

# 1 SEQUENTIAL I/O

## 1.1 Inleiding

### 1.1.1 SEQUENTIËLE ORGANISATIE (= opslagstructuur)

Begrip record:

Elk record heeft een vaste voorganger (behalve het 1ste record), elk record heeft een vaste opvolger (behalve het laatste record). Deze relaties worden tijdens de bestandscreatie vastgelegd en zijn onveranderlijk. (Tussenvoegen van nieuwe records en laten vallen van bestaande records is niet mogelijk.)

### 1.1.2 SEQUENTIËLE TOEGANGSMETHODE (= access)

De records zijn uitsluitend toegankelijk in de fysieke volgorde waarin ze oorspronkelijk zijn geschreven.

## 1.2 ENVIRONMENT DIVISION.

→ CONFIGURATION SECTION.

→ INPUT-OUTPUT SECTION. Geeft een beschrijving van de technische aspecten van de bestanden die in het programma worden gehanteerd.

→ FILE-CONTROL → Deze paragraaf moet voor elk bestand dat in het programma wordt verwerkt een *select-opdracht* bevatten

→ I-O-CONTROL . (later)

Voorbeeld:

```
...
...
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
...
...
INPUT-OUTPUT SECTION.
FILE-CONTROL.
    SELECT artikelbestand
        ASSIGN TO "c:\student\1i ?\naam\cobol\artikel.dat"
        ORGANIZATION SEQUENTIAL
        ACCESS SEQUENTIAL
        FILE STATUS artikel-status.

    SELECT klantenbestand
        ASSIGN TO "a:\klanten.dat"
        ORGANIZATION SEQUENTIAL
        ACCESS SEQUENTIAL
        FILE STATUS klanten-status.
```



### 1.3 DATA DIVISION

→ FILE SECTION: voor elke *select*-opdracht moet hier een bestand benoemd worden.

Vervolg Voorbeeld:

DATA DIVISION.

FILE SECTION.

FD artikelbestand

FD = file description moet in zone A beginnen.

01 A-artikelrecord.

03 A-artikelnummer	PIC 9(6).
03 A-omschrijving	PIC X(30).
03 A-eenheidsprijs	PIC 9(6).
03 A-winstpercentage	PIC 99.
03 A-Btwpercentage	PIC 99.
03 A-aantal	PIC 999.

FD klantenbestand

01 K-klantenrecord.

03 K-klantennaam PIC X(30).

...

WORKING-STORAGE SECTION.

→Intern geheugen←

01 artikel-status PIC XX.

88 EOF-artikel value "10".

01 klanten-status PIC XX.

88 EOF-klanten value "10".

01 I-artikelrecord.

I → input (klavier)

03 I-artikelnummer PIC 9(6).

03 I-omschrijving PIC X(30).

03 I-eenheidsprijs PIC 9(6).

...

01 O-klantenrecord.

O → output (scherm)

03 O-klantennaam PIC X(30).

...

### 1.4 PROCEDURE DIVISION.

#### 1.4.1 De OPEN-opdracht

Deze opdracht maakt het mogelijk om in het programma een bestand te gaan verwerken.

##### 1.4.1.1 OPEN INPUT artikelbestand

Er wordt aangegeven dat men in een reeds bestaand bestand wil gaan lezen → READ-opdracht.

##### 1.4.1.2 OPEN OUTPUT artikelbestand

Men geeft aan dat men een nieuw bestand wil creëren. Direct na de uitvoering van de opdracht bestaat het bestand, het bevat op dat moment nog geen records: het is dan een leeg

bestand. Door het uitvoeren van de WRITE-opdracht wordt het bestand gevuld met records. Het is ook mogelijk om een reeds bestaand bestand te openen voor output. In dat geval komt het bestand als een leeg bestand ter beschikking; alle records zijn weg.

