleemstelling

Een onderneming stelt aan het begin van elk jaar voor haar klanten de bonusuitkering (=extra korting) vast over het voorafgaande jaar. De bonus bedraagt voor elke klant 2% van de omzet in geld vermeerderd met f 5,- voor elk afgenomen artikel. Indien voor een klant de totale bonus minder dan f 500,- bedraagt, wordt aan die klant geen bonus uitgekeerd.

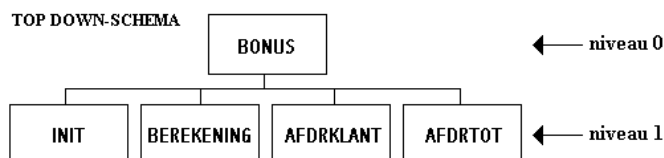
De verkoopafdeling wenst een overzicht met:

1. van elke klant die voor een bonusuitkering in aanmerking komt:
 - het klantnummer;
 - de uit te keren bonus;
2. de totalen van:
 - de bonusbedragen;
 - het aantal klanten dat een bonusuitkering ontvangt.

Oplossing:

We kunnen dit probleem in vier deelproblemen splitsen:

1. Submodule INITIALISEREN:
In deze module wordt een begin- of startwaarde toegekend (=initialiseren) aan de variabelen die de totale bonusuitkering (TOTB) en het aantal klanten dat een bonusuitkering ontvangt (TOTK), bijgehouden.
2. Submodule BEREKENING:
In deze module worden de overige benodigde gegevens ingelezen, te weten van elke klant de omzet in geld (OMG) en de afname in stuks (AFS). Tevens wordt de bonusuitkering (BON) berekend.
3. Submodule AFDRKLANT:
In deze module worden de gegevens afgedrukt van elke klant die voor een bonusuitkering in aanmerking komt. Deze gegevens zijn het klantnummer (KNR) en de bonusuitkering (BON). Tevens wordt in dit module de totalen bijgehouden.
4. Submodule AFDRTOT:
In deze module worden de gewenste totalen afgedrukt.



A. Stuurmodule (niveau 0)

INITIALISEREN
LEES KNR
ZOLANG KNR <> 9999
BEREKENING
AFDRKLANT
LEES KNR
AFDRTOT

B. Vier submodulen (niveau 1)

Submodule INIT

TOTB := 0
TOTK := 0

Submodule BEREKENING

LEES OMG, AFS
BON := 2 * OMG / 100
BON := BON + 5 * AFS

Submodule AFDRKLANT

J	BON >= 500	N
SCHRIJF KNR, BON		
TOTB := TOTB + BON		
TOTK := TOTK + 1		

Submodule AFDRTOT

SCHRIJF TOTB, TOTK



OPMERKINGEN:

1. De stuurmoduul begint met:

Stuurmoduul NAAM

Bijv.

Stuurmoduul BONUS

2. Een verwijzing of aanroep van een submoduul wordt als volgt in het diagram aangegeven:

NAAM SUBMODUUL

Bijv.

BEREKENING

Het aanroepen van een submoduul wil zeggen, dat op die plaats in het diagram de opdrachten van de aangeroepen submoduul moeten worden doorlopen. Er wordt als het ware naar het begin van het aangeroepen submoduul gesprongen.

Na het doorlopen van het submoduul wordt verder gegaan met de eerstvolgende opdracht ná de submoduul-aanroep

3. Het submoduul heeft dezelfde opbouw als het stuurmoduul. Het submoduul begint echter als volgt:

Submoduul NAAM

Bijv.

Submoduul BEREKENING

Het splitsen van programma's in kleine onafhankelijke delen wordt verder uitgewerkt in het stukje over VISUAL BASIC.



§ 5.5. Samenvatting

Bij de modulaire opbouw van een programma bestaat het PSD uit:

1. een hoofd- of stuurmoduul.
2. een of meer submodulen.

Het op te lossen probleem kan schematisch in de vorm van een boomdiagram (top-down benaderingswijze) weergegeven worden.

Stuurmoduul NAAM

Begin stuurmoduul

NAAM SUBMODUUL

Submoduul -aanroep

Submoduul NAAM

Begin submoduul



6 Werken met Visual Basic.

§ 6.1. Inleiding.

Visual Basic of VB is een programma waarmee u een Graphic User Interface-programma kan creëren. Een Graphic User Interface-programma of GUI-programma is een programma waarbij men door middel van klikken met de muis gegevens kan invoeren.

Met Visual Basic 6.0 kunt u programma's maken die gedraaid kunnen worden op Windows 95, 98, 2000 en Windows NT.

