

CURSUS COBOL

2^E JAAR

DEEL 1

CURSUS COBOL - ONTWERPTECHNIEKEN

1 Inleiding - p 3 - 9

1.1 Standaard-COBOL *RMCOBOL*

Prof. COBOL – Microfocus COBOL
Visual Realia - Computer Associates
Net Express 3.1. – Microfocus

1.2 *Gechiedenis van COBOL*

1959 → Amerikaanse “Department of Defense” gaf opdracht tot ontwikkelen van een programmeertaal met volgende eigenschappen:

- gebaseerd op het Engels
- gemakkelijk te lezen / schrijven
- gemakkelijk in onderhoud
- geschikt voor administratieve problemen
- in principe bruikbaar op elke computer

1960 → COBOL-60 (= *Common Business Oriented Language*)
(was niet volledig computer onafhankelijk)

1961 → COBOL-61

CODASYL (= *Conference on Data System Language*)

COBOL-61 Extended

- tabellen (verzameling van gegevens van hetzelfde type)
- adresseerbaar toegankelijke bestanden

Volgnummer	Naam	Adres	Gehuwd	Loon
1				
2				
...				
498				
499				
500				

Men kan dit personeelsbestand op verschillende manieren organiseren:

SEQUENTIEEL: lijn per lijn

ADRESSEERBAAR: records krijgen een nummer

1968 → (*Deze versie werd een standaard*)

ASA + CODASYL

ASA = American Standard Association

Nu: ANSI (= *American National Standard Institute*)

1970 → Tot internationale standaard verheven door de ISO (*International Organization for Standardization*)

1974 → COBOL-74 o.a. Communicatie tussen hoofdprogramma en subprogramma

1985 → COBOL-85 o.a. faciliteiten ten behoeve van het gestructureerd programmeren (vb scope-terminators), beslissingstabellen

20.. → COBOL-..Object-georiënteerd programmeren

1.3 ISO-COBOL-85

wordt beschreven in 11 module's

1. **NUCLEUS**: voor de interne verwerking van de gegevens
2. **SEQUENTIAL I-O**: voor de verwerking van sequentieel georganiseerde bestanden
3. **RELATIVE I-O**: voor de verwerking van adresseerbaar toegankelijke bestanden waarin de records worden geïdentificeerd door hun relatieve volgnummer
4. **INDEXED I-O**: voor de verwerking van adresseerbaar toegankelijke bestanden waarin de records worden geïdentificeerd door een sleutel, en toegankelijk zijn via een index
5. **INTER-PROGRAM COMMUNICATION**: voor de communicatie tussen hoofdprogramma en subprogramma's
6. **SORT & MERGE**: sorteren en/of samenvoegen van bestanden
7. **SOURCE TEXT MANIPULATION**: kopiëren van programmatekst uit een bibliotheek en voor het modifieren van een programma
8. **REPORT WRITER**: voor het min of meer automatisch samenstellen van overzichten
9. **COMMUNICATION**: voor datacommunicatiefaciliteiten
10. **DEBUG**: voor het uittesten van het programma
11. **SEGMENTATION**: voor het uitvoeren van een programma in een overlapstructuur

1.4 Structuur van een COBOL-programma

1.4.1 Inleidend programmavoorbeeld

```
IDENTIFICATION DIVISION.  
PROGRAM-ID. MIJN-EERSTE-COBOL-PROGRAMMA.  
AUTHOR. IKKE.  
DATE-WRITTEN. 28-SEPTEMBER-2000.
```

```
ENVIRONMENT DIVISION.  
CONFIGURATION SECTION.  
SOURCE-COMPUTER. PC.  
OBJECT-COMPUTER. PC.
```

```
DATA DIVISION.  
WORKING-STORAGE SECTION.  
01 SCHERM-1.  
    03 LIJN-11 PIC X(7) VALUE "DIT IS ".  
    03 LIJN-12 PIC X(11) VALUE "MIJN EERSTE".  
    03 LIJN-13 PIC X(16) VALUE " COBOL-PROGRAMMA".  
    03 LIJN-14 PIC X(21) VALUE "DRUK OP EEN TOETS...".  
01 TOETS PIC X.
```

```
PROCEDURE DIVISION.
```