#### 1.4.1.3 OPEN EXTEND artikelbestand

Men geeft aan dat men records wil toevoegen aan het einde van een bestaand bestand; dit gebeurt opnieuw via de WRITE-opdracht.

#### 1.4.1.4 OPEN I-O artikelbestand

Men geeft aan dat men de records van een bestaand bestand wil wijzigen (muteren). Men moet de records eerst met een READ-opdracht lezen en dan met een REWRITE-opdracht op dezelfde plaats terugschrijven.

Voorbeelden :

```
Open input artikelbestand
If artikel-status not = "00"
    Then display "Fout in bestand"
End-if
...
Open input artikelbestand
If artikel-status = "00"
    Then close artikelbestand
        Open extend artikelbestand
    Else open output artikelbestand
End-if
```

### 1.4.2 De CLOSE-opdracht

Beëindigt de verwerking van het bestand → CLOSE <bestandsnaam>  
→ CLOSE artikelbestand

### 1.4.3 De READ-opdracht

Voerin (artikelbestand, A-record) ⇒ READ-opdracht  
O-record ← A-artikelrecord (WS) ⇒ MOVE-opdracht  
Voeruit (scherm, O-artikelnummer) ⇒ DISPLAY

Er wordt een logisch record van een bestaand bestand ter beschikking gesteld in het I/O gebied. Het bestand moet geopend zijn voor Input of I/O.

```
READ artikelbestand
    AT END set EOF-artikel to true
END-READ
Move A-artikelrecord to O-artikelrecord
DISPLAY O-artikelrecord
```

Variant:        READ klantenbestand INTO O-artikelrecord  
                  AT END set EOF-artikel to true  
                  NOT AT END DISPLAY O-artikelrecord  
                  END-READ

Opmerking:

De optie “AT END set EOF-artikel to TRUE” kan vervangen worden door de optie “AT END continue”. De compiler zal de file status waarde automatisch aanpassen.

#### 1.4.4 De WRITE-opdracht

Voerin (klavier, I-artikelnummer) (WS)    ⇒ ACCEPT

...

...

A-artikelrecord ← I-artikelrecord        ⇒ MOVE

voeruit (artikelbestand, A-artikelrecord) ⇒ WRITE

Het WRITE-commando voegt een record toe aan een bestand. Het bestand moet geopend zijn voor OUTPUT of EXTEND.

  
MOVE I-artikelrecord to A-artikelrecord  
WRITE A-artikelrecord  
= WRITE A-artikelrecord from I-artikelrecord

#### 1.4.5 De REWRITE-opdracht

Hierdoor wordt een record gewijzigd. Het bestand moet geopend zijn voor I/O en de REWRITE-opdracht moet worden voorafgegaan door een READ-opdracht.

REWRITE A-artikelrecord

REWRITE A-artikelrecord  
From I-artikelrecord

## 1.5 Opgaven

### 1.5.1 Opgave 12: toevoegen van records

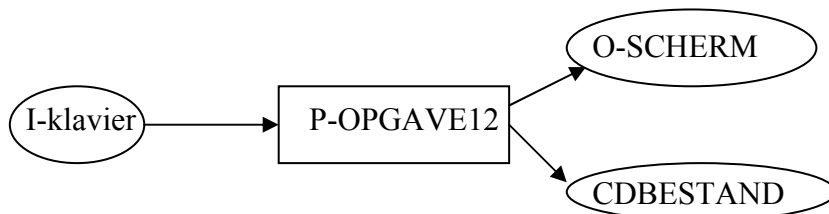
Ontwerp een COBOL-programma dat een sequentieel bestand "CDS.DAT" creëert en records aan het bestand toevoegt.