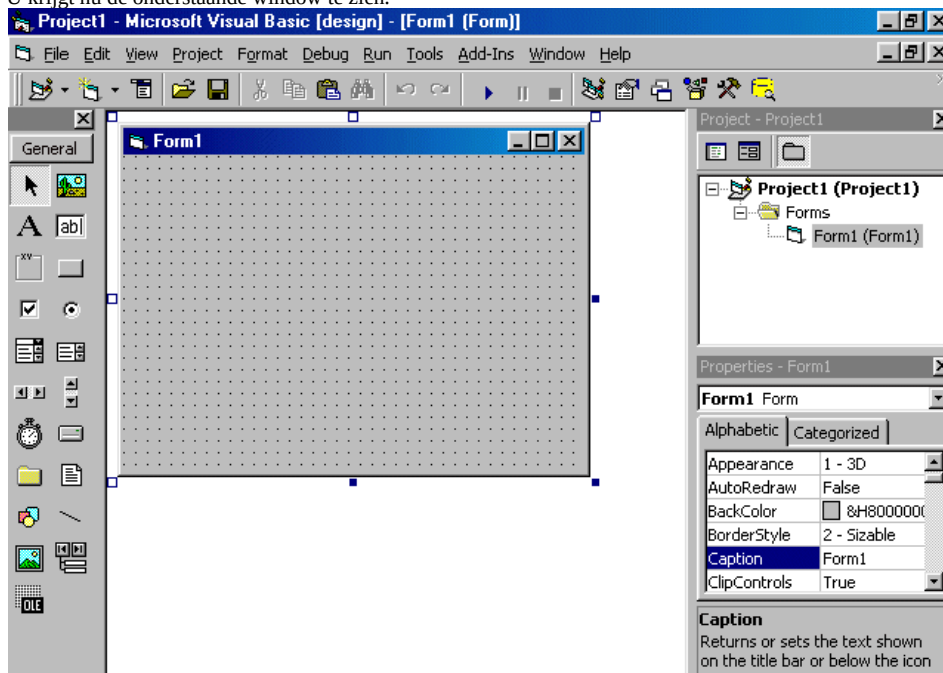
Op de website van Microsoft <http://www.microsoft.com/vb> is meer informatie over Visual Basic te verkrijgen, of op de informaticasite <http://www.informaticavo.nl>

§ 6.2. Visual Basic verkennen.

Om Visual Basic te starten moet u de volgende stappen ondernemen.

1. Klik op **Start**, vervolgens op **Programs** en dan uiteindelijk op **Visual Basic** (of *Visual Studio*).
2. Klik op **Microsoft Visual Basic** om het programma te starten.
3. Nu wordt gevraagd welk type programma u wilt creëren. U selecteert **Standard EXE** en klikt vervolgens op **OPEN**.

U krijgt nu de onderstaande window te zien.

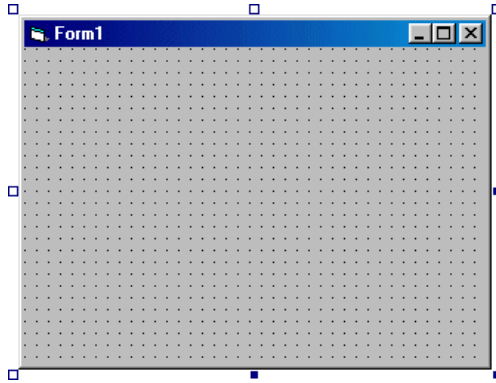


Dit is de **IDE** (= **I**ntegrated **D**eveloping **E**nvironment), een werkomgeving die alle "gereedschappen" bevat om een nieuw programma te ontwerpen, schrijven, maken en te testen. In het begin lijkt het een beetje verwarrend, maar naderhand zult u aan deze werkomgeving wennen.

Het grootste gedeelte van de window bestaat uit het **Form View** gedeelte. Dit is het belangrijkste gedeelte, want hierin vindt het ontwerpen van het programma plaats. Ook de codes die worden uitgevoerd na bepaalde programma-acties worden in dit gedeelte geschreven.



Een **Form of formulier**, in VB terminologie, is gewoonweg een window of dialoog box. In de openingswindow heeft VB al een blanco form (Form1) gecreëerd. U kunt nu een begin maken met het programma ontwerp.



controls

Op de formulieren moet u **controls (besturingselementen)** zetten. Controls zijn onderdelen van het aanzicht van het programma. Voorbeelden van controls zijn onder andere: gewone tekst, textboxes, afbeeldingen, knoppen en schuifbalken (=scroll bars). Aan de linkerzijde van het Form View gebied is er een lijst te zien van alle beschikbare controls (voorgesteld als icons). De lijst met controls wordt de **Toolbox** genoemd. Zie de afbeelding rechtsonder.

