

# COMAL80

## info

Periodiek verschijnend tijdschrift van de

COMAL80 gebruikersgroep Nederland

secretariaat  
p.a. Stichting Pascal  
Bezuidenhoutseweg 225  
2594 AL DEN HAAG



NOT SPECIAL

```

BASIC
10 E=35
20 T=9
30 D=E+T
40 PRINT D

```

Zoals blijkt maakt COMAL onderscheid tussen keywords (in hoofdletters) en andere woorden (bijvoorbeeld variabelen in kleine letters). COMAL heft ook het dubbelzijdige gebruik van het = - teken op.

2 Door de verschillende loop-structuren krijgt de programmeur de gereedschappen in handen om zonder GOTO-statements loops te construeren, immers het gebruik van GOTO zeker bij IF...THEN... maakt juist BASIC-programma's zo ondoorzichtig.

```

COMAL
0010 num := 1; milj := 1000000
0020 teller := 0
0030 REPEAT
0040   num:=num*2
0050   teller:=teller+1
0060 UNTIL num>milj
0070 PRINT teller;" keer"

```

```

BASIC
10 N=1 : M=1000000
20 T=0
30 N=N*2
40 T=T+1
50 IF N<=M THEN GOTO 20
60 PRINT T;" KEER"

```

Bovenstaande programma's beantwoorden de vraag: "Hoe vaak moet het getal 1 worden verdubbeld om meer dan 1 miljoen te krijgen?"

Merk op dat in het BASIC-programma de ontkenning van "groter dan" voorkomt. Het COMAL-programma beantwoordt de vraag op natuurlijke wijze.

Een tweede voorbeeldje.

```

COMAL
FOR T:=1 TO 0 DO
  PRINT "niks"
NEXT T

```

```

BASIC
10 FOR T=1 TO 0
20 PRINT "IETS"
30 NEXT T

```

Er zijn BASIC's die in dit laatste BASIC-programma inderdaad IETS afdrucken. COMAL doet niks.

3 De mogelijkheid om meerdere statements verdeeld over meerdere programmalijnen binnen de IF...THEN-constructie te plaatsen verduidelijkt de structuur van het programma en geeft een natuurlijke afwikkeling daarvan.

```

COMAL
INPUT a
IF a < 20 THEN
  REPEAT
    a:=a*2
  UNTIL a>20
ELSE
  REPEAT
    a:=a/2
  UNTIL a<20
ENDIF
PRINT a

```

```

BASIC
10 INPUT A
20 IF A<20 THEN GOTO 60
30 A=A/2
40 IF A>=20 THEN GOTO 30
50 GOTO 80
60 A=A*2
70 IF A<=20 THEN GOTO 60
80 PRINT A

```

De gestructureerde BASIC's kennen de in het COMAL-programma ALGOL-achtige structuur niet. Overigens is het COMAL-programma eveneens een illustratie van punt 7, de layout.

4 De CASE-structuur vindt zijn ongelijke in de ON...GOTO in BASIC. In COMAL wordt er echter niet gesprongen.

```

COMAL
INPUT a
CASE a OF
  WHEN 1
    PRINT "Was 1"
  WHEN 2
    PRINT "Was 2"
  OTHERWISE

```

```

BASIC
10 INPUT A
20 IF A<>1 AND A<>2 THEN GOTO 60
30 ON A GOTO 40, 50
40 PRINT "WAS 1": GOTO 70
50 PRINT "WAS 2": GOTO 70
60 PRINT "GEEN 1 OF 2"
70 PRINT "EINDE"

```

N.B. Is het de lezer opgevallen dat in de laatste twee COMAL-programma's geen regelnummers meer staan? Dat hoeft ook niet!