CDBESTAND		
C-CDRECORD	C-nummer	PIC X(10)
	C-titel	PIC X(10)
	C-uitvoerder	PIC X (20)
	C-aantal	PIC 99
	C-genre	PIC X(15)
	C-prijs	PIC 9(4)
	C-releasedatum	PIC 99/99/99

#### SCHERM-1

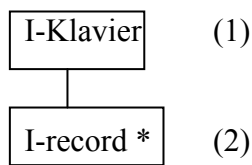
HOGESCHOOL GENT		Datum
TOEVOEGEN CD-RECORDS		
Nummer	:	
Titel	:	
Uitvoerder	:	
Aantal	:	
Genre	:	
Prijs	:	
Releasedatum	:	
Wil je nog een record (CD) toevoegen...? (J/N)		
Naam		Opgave 12

#### 1) SND

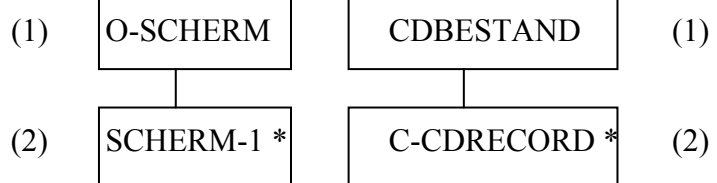


## 2) Gegevensstructuren

• *invoer*

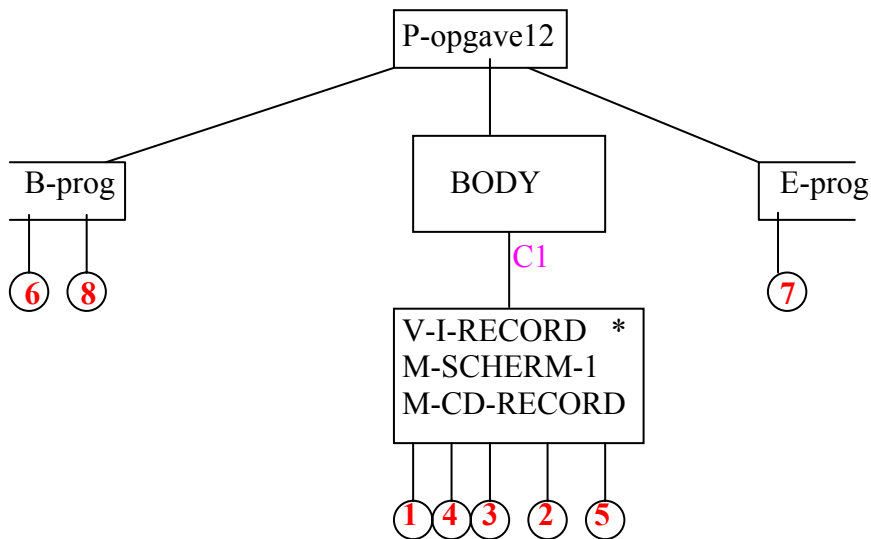


*uitvoer*



I-record: de gegevens van het record en het antwoord dat ingevoerd wordt.

## 3) Programmastructuur



## 4) Opdrachten en condities

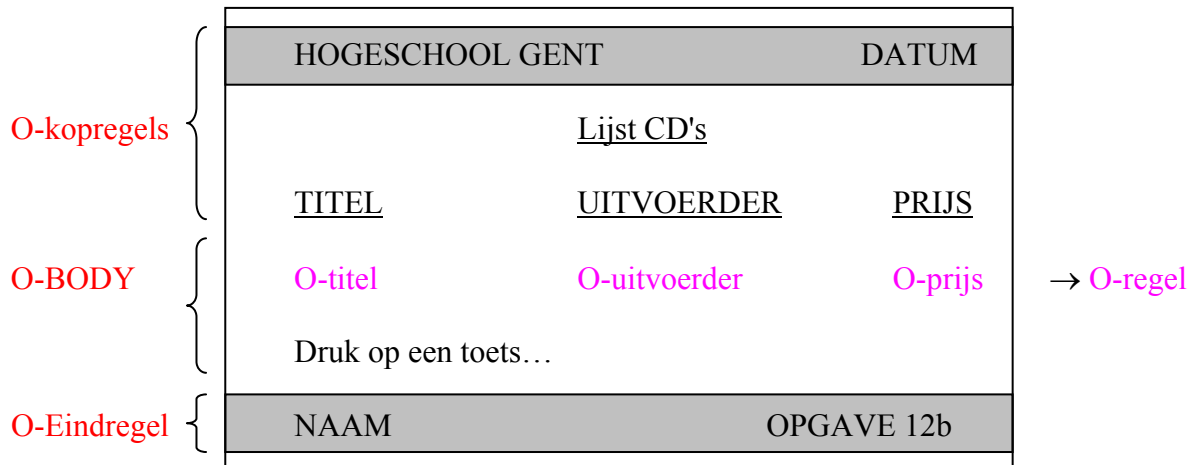
c1) ZOLANG antwoord ≠ "N" DOE...