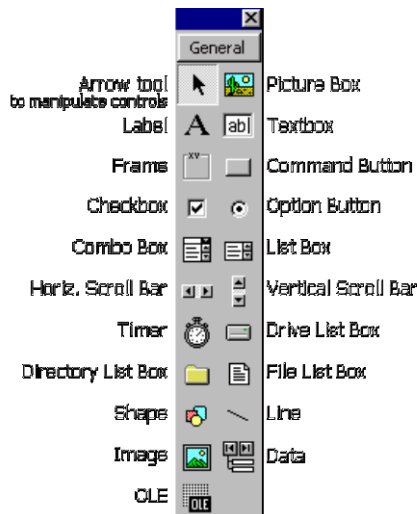
De afbeelding hiernaast laat de Toolbox zien met de algemene controls. Wat elke control voorstelt komt later aan de orde. In het navolgende ziet u hoe een control op het formulier gezet kan worden.

Volgt de aangegeven stappen:

1. Selecteer de Text Box control van de toolbox.
2. Beweeg de cursor over het formulier: het krijgt de gedaante van een kruis.
3. Hou de linkermuisknop ingedrukt, sleep de cursor om een rechthoek te creëren. Laat vervolgens de muisknop los.

U hebt een nieuw Text Box gecreëerd met de naam Text1.

Om de grootte van een control te veranderen kan men het selecteren en daarna naar de gewenste grootte slepen via de size handles (= 9 blauw gekleurde vierkantjes rond de control). Om de positie van control te veranderen kan men het selecteren en naar de gewenste plaats slepen. Als men een control wilt verwijderen moet men de control eerst selecteren en daarna op DEL-toets drukken op het toetsenbord.



Om het formulier waarop men aan het werken is te selecteren, moet men op een leeg gedeelte op het formulier zelf klikken



In deze fase kunt u niet in de textbox schrijven, zoals dat bij andere windows programma's gebruikelijk is. Dit komt omdat u in de design mode zit. (kijk naar de VB title bar). In de design mode of tijdens design time ontwerpt u alleen uw applicatie (=programma).

Om de applicatie te laten draaien, kunt u het start knopje (▶) selecteren van de toolbar bovenaan of op F5 drukken op uw toetsenbord.



In de run mode of runtime kunt u zien hoe applicatie draait als het helemaal af is. Om uit de run mode te gaan kunt u het formulier gewoon sluiten met de end knop(■). Hierdoor komt u weer in de design mode Dit was een eerste kennismaking van Visual Basic. U kunt ook nu de andere controls exploreren.

De eigenschappen van controls.

U weet nu wat controls zijn. De controls kunt u natuurlijk manipuleren.

Bijvoorbeeld: Hoe kunt u tekst op een knop veranderen? Met de control properties (=control eigenschappen). In VB is een property kenmerkend voor een control. Het zorgt voor de kleur, de tekst, het lettertype van een control.

Wanneer u in design mode zit kunt u de properties van een control veranderen door gebruik te maken van de properties window rechtsonder in uw scherm. Zie de afbeelding hiernaast.

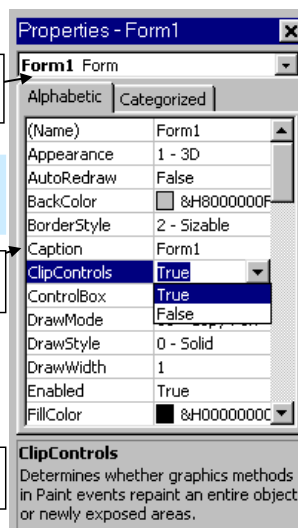
Om een property van een control te veranderen, moet u eerst de property selecteren met de muis. De eigenschap wordt daarnaast ingetikt of u kunt in de dropdown lijst in de properties window de eigenschap kiezen van de -op dat moment geselecteerde- control.

Naam van de control: in dit geval Form 1

Onthoudt dat een formulier ook een control is

Property Caption: Welke tekst op de control komt.

Hier staat uitleg over de geselecteerde control



In de alfabetische lijst ziet u alle properties voor de geselecteerde control. Bijvoorbeeld de tekst in de titelbalk van een formulier kunt u veranderen in de Caption property. Een property kan bij verschillende controls voorkomen. Bijvoorbeeld de caption property geeft bij de control knop, de tekst aan die op de knop verschijnt.

De meest algemene en belangrijke property is **Name**, die later uitgebreid aan bod zal komen.

Om de waarde van een bepaalde property te wijzigen moet u de volgende handelingen verrichten:

1. Klik op de property. In de linkerkolom staan de properties. In de figuur hierboven is de property Caption geselecteerd, het heeft daarom een blauwe kleur. Aan de onderkant, in de figuur aangegeven met A, wordt een beschrijving gegeven van de property.
2. Tik of selecteer een waarde voor de property in de kolom ernaast.

Probeer nu een aantal controls te maken en verander hierbij hun properties.

- Maak een form met height 1000 en width 1000 en start het programma.
 - Beeindig het programma en verander Height en Width dan in 3000, start het programma weer en bekijk de grootte.
 - Geef als Caption; Dit is mijn eerste formulier.
 - Kies verschillende StartUpPositions
- Zet een textbox op het formulier,
 - met een lichtgroene BackColor.
 - kies voor Enabled eerst true en daarna false en probeer na te gaan wat het verschil is.
 - kies verschillende BorderStyles
- Doe hetzelfde met een label.