5 Met betrekking tot het gebruik van PROCedures in COMAL en GOSUB in BASIC herhalen we alleen de woorden van Roy Atherton (director Computer Education Centre, Reading, England): "GOSUB 70 is one of the most dismal (=trieste) statements in computing." In onderstaand COMAL-voorbeeld staat een procedure met parameters (formele variabelen, in dit geval x en y).

```

COMAL
a:=3; b:=5
EXEC som(a,b)
EXEC som(8,a)
PRINT "EINDE"
PROC som(x,y)
  totaal:=x+y
  PRINT totaal
ENDPROC som

```

```

BASIC
10 A=3 : B=5
20 X=A : Y=B
30 GOSUB 70
40 X=8 : Y=A
50 GOSUB 70
60 PRINT "EINDE": STOP
70 T=X+Y
80 PRINT T
90 RETURN

```

In beide gevallen wordt afgedrukt:

```

8
11
EINDE

```

Opgemerkt kan nog worden dat de procedures in COMAL ook recursief (in de procedure de procedure zelf weer aanroepen) gebruikt kunnen worden:

```

COMAL
EXEC doe
DATA 1,2,3,4,5,6,0
PROC doe CLOSED
  READ x
  IF x<>0 THEN EXEC doe
  PRINT x
ENDPROC doe

```

```

BASIC
Opdracht aan de lezer.
- Wat wordt door bovenstaand programma afgedrukt?
- Schrijf een overeenkomstig Basic-programma

```

De variabele x in het programma is lokaal (door het gebruik van CLOSED). Deze variabele heeft daardoor alleen betekenis in de procedure zelf.

N.B. Ook als CLOSED wordt weggelaten "loopt" het programma.

7 Alle implementaties van COMAL bevatten de automatische layout van de programmatuur. Dit geeft niet alleen een goed zicht op de structuur maar het biedt ook een mogelijkheid om systematisch naar fouten in de programma's te zoeken.

8 Door het duidelijk verschil tussen de "afsluiters" wordt de PASCAL-kwaal END; END; ... vermeden. Gemakkelijk kan worden vastgesteld of een PROCEDURE, een CASE of een WHILE wordt geEND:

In het bovenstaande hebben we ons wat betreft BASIC gehouden aan een bepaald dialect. Wat toegegeven moet worden is, dat in een enkel geval de spaghetti-structuur van BASIC wat is overdreven, en ook dat de voorbeelden in een ander dialect zeker ook een ander uiterlijk (doorzichtiger) zouden hebben gekregen. De faciliteiten die COMAL verder te bieden heeft (strings, files e.d.) zijn buiten beschouwing gelaten.

# COMAL-structuren of waarom geen BASIC

Dick Klingens  
leraar wis- en computerkunde  
S.G. De Krimpenerwaard  
Krimpen aan de IJssel

LEER HET KIND DE EERSTE  
BEGINSELEN NAAR DE EISCH  
ZIJNS WEGS; ALS HIJ OOK  
OUD GEWORDEN ZAL ZIJN,  
ZAL HIJ DAARVAN NIET AF-  
WIJKEN.  
Spreuken, 22-6

COMAL-80 is een computertaal die is ontstaan in het onderwijs; een feit dat eigenlijk niet voldoende kan worden benadrukt. En wat daarbij belangrijk is; het is een taal die door leraren in hun onderwijssituatie verder is ontwikkeld en is uitgetest. Is COMAL echter een taal die boven BASIC moet worden verkozen om als voertaal te dienen in het Nederlandse onderwijs? Met andere woorden moet het Nederlandse onderwijs het Deense en Ierse volgen, immers in die landen wordt bij het computeronderwijs alleen maar met COMAL gewerkt.

Voor een antwoord op deze vraag wordt gegeven allereerst enkele feiten over BASIC. BASIC is ongetwijfeld de meest verspreide computertaal; BASIC is gemakkelijk te leren; de BASIC-interpret is klein genoeg om in ROM van een goedkope micro-computer te worden ingebouwd; BASIC heeft een vriendelijke werkomgeving; van BASIC bestaan een groot aantal dialecten: APPLE-BASIC, BASIC-PLUS, PET-BASIC, MICROSOFT-BASIC, enz.; BASIC is niet gestandaardiseerd. Te beginnen bij dit laatste.