HOOFDPROGRAMMA.

```
DISPLAY SPACES UPON CRT
DISPLAY LIJN-11 AT 0837
DISPLAY LIJN-12 AT 1035
DISPLAY LIJN-13 AT 1233
DISPLAY LIJN-14 AT 2005
ACCEPT TOETS AT 2030
STOP RUN.
```

Vroeger waren hoofdletters verplicht (bij oude code)
Let op de punten (leestekens)
Lijn-11 betekent: lijn 1 van scherm1

1.5 Schrijfvoorschriften

1.5.1 Indeling en gebruik van programmaregels – p 30

Een COBOL-programma bestaat uit programmaregels. De indeling van een programmaregel is aan voorschriften gebonden.

Positie 1 → 6 : volgnummergebied (uit de tijd van de ponskaarten)

Positie 7 : indicatiegebied

- spatie: normale programmaregel
- */: commentaarregel – p 41
- -: vervolgregel
- D: testregel

Positie 8 → 11 : gebied A } hierin wordt het programma gecodeerd

Positie 12 → 72: gebied B }

Sommige taalelementen moeten in gebied A beginnen:

- hoofdingen van een divisie, sectie, paragraaf
- niveaunummers 01 en 77
- woord DECLARATIVES
- woord END (van END DECLARATIVES en END PROGRAM)
- Bestandsaanduiding FD, SD, CD, RD

De meeste andere taalelementen moeten in gebied B beginnen.

Suggesties voor het schrijven van programma's p 47

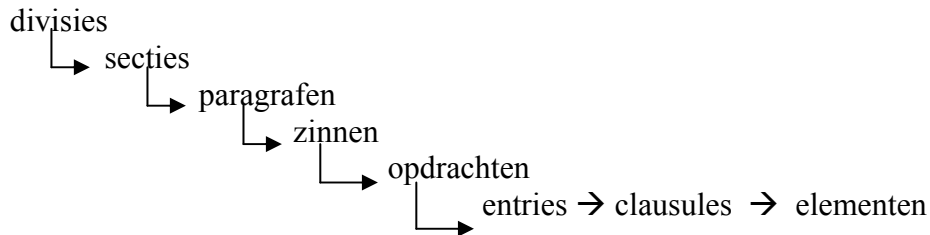
Een prototype voor een COBOL-programma p 48

Criteria p 51 – 62

Vereenvoudig, schrap wat onnodig is, geef uitleg, waarom geen GO TO, inspringen van code, geef informatie in code, schrijf programma's zo dat ze gemakkelijk te veranderen zijn,

www.wiley.com/compbooks/cobol

1.5.2 Algemene opbouw van een COBOL-programma



Opmerkingen:

- de namen van de divisies zijn vast
- de namen van de secties en paragrafen liggen eveneens vast, behalve in de procedure division (daar kiest men zelf de namen)

1.5.3 De 4 divisies p 46

◆ IDENTIFICATION DIVISION.

Bevat commentaar over het programma

PROGRAM-ID *naam van het programma*

AUTHOR.. *naam auteur die het programma heeft geschreven*

DATE-WRITTEN.. *datum waarop het programma geschreven is*

◆ ENVIRONMENT DIVISION.

Er kunnen 2 secties voorkomen

CONFIGURATION SECTION. *(voor hard- en softwarekenmerken)*

Source-computer. *(apparatuur waarmee het programma gecompileerd wordt)*

Object-computer. *(apparatuur waarmee het programma wordt uitgevoerd)*

INPUT-OUTPUT SECTION.