7 Events en Codes

§ 7.1. Inleiding

In het hoofdstuk 6 was het accent gelegd in de creatie van een raamwerk voor een applicatie. Daarbij werd aandacht besteed aan het aanmaken van controls en hun properties. In dit hoofdstuk leert hoe u uw programma kan manipuleren. Met andere woorden u “vertelt” uw programma hoe te reageren op bepaalde events (=gebeurtenissen), bijvoorbeeld het klikken op een knop.

Als u een applicatie (=gebruikersprogramma) maakt, bent u het grootste gedeelte van de tijd bezig om de codes te schrijven. Dat zijn de commando's in een bepaalde volgorde, die het programma vertellen wat het moet doen. (bijvoorbeeld: berekeningen). De code moet in de Visual Basic language (=VB programmeertaal) geschreven worden.

VB wordt een event-driven language genoemd. Dit betekent dat de code wordt gesplitst in verschillende delen die onafhankelijk worden uitgevoerd afhankelijk van acties van de gebruikers.. De acties die het uitvoeren van de codeopdrachten tot gevolg heeft noemt men events.

§ 7.2. Code toevoegen aan control events

Elke control heeft zijn eigen events. Bijvoorbeeld command buttons hebben de Click event (de click event wordt gestart telkens wanneer de gebruiker op het knopje klikt.). Formulieren hebben de Load event (de Load event wordt gestart telkens wanneer het formulier opstart). Door code toe te voegen aan de events kun je het programma laten doen wat jij wilt.

Hoe kunt u code toevoegen aan een event

1. dubbelklik op een control om toegang te krijgen tot zijn events.

2. het Form View gebied of het scherm wordt vervangen door de Code View. In het Code View gebied kunt u de code tikken. Er zijn twee dropdown lijsten bovenaan de Code View. Zie de figuur hieronder.

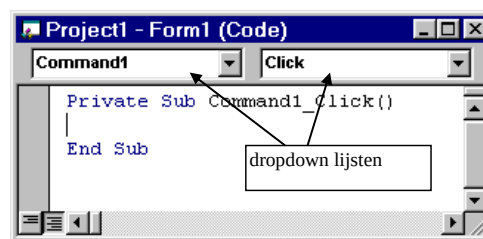
De eerste geeft aan welke control u heeft geselecteerd. (u kunt met deze lijsten schakelen naar een andere control, zonder eerst naar de Form View te gaan). De tweede lijst geeft de op dat moment geselecteerde event. (u kunt met de lijst naar andere events gaan; controls hebben verschillende events).

3. In het grootste gedeelte van de Code View, kunt u de codes intikken. VB geeft altijd de volgende twee code regels aan

```
Private Sub controlname_eventname()  
End Sub
```

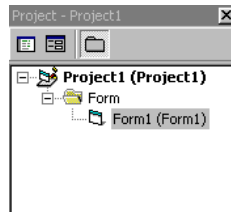
Al de codes voor de aangegeven event moet tussen die twee regels geschreven worden.

De eerste regel vertelt aan VB, dat de code gerelateerd is aan een bepaalde event van een control, daar start. En de laatste regel vertelt aan VB dat de code daar eindigt. Dus u moet zorgen dat de cursor tussen de twee regels staat als u begint met typen.





Om terug te gaan naar **Form View**, moet je dubbelklikken op de form's naam in de **Project Explorer** aan de rechterkant van het scherm



Laten we een stukje code schrijven. Open VB en creëer een command button op het formulier. Geef nu de volgende code op de button's Click event:

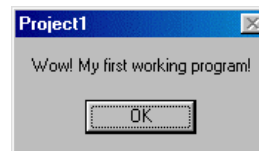
MsgBox "Wow! My first working program!"

MsgBox is een klein commando in VB dat een klein informatie rechthoek opent. De tekst die volgt op het commando die tussen aanhalingstekens is, is de tekst die te zien is in de info rechthoek.

Je event code moet dan als volgt uitzien:

```
Private Sub Command1_Click()  
    MsgBox "Wow! My first working program!"  
End Sub
```

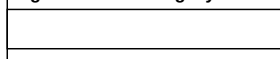
Als je het programma nu runt (Start knopje of F5) en vervolgens op het command knopje klikt, dan wordt het Click event gestart en de gerelateerde code uitgevoerd. Het resultaat hiervan is de MessageBox die aan de rechterkant te zien is.