COMAL is geen BASIC-dialect. Natuurlijk, COMAL is ontstaan uit BASIC; COMAL bevat echter ook elementen van PASCAL, maar het heeft zijn eigen specificaties die vastgelegd zijn in een COMAL-report. COMAL is niet een gestructureerde BASIC, maar is de gestructureerde BASIC. Het feit dat vele andere van structuur-elementen voorziene versies van BASIC in de laatste tijd zijn verschenen toont aan dat de combinatie van de eenvoud van BASIC en de goede controle-structuren van PASCAL een juiste combinatie is. COMAL heeft alle mogelijkheden die door Dijkstra, Hoare en Wirth zo noodzakelijk worden geacht om te komen tot leesbare, begrijpelijke programma's, waarin zich tevens de structuur van het probleem waarvoor het programma geschreven is kan weerspiegelen. Kortom, COMAL biedt de faciliteiten die leiden tot een goede programmeerstijl. We zetten deze faciliteiten eens op een rijtje:

betekenisvolle namen van variabelen  
verschillende loop-structuren: FOR...NEXT, REPEAT...UNTIL, WHILE...ENDWHILE, LOOP...ENDLOOP  
IF...THEN...ELSE...; binnen deze structuur kunnen rijtjes statement worden geplaatst  
CASE-statement voor meerkeuze-mogelijkheden onder naam gedefinieerde procedures (subroutines)  
procedures waarin variabelen lokaal kunnen worden gedefinieerd, en waarin het ook mogelijk is variabelen formeel te gebruiken  
programma-layout die aansluit bij de structuur van het programma

bij elke verschillende structuur behoren de verschillende keywords, die de structuur afsluiten zoals ENDPROC, ENDWHILE, ENDF.

We lichten bovengenoemde punten hieronder één voor één, soms met een programmaatje, kort toe.

Het belang van lange variable-namen zal niemand ontkennen; ook diverse BASIC-versies bedienen zich ervan. COMAL accepteert zelfs namen met een lengte van 80 karakters.

```

COMAL
0010 eerste:=35
0020 tweede:=9
0030 derde:=eerste+tweede
0040 PRINT derde

```

De voorbeelden die gegeven zijn illustreren de structuren van COMAL. Een realistisch programma dat alle structuren dan kan zeker worden gegeven, maar zo'n programma bevat dan stellig ook een aantal technische details die de helderheid weer zouden vertroebelen.

Vandaar het volgende programma dat ontleend is aan het boek "Structured Programming with COMAL" van Roy Atherton.

### Opdracht aan een grondwerker

Er moeten gaten voor palen worden gegraven op elke hoek van een terrein; ook langs de vier zijden moeten gaten komen, en wel op afstand van 15 meter van elkaar.

Als de grond ter plekke hard is, moet het gat 30 cm diep zijn, als de grond zacht is 50 cm.  
Als er een boom in de weg staat moet de firma Exarbor worden ingeschakeld. Is dit het geval dan moet de grootte van de boom worden opgegeven. Exarbor zal dan mededelen wat er gedaan moet worden.

De afmetingen van het terrein kunnen worden afgelezen van een kaart.

Of de grond hard of zacht is, of er een boom in de weg staat of niet, wordt "at random" bepaald.

Bovenstaande opdracht zetten we om in een COMAL-programma.

```
//grondwerk
DIM boomgrootte$(3) OF 6
FOR k:= 1 TO 3 DO
  READ boomgrootte$(k)
NEXT k
DATA "klein", "middel", "groot"
FOR zijde:= 1 TO 4 DO
  PRINT "Graaf een hoekgat."
  READ zijde_lengte
  REPEAT
    boom:=RND(0, 1)
    IF boom THEN EXEC exarbor
    (boomgrootte$(RND(1,3)))
    zacht:=RND(0,1)
    IF zacht THEN
      PRINT "Graaf een gat van 50 cm diep."
    ELSE
      PRINT "Graaf een gat van 30 cm diep."
    ENDIF
    zijde_lengte:=zijde_lengte-15
  UNTIL zijde_lengte<15
NEXT zijde
//
PROC exarbor(boomtype $)
CASE boomtype$ OF
WHEN "klein"
  PRINT "Trek de boom uit de grond."
WHEN "middel"
  PRINT "Hak de takken eraf."
  PRINT "Trek de stam met de wortel uit de grond."
WHEN "groot"
  PRINT "Zaag de boom om."
  PRINT "Graaf de wortels uit."
OTHERWISE
  PRINT "Boomtype onbekend."
ENDCASE
ENDPROC exarbor
//
DATA 100, 120, 140, 110
```

We herhalen nog eens de vraag die in de eerste alinea van dit artikel is gesteld:

"Is COMAL een taal die in het Nederlandse onderwijs boven BASIC moet worden verkozen?"