- 1) voeruit (scherm, scherm-1)
- 2) voeruit (CDBESTAND, C-CDRECORD)
- 3) C-CDRECORD ← I-CDRECORD
- 4) voerin (klavier, I-record) meerdere velden
- 5) voerin (klavier, antwoord)
- 6) open uitvoer CDBESTAND
- 7) sluit CDBESTAND
- 8) antwoord ← "J"

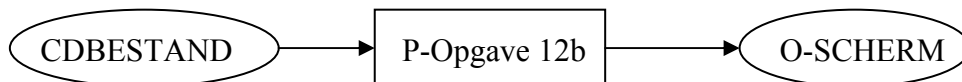
## 5) tekststap

## 1.5.2 Opgave 12 b : maken van een lijst

Ontwerp een COBOL programma dat van het sequentieel georganiseerd bestand CDBESTAND een lijst afdruckt op het scherm met volgende velden: TITEL, UITVOERDER, PRIJS.

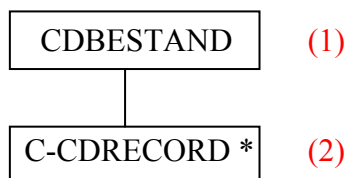


### 1) SND

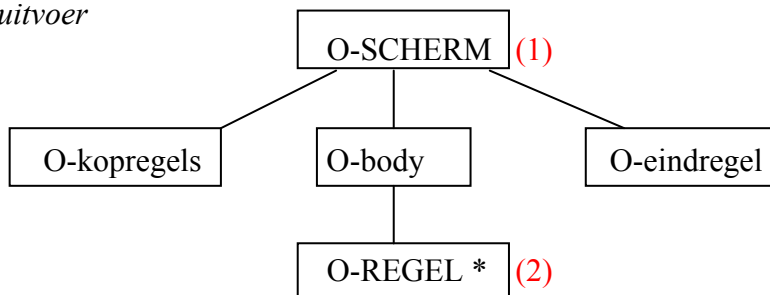


### 2) Gegevensstructuren

- invoer*

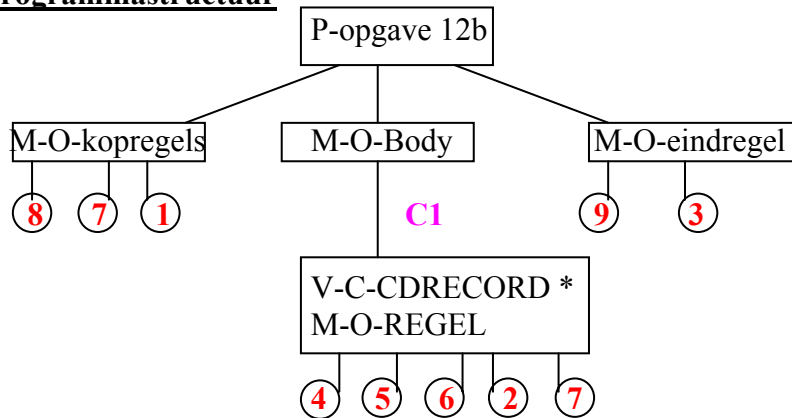


- uitvoer*





### 3) Programmastructuur



### 4) Opdrachten en condities

c1) ZOLANG NIET EOF-CDBESTAND                      DOE...