→ dient om de verbinding vast te leggen tussen programma en randapparatuur

◆ DATA DIVISION.

Beschrijft de te gebruiken bestanden en andere gegevens die in het programma gebruikt zullen worden

Er kunnen 5 secties voorkomen zoals

WORKING-STORAGE SECTION.

Hierin worden de gegevens beschreven die alleen een functie hebben binnen het programma

FILE SECTION.

Hierin worden bestanden beschreven

◆ PROCEDURE DIVISION. – naamgeving van procedures (p 40)

Hierin worden de opdrachten statements (p 35) gecodeerd die moeten worden uitgevoerd

De structuur van een cobolprogramma is zodanig opgevat, dat categorieën van veranderingen die zich vaak voordoen slechts een lokale invloed hebben op het programma.

Wil men bevoorbeeld hetzelfde programma met dezelfde gegevens op een ander computertype laten draaien, dan moet men in principe slechts de ENVIRONMENT DIVISION veranderen.

Wil men hetzelfde programma met andere gegevens op hetzelfde computertype laten draaien, dan moet men in principe slechts de DATA DIVISION veranderen.

Veranderen noch de gegevens, noch het computertype, maar wijzigen enkel de opdrachten die op die gegevens moeten toegepast worden, dan hoeft men slechts de PROCEDURE DIVISION te wijzigen.

1.6 Taalelementen

1.6.1 Tekenset van Cobol p 39

A → Z

a → z

0 → 9

spatie

+ - * / = \$, ; . " () > < :

In COBOL mogen hoofdletters en kleine letters door elkaar worden gebruikt. Ze hebben dezelfde betekenis.

Vb. scherm-1 SCHERM-1

1.6.2 Gereserveerde woorden p 42 – appendix B

Het zijn woorden die voor een compiler een vaste omschreven betekenis hebben en meestal slechts in een bepaald deel van het programma mogen voorkomen.

Vb. IDENTIFICATION, SECTION, DISPLAY, ...

Deze woorden zijn gereserveerd, d.w.z. dat men ze niet voor iets anders mag gebruiken. Het moet geschreven worden zoals is voorgeschreven (lijst STANDAARD-COBOL)

Vele compilers hanteren extra gereserveerde woorden. Vb. line, position, at ...

1.6.3 Niet-gereerveerde woorden of zelfgekozen naam of cobol-woord p 39

Ze mogen door de programmeur vrij gekozen worden mits het inacht nemen van de volgende regels:

- het woord mag max. 30 tekens lang zijn
- het mag enkel bestaan uit tekens A → Z a → z 0 → 9 en "-" (minteken)
- het mag niet identiek zijn met een gereserveerd woord
- het moet tenminste 1 alfabetisch teken bevatten ~~007~~ → agent-007
- het mag niet beginnen, noch eindigen met een streepje

1.6.4 Gegevens

Gegevens kunnen worden aangeduid op 1 van de 3 volgend wijzen:

- ◆ Namen (identifiers) p 41
- ◆ Constanten (literals)
- ◆ figuratieve constanten p 43

Namen duiden geheugenzones aan waar zich de betreffende gegevens bevinden.

Een literal is het gegeven zelf.

Vb. 03 lijn-11 PIC X(7) VALUE "Dit is ".

D	i	t		i	s	
---	---	---	--	---	---	--

1.6.5 Literals (constanten)

1.6.5.1 Numerische constanten: rekenkundige waarden

- ◆ bestaat uit 1 of meer cijfers (max. 18 cijfers)
- ◆ eventueel vooraf gegaan door een + of - teken
- ◆ mag een decimale komma, aangeduid door een punt, bevatten maar ze mag niet als laatste teken staan

Vb. 16 +16 -16 ~~+1234567890.1234567890~~ 3.14

1.6.5.2 Niet-numerische literals of alfanumerische constanten

- ◆ bestaat uit 1 of meer tekens (max. 120 tekens) van de tekenset van de betrokken computer, ook de spatie
- ◆ wordt geplaatst tussen aanhalingstekens (‘ ___ ’ of “ ___ ” is afhankelijk van de versie)

Vb. “Dit is” “-702” “Mijn naam is “”Bart””” ”|+|+|+|+|+|”

1.6.5.3 Figuratieve constanten

Dit is een gereserveerd woord, dat gegevens rechtstreeks aanduidt.