§ 7.3. PSD's omzetten in Visual Basic code

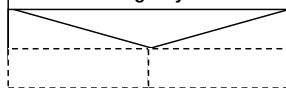
Programma Structuurdiagrammen of kortweg genoemd P.S.D's bevatten drie belangrijke symbolen. Men onderscheidt hierbij het *algemeen handelingssymbool*, het *beslissingsymbool* en de *herhaallus*. Zie de figuur hieronder.

algemeen handelingssymbool

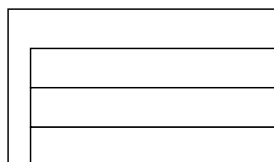


Door middel van een aantal voorbeelden wordt geprobeerd duidelijk te maken hoe een PSD in Visual Basic code wordt omgezet. Alle opdrachten in een *handelingssymbool* worden vrijwel op dezelfde manier overgenomen in Visual Basic code. Hieronder volgt een aantal voorbeelden:

beslissingsymbool



herhaallus



In deze voorbeelden worden de opdrachten in Visual Basic code heel simpel. Hierbij wordt de code altijd weergegeven tussen de woorden Private Sub ... en ..End Sub. In de daadwerkelijke codering kunnen deze twee woorden anders zijn. Dit wordt automatisch door Visual Basic aangegeven.

PSD

voorbeeld 1

```
a := 56
```

voorbeeld 2

```
b := c * 25
```

VISUAL BASIC

```
Private Sub  
a=56 Let wel de dubbele punt is in Visual Basic verdwenen.  
End Sub
```

```
Private Sub  
b=c*25  
End Sub
```



voorbeeld 3

inkooppr : = 100
perc : = 0.32
verkoop : = inkooppr * perc
SCHRIJF verkoop

```
Private Sub
inkooppr = 100
perc = 0.32
verkoop = inkooppr * perc
MsgBox verkoop
End Sub
```

Drie belangrijke opmerkingen:

- ❑ **de SCHRIJF-opdracht** wordt hier in Visual-Basic een **MsgBox-opdracht** (andere mogelijkheden voor uitvoer zijn **de textbox** of voor SCHRIJF-opdrachten in de lus **de Listbox**)
- ❑ code voor een PSD moet altijd tussen twee regels geschreven worden, die het begin en het einde van de event aangeven. Visual Basic zet deze twee regels automatisch. Hier is dat Private Sub om het begin aan te geven en End Sub om het einde aan te geven.
- ❑ **elke opdracht** schrijven we op een **nieuwe regel**.

voorbeeld 4

LEES a
SCHRIJF "antwoord=" a * 7

```
Private Sub Command4_Click()
a = InputBox("Geef een cijfer")
MsgBox ("antwoord=") & a * 7
End Sub
```



In de code van voorbeeld 4 heb je een aantal nieuwe dingen gezien.

- **De LEES-opdracht** wordt d.m.v. een InputBox aangegeven. De variabele a krijgt de waarde die je in de InputBox intikt.
- **gebruikersvriendelijkheid:** Om de bedoeling van de inputbox of msgbox aan te geven, wordt een tekst geplaatst tussen de aanhalingstekens staat aangegeven. Deze tekst komt dan leetrlijk zo in de boxen te staan. Zie de figuren hiernaast.
- **Meerdere dingen afdrukken.** Bij de MsgBox zie je dat wat tussen haakjes staat letterlijk wordt overgenomen en na &-teken komt de waarde van een variabele of van een berekening



7.1 Opdracht

Schrijf de Visual Basic code om een MsgBox te krijgen zoals hiernaast, met de tekst 'Het getal 9 is geweldig'. Hierbij moet het getal 9 door de optelsom 4 +5 worden verkregen. Geef deze opdracht in één regel op.



Het *beslissingssymbool* wordt op een andere manier omgezet dan het *handelings-symbool*. Aan de hand van een aantal voorbeelden zullen we dat duidelijk maken.

voorbeeld 5

J	a > 7	N
b : = a + 5		

```
Private Sub
IF a > 7 THEN
b = a + 5
END IF
End Sub
```

voorbeeld 6

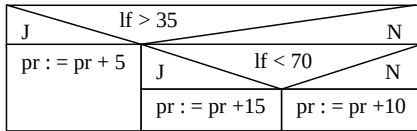
J	pr < 35	N
pr : = pr + 5	pr : = pr +10	

```
Private Sub
IF pr < 35 THEN
pr = pr + 5
ELSE
pr = pr + 10
END IF
End Sub
```

Belangrijk: het *beslissingssymbool* wordt in Visual Basic code omgezet in een **IF ... THEN** - statement. Deze statement sluiten wij **altijd** af met **END IF**.



voorbeeld 7



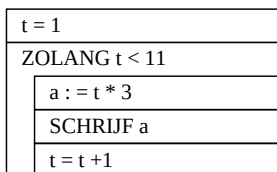
```

Private Sub
IF If > 35 THEN
pr = pr + 5
ELSE
IF If < 70 THEN
pr = pr + 15
ELSE
pr = pr + 10
ENDIF
END IF
End Sub

```

Dan volgt als laatste de herhaallus. Aan de hand van een voorbeeld zullen we ook de herhaallus bespreken. Voorbeeld 8 wordt als volgt in Visual Basic code omgezet:

voorbeeld 8



```

Private Sub
t=1
WHILE t < 11
a = t * 3
MsgBox a
t = t + 1
WEND
End Sub