- 1) voeruit (scherm, O-kopregels)
- 2) voeruit (scherm, O-regel)
- 3) voeruit (scherm, O-eindregel)
- 4) O-TITEL ← C-TITEL
- 5) O-UITVOERDER ← C-UITVOERDER
- 6) O-PRIJS ← C-PRIJS
- 7) voerin (CDBESTAND, C-CDRECORD)
- 8) open invoer CDBESTAND
- 9) sluit CDBESTAND

in 3 stukken, omdat er meer dan 3 velden  
in het record zitten, anders wordt  
het record volledig removed.

### 5) Tekststap

P-opgave 12b

Begin

M-O-kopregels

M-O-body

M-O-eindregel

Einde

M-O-kopregels

Begin

open invoer CDBESTAND

voerin (CDBESTAND, C-CDRECORD)

voeruit (scherm, O-kopregels)

Einde

M-O-body

Begin

ZOLANG NIET EOF-CDBESTAND

DOE V-C-CDRECORD-M-O-REGEL

Einde

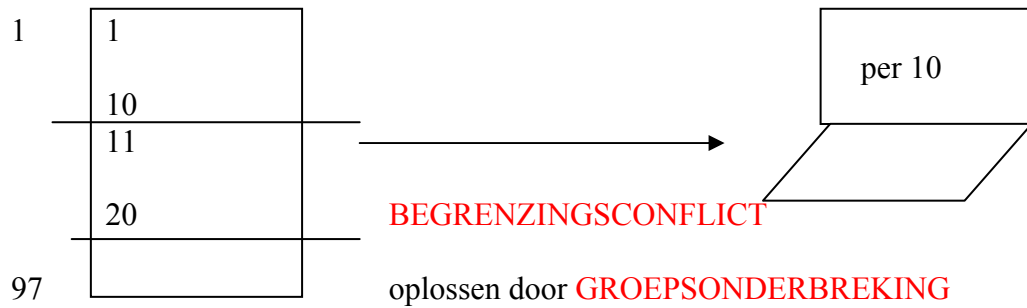
M-O-eindregel  
 Begin  
     sluit CDBESTAND  
     voeruit (scherm, O-eindregel)  
 Einde

V-C-CDRECORD-M-O-REGEL  
 Begin  
     O-TITEL ← C-TITEL  
     O-UITVOERDER ← C-UITVOERDER  
     O-PRIJS ← C-PRIJS  
     voeruit (scherm, O-regel)  
     voerin (CDBESTAND, C-CDRECORD)  
 Einde

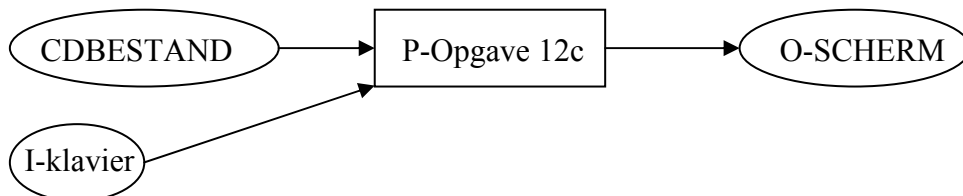
### 1.5.3 Opgave 12c

Ontwerp een COBOL-programma dat van het CDBESTAND een lijst afdruckt op het scherm (zie 12b) maar nu worden telkens 10 records getoond. Telkens er 10 records getoond worden, wordt er gevraagd om een toets te drukken om de volgende 10 te tonen.