Vb. 03 lijn-11 PIC X(7) VALUE “ “.
-of- 03 lijn-11 PIC X(7) VALUE SPACES.

Space – Spaces – Zero – Zeros – Zeroes – Low-Value – Low-Values – High-Value – High-Values – Quote – Quotes – All

Niet gebruiken bij alfanumerische constanten, wel bij display-instructie.

Vb. DISPLAY quote “einde” quote

1.6.5.4 Booleaanse constanten

TRUE - FALSE in EVALUATE-opdracht

1.6.6 Speciale Tekens

1.6.6.1 Leestekens p 32 - 33

- ◆ De spatie: de spatie is nodig om taalelementen van elkaar te scheiden.
Vb. scherm-11 scherm - 11
- ◆ De komma en de puntkomma: overal waar een spatie verplicht is, mag men ook een komma of een puntkomma schrijven, gevolgd door een spatie. Pal voor een komma of een puntkomma mag men eveneens een spatie schrijven
Vb. MOVE a TO b c d
MOVE a TO b, c, d
MOVE a TO b; c; d
MOVE a TO b; c; d

1.6.6.2 De functietekens

- ◆ De punt: Sluit een groep van bij elkaar behorende elementen af. Mag worden voorafgegaan door een spatie en moet gevolgd worden door een spatie. Als een punt ontbreekt, kunnen fouten ontstaan.
03 lijn-11 PIC X(7) VALUE “Dit is “.

◆ De operatoren

◆ rekenkundige operatoren

+	= Optelling
-	= Aftrekking
*	= Vermenigvuldiging
/	= Deling
**	= Machtsheffing

Er moet zowel een spatie vóór als na de rekenkundige operator komen.

◆ relationele operatoren

<	= Kleiner dan
<=	= Kleiner dan of gelijk aan
=	= Gelijk aan
>	= Groter dan
>=	= Groter dan of gelijk aan

Er moet zowel een spatie vóór als na de relationele operator komen.

◆ Logische operatoren p 78

NOT - AND - OR

1.6.7 Formaat en notatie

Door middel van formaatbeschrijvingen en van aanvullende voorschriften wordt vastgelegd hoe de taal moet worden geschreven (SYNTAX) en wat de gebruikerseffecten zijn (SEMANTIEK).

◆ Woorden in hoofdletters: gereserveerde woorden

Onderstreept: gebruik verplicht

Niet onderstreept: gebruik facultatief (om leesbaarheid te verhogen)

◆ Woorden in kleine letters: men moet een zelfgekozen naam of constante invullen

◆ Accolades: men moet kiezen uit de mogelijkheden die er staan

{ }

◆ Teksthaken: men mag dit gedeelte naar behoeven gebruiken of weglaten

[]

◆ Herhalingsteken: men mag dat gedeelte naar behoeven herhalen

...

Vb.

DISPLAY $\left\{ \begin{array}{l} \text{identifier-1} \\ \text{literal} \end{array} \right\}$

DISPLAY lijn-11
DISPLAY "HALLO"

Oefening :

SINT-MAARTEN BEDENKT integer $\left\{ \begin{array}{l} \text{MENSEN} \\ \text{KINDEREN} \end{array} \right\} \left[\text{MET} \left\{ \begin{array}{l} \text{VELE} \\ \text{VEEL} \end{array} \right\} \left\{ \begin{array}{l} \text{CADEAUX} \\ \text{SNOEP} \\ \text{PRESENTJES} \end{array} \right\} \dots \right]$

Welk van de volgende uitdrukkingen is foutief?