```

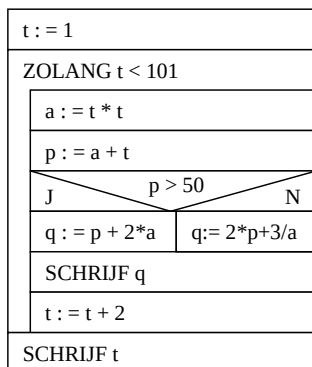
Belangrijk: de **herhaallus** wordt in Visual Basic code omgezet in een **WHILE ...WEND** - statement. Deze statement begint met **WHILE** en wordt altijd afgesloten met **WEND**.

Declaratie van variabelen

Hoewel het niet verplicht is is het een goede gewoonte om alle variabelen waarmee een programma werkt te declareren (= naam en het soort variabele aangeven) om fouten te voorkomen en geheugenverspilling tegen te gaan. Dit doe je met de instructie: **DIM**. (Bijv DIM teller AS Integer of DIM naam As String, etc).

Enkele belangrijke soorten zijn: Integer (hele getallen) Double (decimale getallen)
String (tekst) Date (datum en tijd)
Boolean (logische waarden deze kunnen zijn *true* of *false*)

voorbeeld 9



```

Private Sub
Dim t As Integer
Dim a As Integer
Dim p As Integer
Dim q As Double
t=1
WHILE t < 101
a = t * t
p = a + t
IF p > 50 THEN
q = p + 2*a
ELSE
q = 2*p + 3/a
ENDIF
MsgBox q
t = t + 2
WEND
MsgBox t
End Sub

```

} declaraties



§ 7.4. Benaming van Controls



Label

Afkorting: lbl

De eenvoudigste control: het wordt gebruikt om tekst op het formulier te laten zien. Met de Caption property kan men de tekst aangeven die men wil laten zien. Ook BackColor, ForeColor and Font properties zijn belangrijk in het uiterlijk van de label.



Textbox

Afkorting: txt

Wordt veel gebruikt om de gebruiker informatie te laten intypen. Met de Text property kan men de ingetypte tekst bewerken. Belangrijk zijn ook de Font property en de MaxLength property (om de lengte van de ingetypte tekst aan te geven).



Checkbox

Afkorting: chk

Wordt gebruikt om settings aan/uit te zetten in een programma. Belangrijkste properties zijn Caption en Value: De laatste geeft de waarde / status van de checkbox, 0 (= niet aangevinkt), 1 (= aangevinkt).



Option Button

Afkorting: opt

Met twee of meer van deze controls kan de gebruiker kiezen tussen verschillende opties. De Value property geeft de status weer: True (aangevinkt), False (niet aangevinkt).



List Box

Afkorting: lst

Deze control bevat een lijst van tekst items. Om een nieuw element toe te voegen gebruikt men de AddItem methode als volgt:
`listbox.AddItem "Item Text"`

Elk nieuw item krijgt een toenemende *index*, beginnend bij 0. De ListIndex property geeft de index van de geselecteerde item. Om een item weg te halen, gebruikt men de RemoveItem methode, aangevuld met de index nummer:

`listbox.RemoveItem index`

Om een listbox helemaal leeg te maken, gebruikt men de Clear method, die geen parameters bevat.

Om de waarde van de geselecteerde item te krijgen, gebruikt men de Text property. Om het nummer van de item te krijgen, gebruikt men de ListCount property.

Andere belangrijke properties zijn Sorted (bij activatie, sorteert het de items alfabetisch) en Style (Als het de waarde 1 krijgt, wordt een checkbox naast elk item gezet).



Combo Box

Afkorting: cmb

Bijna hetzelfde als een listbox, het enige verschil is de verschijning: het hangt af van de Style property, of de combo box als een dropdown lijst (=a listkader die tevoorschijn komt als het geklikt wordt, Style = 2) lijkt, een simpele combo (een normale listkader met een tekst gebied dat gebruikt kan worden om elementen op te zoeken, Style = 1), een dropdown combo (een dropdown lijst waarin men kan typen als een combo box, Style = 0).



Command Button

Afkorting: cmd

De meest gebruikte Windows control. Wordt gebruikt om elk type handling te activeren: De enige property die belangrijk is, is Caption. De code die wordt uitgevoerd wordt gezet bij de Click event.



Frame

Afkorting: fra

Een container control de mogelijkheid biedt om controls groeperen. Het wordt niet allen gebruikt om een formulier mooi te maken: bijvoorbeeld als je twee sets of option buttons op een formulier maakt, moet elke set zijn eigen frame bevatten. De belangrijkste property is, alweer, Caption.