Het zou onjuist zijn alleen uit de gegeven voorbeelden af te leiden dat het antwoord op die vraag "ja" zou moeten luiden. Verdere bestudering van COMAL zal echter een ieder die het met het onderwijs in/en de informatica goed meent, tot hetzelfde antwoord brengen als Borge Christensen en zijn collega's, toen die vraag met betrekking tot het Deense onderwijs werd gesteld: "Ja, natuurlijk, COMAL!"

### BEGINNERSHOEKJE

(Voor hen die nog niet kunnen, maar wel willen.)  
door Wim van Drongelen.

Het is soms heel frustrerend om lid te worden van een vereniging. Als "beginner" wordt je overspoeld met informatie waar je weinig mee kunt, omdat je nog niet mee kunt praten over de betreffende onderwerpen. Zeker wanneer de vereniging zich bemoeit met computers en programmeertalen is er een groot verschil tussen de "niveau's" van de leden. Er zijn "experts", "profs", gevorderden en beginners; en helaas zijn de verschillen te groot om een goede samenspraak te onderhouden.

Zo niet bij onze COMAL-80 GebruikersGroep. Wel zijn er de niveauverschillen, maar niet de communicatieproblemen. Misschien komt dat doordat de meeste leden met het onderwijs te maken hebben? Voor hen die zich desondanks "beginner" voelen, is deze kolom bedoeld. Het is niet de bedoeling dat dit een programmeercursus wordt, maar in het BEGINNERSHOEKJE zullen telkens kleine programma-voorbeeldjes met uitleg staan. Misschien nodigt dit uit tot vragen stellen, of tot "verbeteren" van de voorbeelden.

Ma deze uitgebreide inleiding is het misschien goed om een klein programma te tonen en te vertellen wat het moet doen.

getal:=RND(1,10) betekent dat de variabele "getal" een willekeurige waarde tussen 1 en 10 moet worden toegekend.

REPEAT spreekt voor zichzelf: vanaf hier wordt een programma-onderdeel herhaalt totdat (UNTIL) er aan een bepaalde voorwaarde wordt voldaan.

UNTIL geraeden=TRUE is de voorwaarde waaraan voldaan zal moeten worden voordat de herhaling wordt gestaakt.

geraden=false moet voorkomen dat al aan de voorwaarde is voldaan.

INPUT "raad...": "raad vraagt de gebruiker om een getal; de tekst tussen aanhalingstekens met daarna de dubbele punt zijn als geheel naar keuze (met tekst is het programma wel duidelijker voor de gebruiker).

IF raad=getal THEN DO spreekt ook weer voor zich: wanneer het door de gebruiker ingegeven getal "raad" kleiner is dan het willekeurige getal "getal" zal de computer het volgende uitvoeren: "Mijn ...." drukt een tekst af op het scherm.

ELIF raad=getal THEN DO doet hetzelfde als IF THEN DO maar altijd in samenwerking daarmee.

ELSE doet wederom hetzelfde als IF THEN DO en eveneens in samenwerking daarmee, maar bestraft alle andere, niet genoemde toestanden..

geraden=true maakt dat aan de voorwaarde is voldaan, zodat de herhaling kan worden gestaakt. Na een IF ... THEN DO, een ELIF ... THEN DO en een ELSE kunnen verscheidene regels opdrachten worden geplaatst, die worden uitgevoerd wanneer de beslissing "waar" is.

ENOIF zorgt ervoor dat de beslissingstoestand wordt afgesloten; vergeten we dit statement, dan zal er bij het "runnen" een foutmelding komen: programmastructuur niet afgesloten.

Dit korte programma kan natuurlijk uitgebreid worden zodat het meerdere malen wordt uitgevoerd. Als we eraan toevoegen: DO15 FOR keer:=1 TO 5 DO en DO142 NEXT keer, dan zal het programma vijf maal een getal genereren en daarnaar laten raden.

We hebben nu kennis gemaakt met enkele statements van COMAL-80 die ook in sommige BASIC-dialecten voorkomen. Een duidelijk verschil tussen BASIC en COMAL-80 is dat de laatste de gebruiker uitnodigt tot goed gestructureerd programmeren, terwijl BASIC rommelig werk toestaat. Ook foutjes worden aan de gebruiker direct herkend en ter verbetering door de gebruiker getoond.