- ◆ SINT-MAARTEN 102 KINDEREN
- ◆ SINT-MAARTEN BEDENKT 2 MENSEN MET CADEAUX
- ◆ SINT-MAARTEN BEDENKT 5 KINDEREN MET VEEL SNOEP, VELE CADEAUX
- ◆ SINT-MAARTEN 1000 KINDEREN MET VEEL SNOEP
- ◆ SINT-MAARTEN BEDENKT 1000 MENSEN MET VELE CADEAUS MET VEEL SNOEP

2 NUCLEUS

2.1 BENOEMEN VAN VELDEN

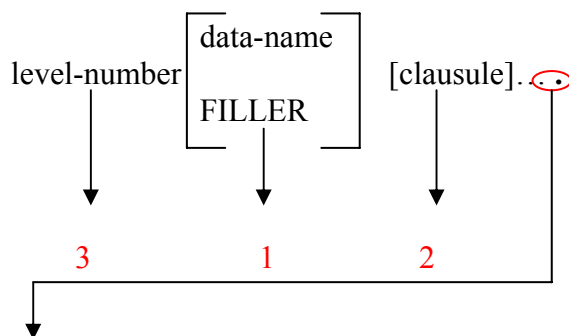
2.1.1 Algemene opbouw van een veldbenoeming

Een veld dat in een COBOL-programma gebruikt wordt moet benoemd worden.

- Elk veld:
- een naam krijgen (1)
 - aangeven van de grootte
 - aangeven van de soort
 - geven van een beginwaarde
 - optimalisatie van geheugengebruik
 - onderlinge samenhang tussen verschillende velden (3)

Veldbenoemingen in DATA DIVISION:

WORKING-STORAGE SECTION
FILE SECTION
SCREEN SECTION



Elke veldbenoeming moet met een punt afgesloten worden.

2.1.2 Niveau-nummer p 12 - 13

Niveaunummers worden gebruikt om de relatie tussen verschillende velden aan te geven op een hiërarchische wijze.

- Record: grootste eenheid van informatie die in een keer kan worden verwerkt
- Groepsvelden: zijn nog onderverdeeld
- Elementaire velden: niet onderverdeeld

Vb.:

Artikelgegevens		
Artikelcode		Omschrijving
Magazijncode	Kleurnummer	

AAL1709D	23	BUREAUSTOEL CLASSIC
----------	----	---------------------

IN COBOL:

01 ARTIKELGEGEVENS.
 03 ARTIKELCODE.
 05 MAGAZIJNCODE_____.
 05 KLEURNUMMER_____.
 03 OMSCHRIJVING_____.

Niveaunummer 01 moet in gebied A beginnen. De andere niveaunummers mogen in gebied A beginnen (doe dat beter niet)

Regels

- 1- hoogste niveau moet 01 zijn
- 2- de niveaunummers mogen lopen van 01 tot en met 49
- 3- niveaunummers van 01 tot en met 09 mogen eventueel zonder "0" worden geschreven
- 4- elk volgend niveau moet een hoger, maar niet noodzakelijk aansluitend niveaunummer hebben (mogelijkheid om tussen te voegen)
- 5- per groep moeten de niveaunummers gelijk zijn
- 6- een nieuw niveaunummer mag op gelijk niveau niet kleiner zijn dan een vorig niveaunummer, gerekend vanaf het voorgaande 01-nummer

Het niveaunummer 77

Indien men een veld wil benoemen dat geen onderdeel vormt van een ander veld en dat zelf ook niet is onderverdeeld, dan mag men daarvoor het niveaunummer 77 gebruiken.
(=elementair veld)

Vb.: 77 toets PIC X

2.1.3 PICTURE-clausule: lengte + soort

$\left. \begin{array}{l} \underline{\text{PICTURE}} \\ \underline{\text{PIC}} \end{array} \right\}$ IS character-string