CDBESTAND

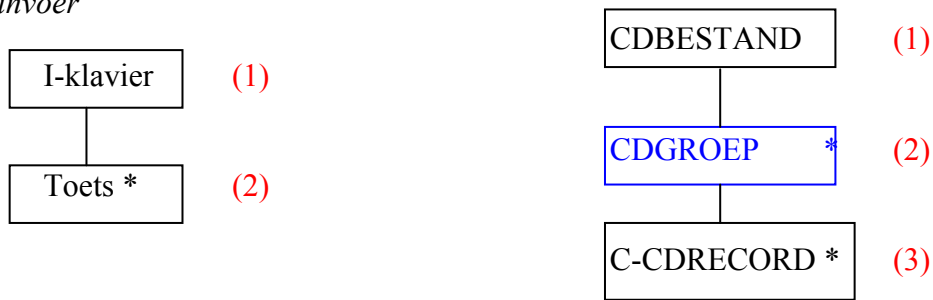


#### 1) SND

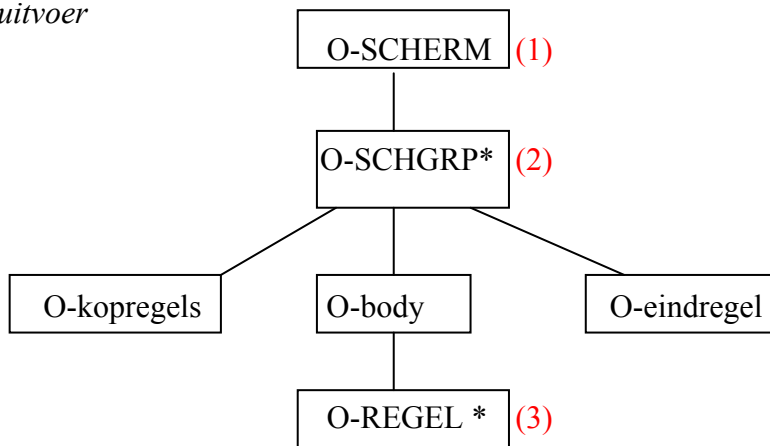


## 2) Gegevensstructuren

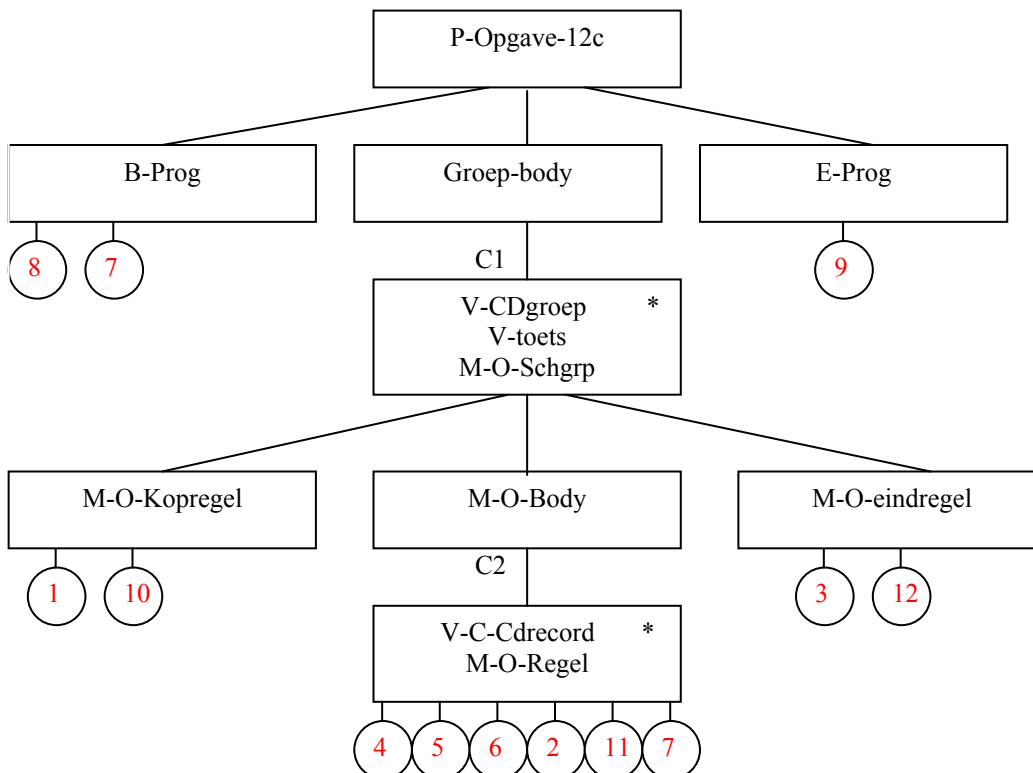
- *invoer*



- *uitvoer*



## 3) Programmastructuur



#### 4) Opdrachten en condities

- C1) ZOLANG NIET EOF CDbestand
- C2) ZOLANG NIET EOF CDbestand en Teller <= 10

- 1) Voeruit (Scherm, O-Kopregels)
- 2) Voeruit (Scherm, O-regel)
- 3) Voeruit (Scherm, O-Eindregels)
- 4) O-titel ← C-titel
- 5) O-uitvoerder ← C-uitvoerder
- 6) O-prijs ← C-prijs
- 7) Voerin (CDbestand, C-CDrecord)
- 8) Open invoer CDbestand
- 9) Sluit CDbestand
- 10) Teller ← 1
- 11) Teller ← Teller + 1
- 12) Voerin (Klavier, toets)

#### 5) Tekststap

P-oef-12c

Begin

B-Prog

Groep-body

E-Prog

Einde

B-Prog

Begin

Open Invoer CDbestand

Voerin (CDbestand, C-CDrecord)

Einde

Groep-body

Begin

ZOLANG NIET EOF CDbestand