Image

Afkorting: img

Geweldig! Versier de formulieren met Beelden. Beelden worden geladen met de Picture property. Om een bestand tijdens runtime te laden, kan men gebruik maken van de LoadPicture command:

`image.Picture LoadPicture("file path")`



Picture Box



Afkorting: pct

Dit is niet een gekopieerde control: terwijl images alleen gefixeerde beelden laat zien (om je formulieren te versieren), picture boxes zijn geavanceerder en bevatten dynamic graphics. Het verschil tussen een Image en een Picture Box is, hetzelfde als tussen een Label en een Textbox.



Horizontal and Vertical Scroll Bar

Afkorting: hsb, vsb



Deze controls worden gebruikt om de elementen text en lists op het formulier te schuiven. Veel controls (zoals textboxes, listboxes, comboboxes) hebben ingebouwde schuifbalken, zodat men ze niet hoeft toe te voegen. Het gebruik is simpel: zet de Min en Max properties met minimum en max waarden; zet de code in de Change event (wordt gestart als de schuifbalken worden gebruikt: gebruik de Value property om de positie van de *drag-box* te bepalen).



Timer

Afkorting: tmr

Een belangrijke control: met deze control kan men code laten uitvoeren in bepaalde tijdsintervallen (bijv. elke seconde) Om het te gebruiken vult men de Interval property met een bepaalde tijd in milliseconden (1 s = 1000 ms) vervolgens geeft men de code aan die moet worden uitgevoerd aan de enige event die mogelijk is, genaamd Timer. Deze control is *onzichtbaar* tijdens runtime.



Drive List Box

Afkorting: drv

Een kleine combo box die al gevuld is met de letters en namen voor alle systeem drives (floppy, hard disk, CD-ROM, ...). De geselecteerde drive letter is in de Drive property vastgelegd. De belangrijkste event is Change.



Directory List Box

Afkorting: dir

Deze list box laat alle folders in de vorm van een boomstructuur zien. Zo kan de gebruiker een bepaald folder of subfolder kiezen. Het uiteindelijk gekozen pad is te vinden in de Path property van de control. Elke keer als de geselecteerde folder veranderd wordt de Change event opgestart.



File List Box

Afkorting: fil

De derde file-gerelateerde control-groep: het laat een listbox zien met alle files die in een specifiek pad zijn opgeslagen. Het pad waarnaar gezocht moet worden vindt men in de Path property, de naam van de geselecteerde file in de FileName property. Deze control wordt vaak samen gebruikt met de DirList control om de juiste folder te selecteren. Deze control heeft ook alle properties/methoden die typisch zijn voor een ListBox (ListIndex, ListCount, ...).



Line and Shape

Afkorting: lin, shp



Met de eerste control kan men een lijn tekenen. Met de tweede kan men een cirkel, vierkant, rechthoek, ovaal, etc tekenen. Om te bepalen wat men met de Shape control kan tekenen, verandert men de Shape property. Beide controls kunnen gekleurd worden met de ForeColor, FillColor and FillStyle properties. Deze controls gebruiken geen methoden en geen events.



Data

Afkorting: dat

Als men met databases wil werken zal men deze control veel gebruiken. Het creëert een "link" tussen een database tabel en andere controls (textboxes, listboxes, ...). Men moet eerst het pad van de DB file in de DatabaseName property zetten om vervolgens de tabelnaam van de RecordSource property te selecteren. Vervolgens kan men andere controls toevoegen (bijvoorbeeld textboxes) en zet men hun DataSource property aan de data control naam en hun DataField property aan het recordsetveld om lezen/schrijven. Tijdens runtime, gebruikt men data control om de records te scrollen. De geassocieerde controls worden gebruikt om velden te lezen / schrijven.



OLE container

Afkorting: ole

OLE (Object Linking & Embedding), is standaard in Windows. Het laat een applicatie objecten lenen van andere applicaties. Geleende objecten kunnen zijn sound files, Word files, Equation Editor objecten en andere.

De bovenstaande afkortingen worden door Visual Basic programmeurs gebruikt.

De *afkorting* wordt in kleine letters aangegeven en het *gekozen gedeelte begint met een hoofdletter*.

Bijv: optWaar, cmdBereken, frmInvoer, txtNaam, etc.



optWaar naam
 opt gedeelte dat het type control aangeeft
 Waar gedeelte dat de eigenbedachte naam aangeeft

- Je moet een control een naam geven die bij zijn functie past Bijvoorbeeld als je een Command Button aanmaakt die een applicatie beëindigt kun je de Command Button cmdEinde noemen.
- Geef een control eerst een naam voordat je de codes ervan intypt!!

§ 7.5. Properties en codes.

Wanneer een control een naam heeft kun je hem manipuleren met code. Een van de simpele acties die uitgevoerd kan worden is de properties (=eigenschappen) met code veranderen tijdens runtime. Om een control's property met code te veranderen gebruikt men de volgende syntaxis (=zinsbouw):

Controlname.propertyname = newvalue.