- een groepsveld mag nooit een PIC-clausule bevatten
- een groepsveld is per definitie alfanumeriek
- de lengte van een groepsveld = som van de lengten van de elementaire velden

Principe van variabelen en geheugenzones

Categorieën van velden

- Numeriek veld → 9
(in het veld mogen de cijfers 0 tot 9 voorkomen en eventueel een waarde aanduiding)

Vbn. 05 kleurnummer PIC 99.
07 loon PIC 9999999.
07 loon-2 PIC 9(7).

loon

1	2	4	0	7	9	3
---	---	---	---	---	---	---

Een numeriek veld mag max 18 cijfers groot zijn

- Alfanumeriek veld → X ASCII-TABEL
(In het veld mogen alle tekens uit de tekenset van de computer voorkomen)
- Vbn. 05 magazijncode PIC XXXXXXXX. Is identiek met PIC X(8).
03 omschrijving PIC X(30).

magazijncode

A	A	L	1	7	0	9	D
---	---	---	---	---	---	---	---

- Alfabetisch veld → A
(Alleen de hoofdletters en kleine letters en de spatie kunnen voorkomen)
- Vb. 03 omschrijving PIC A(30).

- Opgemaakt veld – indexveld : zie later

Regels

- 1- De PICTURE-clausule mag alleen gebruikt worden bij een elementair veld, nooit bij een groepsveld
- 2- De kleine letter a en x mogen eveneens gebruikt worden.
- 3- De PICTURE-string mag maximum 30 tekens lang zijn.

03 omschrijving PIC XXX...X (35 tekens)
03 omschrijving PIC X(35) (5 tekens)

Nu volgt daaruit:

- 01 artikelgegevens.
03 artikelcode.
05 magazijncode PIC X(8).
05 kleurnummer 99.
03 omschrijving PIC X (30).

NAAM	SOORT	LENGTE
artikelgegevens	GV - alfanumeriek (*)	40 (**)
artikelcode	GV - alfanumeriek (*)	10
magazijncode	EV - alfanumeriek	8
kleurnummer	EV - numeriek	2
omschrijving	EV - alfanumeriek	30

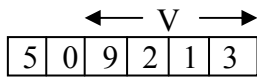
(*) een groepsveld is altijd alfanumeriek

(**) de lengte van een groepsveld = som van de lengte van de elementaire velden

2.1.4 De decimale positie

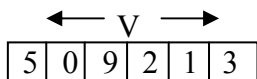
V : denkbeeldige komma

loon 5092.13
03 loon PIC 9(4)V99.



V = denkbeeldige decimale positie
Er wordt geen geheugenruimte voorzien

loon 509.213
03 loon PIC 999V999.



vb.	waarde	PICTURE	rekenwaarde
	12345	999V99	123,45
		V99999	0,12345
		99999V	12345

P : de decimale positie schuif 1 plaats op

Vb.	01 A	PIC P99	waarde 12	rekenwaarde 0,012
	01 B	PIC PPP99	waarde 12	0,00012
	01 C	PIC 99P	waarde 12	120
	01 D	PIC 99PPP	waarde 12	12000

P wordt niet meegeteld voor de lengte van het veld.

2.1.5 Waarde-aanduiding → S

Nodig als een numeriek veld zowel positieve als negatieve waarden kan hebben.

01 A PIC S99.

01 B PIC 99.

move -35 to A

move -35 to B

A

L	5
---	---

 rekenwaarde = -35

B

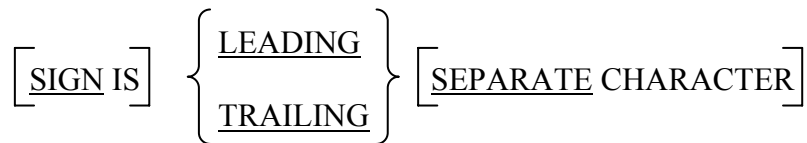
3	5
---	---

 rekenwaarde = -35

← De leverancier bepaalt zelf hoe en bij welk cijfer de waarde-aanduiding wordt toegevoegd.