DOE V-CDgroep-V-Toets-M-O-Scherm

EINDE-ZOLANG-DOE

Einde

V-CDgroep-V-Toets-M-O-Scherm

Begin

M-O-KopRegel

M-O-Body

M-O-Eindregel

Einde

E-Prog

Begin

Sluit CDbestand

Einde

M-O-Kopregel

Begin

Voeruit (Scherm, O-Kopregels)

Teller  $\leftarrow$  1

Einde

M-O-Body

Begin

ZOLANG NIET EOF CDbestand En Teller  $\leq$  10

DOE V-C-CDrecord-M-O-Regel

EINDE-ZOLANG-DOE

Einde

M-O-Eindregel

Begin

Voeruit (Scherm, O-eindregels)

Voerin (Klavier, Toets)

Einde

V-C-CDrecord-M-O-regel

Begin

O-titel  $\leftarrow$  C-titel

O-uitvoerder  $\leftarrow$  C-uitvoerder

O-prijs  $\leftarrow$  C-prijs

Voeruit (Scherm, O-regel)

Teller  $\leftarrow$  Teller + 1

Voerin (CDbestand, C-CDrecord)

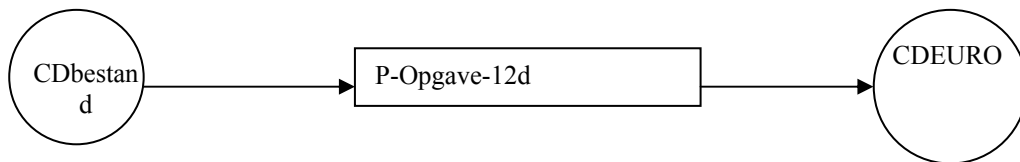
Einde

### 1.5.4 Opgave12d

Ontwerp een COBOL-programma dat van het sequentieel georganiseerd bestand CDBESTAND een nieuw sequentieel georganiseerd bestand CDEURO creëert