Oefening / voorbeeld: Maak een Command Button met de naam cmdPrimi en caption "Kua skol?". Voeg de volgende code aan de Click event:

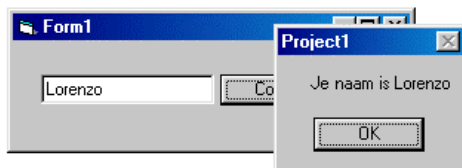
`CmdPrimi.Caption = "MIL"`

Als het programma wordt gerund, zie je dat de Caption van de Command button veranderd.

Het commando `Controlname.propertyname = newvalue` kan in andere VB instructies gebruikt worden zoals de `MsgBox`.

Bijvoorbeeld als je formulier heb met een textbox met de naam `txtNaam` en je hebt ook een Command Button op het formulier, kun dit commando bij de Click event van de Command Button plaatsen:

`MsgBox ("Je naam is ") & txtNaam.Text.` Als je in de textbox Lorenzo invult en vervolgens op de Command



button drukt, dan krijg je het resultaat hieronder.

In de volgende paragraaf worden methoden onder de loep genomen.

§ 7.6. Methoden

Methoden zijn net als properties, elementen die tot een object horen. Het verschil is dat properties het uiterlijk van de control en methoden de acties van de control bepalen. Om een methode te gebruiken wordt de volgende syntaxis gebruikt:

Controlname.methodname [parameters]

De parameters zijn niet altijd verplicht, het varieert per methode type.

Wanneer men de syntaxis van een commando leest, zal men vaak elementen zien die tussen *rechte haakjes* zijn geplaatst ([]): dat is een manier om niet verplichte parameters aan te geven. De verplichte delen zijn *schuingedrukt* (=italics.)

Voorbeelden van code die een methode bevatten:

<code>pctPhoto.Refresh</code>	'ververst een picture box
<code>frmHoofd.Show</code>	'laat het formulier frmHoofd zien
<code>lstWerknemer.AddItem " Nanzi"</code>	'een element toevoegen aan een list box

In Visual Basic kan men commentaar geven tussen de code. Deze commentaren zijn heel nuttig om de code te begrijpen. VB zal alle tekst die volgt na een apostrof zien als een commentaar ('). Deze tekst zal altijd in groen worden weergegeven.



Laten we als voorbeeld een simpel programma maken die een methode gebruikt. Zet een listbox, een textbox en een command button op een formulier. Geef hen respectievelijk de volgnede namen: lstNamen, txtNieuweNaam, cmdToevoegen. Geef de command Button de Caption “Voeg Naam”.

Wanneer op de command button wordt gedrukt willen we dat het volgende gebeurt:

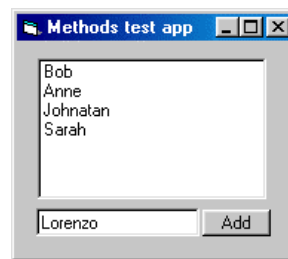
- ❑ De inhoud van de textbox wordt toegevoegd als een nieuw element aan de listbox
- ❑ De textbox wordt vervolgens leeggemaakt.

Om dit te bereiken kun je de volgende code intikken bij de Click event van de command button:

```
lstNamen.AddItem txtNieuwNaam.Text
txtNieuwNaam.Text = ""
```

Het eerste commando zorgt ervoor dat de ingetypte naam als een nieuw element in de listbox terechtkomt. Hierbij dient gezegd te worden dat AddItem de methode is en txtNieuwNaam.text de parameter is. Het tweede commando zorgt ervoor dat de Text property van de textbox geen leeg wordt gemaakt. Dit wordt aangegeven met "".

Het formulier ziet ongeveer er uit als het formulier hiernaast.



7.2 opdracht

Schrijf een programma om de tafel van 6 te produceren zoals hiernaast, als de commandbutton wordt ingedrukt.



Samengevat kunnen we dus zeggen:

- **Forms** of formulieren vormen de gebruikersinterface van het programma. Je zet de controls waarmee je werkt daarin. Verder bevat het ook de codes van de objecten die het bevat. Zowel zichtbare als onzichtbare codes.
- **Controls** of besturingselementen zijn de objecten die je op het formulier zet voor bepaalde taken. Controls kennen events (gebeurtenissen), properties (eigenschappen) en methods (methoden)
- **Events** zijn reacties van een control op een actie. Wanneer je bijvoorbeeld op een command button (opdrachtknop) drukt, dan roept deze knop de event Click op.
- **Properties** zijn de eigenschappen van een object. Deze bepalen het uiterlijk, positie, de naam, etc van het object.
- **Methods** bepalen de taken die een object kan vervullen. Een method is een soort programmafunctie die in een object ingebouwd is. Een form beschikt bijvoorbeeld over de method PrintForm. De instructie Form1.PrintForm drukt een exact duplicaat af van Form1 via de printer. De programmeur hoeft verder niet tot in detail te programmeren hoe dit moet gebeuren.