De Sign-clausule

Hiermee kan men zelf bepalen waar de waarde-aanduiding moet staan



01 C PIC S99 sign leading.
move -35 to C

L	5
---	---

01 D PIC S99 sign trailing.
move -35 to D

3	N
---	---

01 E PIC S99 sign leading separate.
move -35 to E

-	3	5
---	---	---

01 F PIC S99 sign trailing separate.
move -35 to F

3	5	-
---	---	---

rekenwaarde = -35

2.1.6 De VALUE-clausule

Dient voor het geven van een beginwaarde

VALUE IS literal

➤ De VALUE-clausule voor een elementair veld

01 A PIC 999 value 10. →

0	1	0
---	---	---

rekenwaarde = 10

01 B PIC XXX value spaces. →

β	β	β
---	---	---

01 C PIC S999V99 value zero. →

v				
0	0	0	0	0

rekenwaarde = 0

01 D PIC S999V99 value -0.2. →

v				
0	0	0	2	0

rekenwaarde = -0.2

01 E PIC X(5) value "EINDE". →

E	I	N	D	E
---	---	---	---	---

01 F PIC 99PP value 1200. →

1	2
---	---

rekenwaarde 1200

01 G PIC X(10) value "ABC". →

A	B	C	β	β	β	β	β	β	β
---	---	---	---	---	---	---	---	---	---

01 H PIC X(10) value all "ABC". →

A	B	C	A	B	C	A	B	C	A
---	---	---	---	---	---	---	---	---	---

De VALUE-clausule bij een groepsveld

01 machten-van-2 value "0102040816".

03 nulde PIC 99.

03 eerste PIC 99.

03 tweede PIC 99.

03 derde PIC 99.

03 vierde PIC 99.

0	1	0	2	0	4	0	8	1	6
---	---	---	---	---	---	---	---	---	---

2.1.7 FILLER

Als men een bepaald veld geen naam wil geven mag men het woord "filler" gebruiken. Men mag ook niets schrijven.

Opm: andere clausules (USAGE, REDEFINES, JUSTIFIED, SYNCHRONIZED, RENAMES) worden later besproken.

2.2 Enkele Basisopdrachten

Cobol-statements p 35 – Zinnen p 36 – procedures p 37

2.2.1 De STOP-opdracht: de run-unit wordt beëindigd.

STOP $\left\{ \begin{array}{l} \underline{\text{RUN}} \\ \text{literal} \end{array} \right\}$

2.2.2 De ACCEPT-opdracht p 82 en p 92 - 95

Inlezen van een variabele via het klavier
zie handboek compiler

ACCEPT identifier

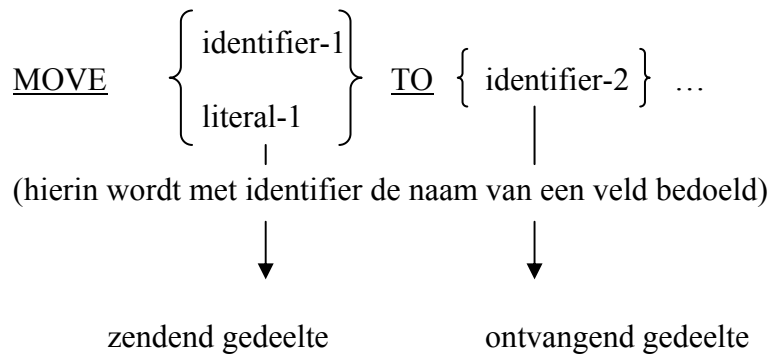
01 DATUM PIC 99/99/99 ... ACCEPT DATUM FROM DATE DISPLAY DATUM