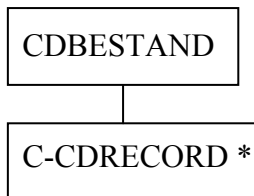
CDEURO-RECORD	}	CE-nummer	PIC X(10)
		CE-titel	PIC X(10)
		CE-uitvoerder	PIC X (20)
		CE-aantal	PIC 99
		CE-genre	PIC X(15)
		CE-prijs	PIC 9(4)
		CE-releasedatum	PIC 99/99/99

#### 1) SND

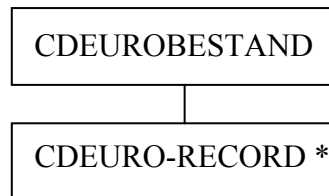


#### 2) Gegevensstructuur

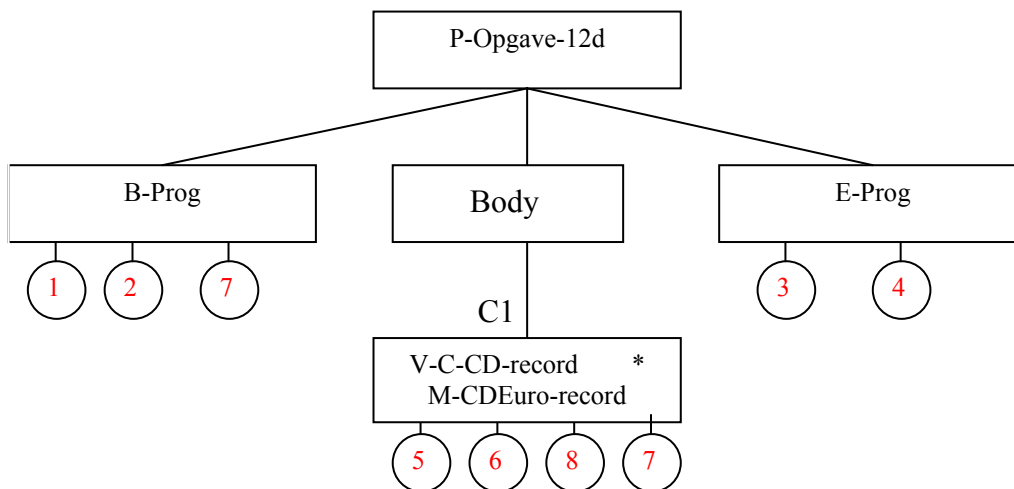
Invoer



Uitvoer



#### 3) Programmastructuren



#### 4) Opgaven en condities

C1) ZOLANG NIET EOF CDbestand

- 1) Open Invoer CDbestand
- 2) Open Uitvoer CDEuro
- 3) Sluit CDbestand
- 4) Sluit CDEuro
- 5) CE-prijs  $\leftarrow$  C-prijs / 40,399
- 6) CDEuro 'andere gegevens'  $\leftarrow$  CDbestand 'andere gegevens'
- 7) Voerin (CDbestand, C-CDrecord)
- 8) Voeruit (CDEuro, CDEURO-record)

#### 5) Tekststap

P-Opgave-12d

Begin

B-Prog

Body

E-prog

Einde

Body

Begin

ZOLANG NIET EOF CDbestand

DOE V-C-CD-record-M-CDEuro-record

EINDE-ZOLANG-DOE

Einde

B-Prog

Begin

Open Invoer CDbestand

Open Uitvoer CDEuro

Voerin (Cdbestand, C-CDrecord)

Einde

V-C-CDrecord-M-CDEuro-record

Begin

CE-prijs  $\leftarrow$  C-prijs / 40,3399

CDEuro 'andere gegevens'  $\leftarrow$  CDbestand 'andere gegevens'

Voeruit (CDEuro, CDEuro-record)

Voerin (CDbestand, C-CDrecord)

Einde

E-Prog

Begin

Sluit CDbestand

Sluit CDEuro

Einde