2.2.3 De DISPLAY-opdracht p 82 en p 95 - 97

Afdrukken (tonen) van een constante of de inhoud van een variable op het scherm
zie handboek compiler

DISPLAY $\left\{ \begin{array}{l} \text{identifier-1} \\ \text{Literal} \end{array} \right\} \dots$

2.2.4 De MOVE-opdracht p 21 en p 63 – 66 en p 166



Vb.:

MOVE 12.5 to percentage

MOVE naam-1 to naam-2, naam-3

MOVE spaces to regel

	Alfabetisch	Alfanumeriek	Numeriek	Groep
Alfabetisch	J	J	N	J
Alfanumeriek	J	J	J	J
Numeriek geheel getal	N	J	J	J
Numeriek niet-geheel getal	N	N	J	J
Groep	J	J	J	J

Zendend veld
Ontvangend veld

Alfanumerieke overdracht
Numerieke overdracht

move -12345 to D

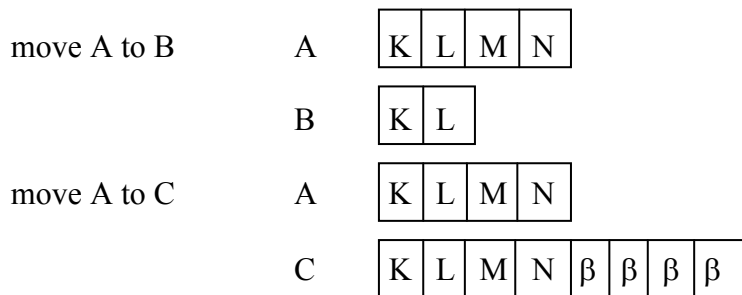
-	2	3
---	---	---

 rekenwaarde = -2300

➤ **Alfanumerieke overdracht**

De tekens worden van links naar rechts overgebracht. Als het ontvangende veld langer is dan het zendende veld, dan wordt het ontvangende veld aan de rechterkant aangevuld met spaties. Als het ontvangende veld korter is, dan worden de over te brengen waarden aan de rechterkant afgekapt. (uitz. JUSTIFIED-clausule)

01 A PIC X(4) value "KLMN".
 01 B PIC XX.
 01 C PIC X (8).

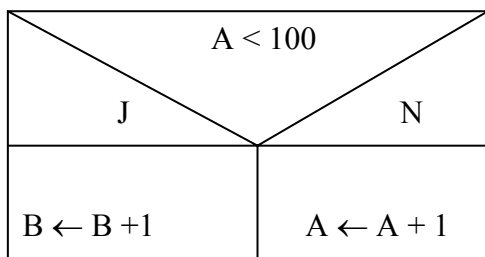


2.2.5 De IF-opdracht p 97 - 102

IF condition THEN { {statement-1} ...
NEXT SENTENCE } { ELSE {statement-2} ...
END-IF }
ELSE NEXT SENTENCE
END-IF }

Voorbeeld 1

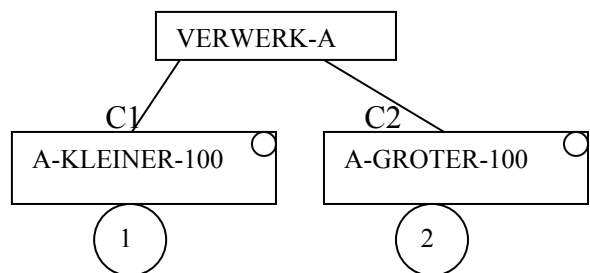
➤ NS-diagram:



➤ pseudo-code

ALS (A < 100)
 DAN B ← B + 1
 ANDERS A ← A + 1
 EINDE-ALS-DAN

Jackson:



Conditioes en opdrachten:

C1: ALS A < 100
 C2: ALS A >= 100
 1: B ← B + 1
 2: A ← A + 1