Het BEGINNERSHOEKJE kan voor beginners een stimulans zijn om te leren programmeren, maar voor gevorderden kan het een prikkel betekenen om kleine programma's te bedenken en in te zenden. Ik wil graag van u weten of deze rubriek mogelijkheden biedt voor zowel beginners als gevorderden. Ook kunt u vragen met betrekking tot programmeren in COMAL-80 inzenden naar ons secretariaat. We zullen ze of zelf beantwoorden of ter oplossing afrukken in het volgende nummer van dit blad (dan kunnen zelfs de "experts" hun hart ophalen).

Zoals elders in dit blad te lezen staat, wordt er les gegeven in programmeren met COMAL80 op verschillende computersystemen. Ik kan me voorstellen dat er (veel) meer mensen zijn die willen leren programmeren. Wilt u dat ook leren, schrijf dan aan ons secretariaat of bel een bestuurslid van de GebruikersGroep op om informatie in te winnen: misschien kunnen we dan meer cursussen organiseren ook in uw omgeving.

bewegen. Ook daarvoor zijn een aantal commando's/statements in COMVAL opgenomen. De sprites worden opgebouwd in een string bestaande uit 64 tekens. Deze strings kunnen op disk (of kassette) worden opgeslagen voor later gebruik in verschillende programma's.

In het bestek van dit blad kan helaas geen volledige beschrijving van het gebruik van de nieuwe mogelijkheden worden gegeven. In het orderstaande zal echter wel op een aantal aspecten wat dieper worden ingegaan.

Allereerst COMVAL zelf.

## 2. De taal zelf

Bij de C-64-implementatie, die voor de CBM-machines is verzorgd door UniComal Aps in Lyngby (Denemarken), is uiteraard uitgegaan van het jongste definitieve COMVAL-rapport (kernel). Bijna alle daarin voorkomende COMVAL-woorden kunnen worden gebruikt, hier en daar echter met wat kleine aanpassingen en in een enkel geval met een fraaie uitbreiding.

We beschrijven enkele zaken.

Na het opstarten van COMVAL (via het BASIC-ROM van de C-64) kunnen de programmaregels voorzien van een automatisch verspreidend regelnummer worden ingetoetst. Na het intoetsen van elke regel vindt controle op juiste syntax plaats. Er zijn een 60-tal foutmeldingen (niet alle betrekking hebbend op de regelsyntax) mogelijk. De foutboodschappen zijn opgeslagen in een sequentiële diskette COMPALERRORS. Na het niet-standaard-commando 'SETMSG-' verschijnen alleen foutnummers. De COMPALERRORS kunnen door de gebruiker zonnig worden aangepast aan de behoefte, omdat ook een programma dat de meldingen in de file plaatst op de systeemdiskette voorkomt. Dit programma moet ook gebruikt worden om de file op andere te gebruiken diskettes te plaatsen.

Via het LIST-commando verschijnen de regels zoals gebruikelijk waarnodig ingesprongen op het scherm. Er is echter geen zichtbaar onderscheid tussen COMVAL-woorden en namen, omdat de listings of in kleine letters of alleen in hoofdletters worden gegeven.

Het EDIT-commando moet gebruikt worden bij het wijzigen van in het geheugen staande programma's of programma-delen. De C-64 bezit een scherm-editor die onder COMVAL gaat tot zijn recht komt. Deze editor werkt op het uit 25 regels en 40 kolommen bestaande scherm.

## COMVAL-60 voor de Commodore 64

### 1. Inleiding

Reeds een aantal jaren bestaan er enkele COMVAL-versies voor Commodore-machines: CBM COMVAL versie 0.11 voor elke 32k CBM-machine of PET-computer en versie 1.02 voor de 96k machines. Versie 0.11 is in voorkomende gevallen gesplitst in twee stukken: COMVALBASIC - de input module en COMVALDEX - de execute module.

Voor de jongste tijd uit de CBM-keuken - de Commodore 64 of C-64 - is er een van versie 0.11 afgeleide pre-release versie 01 (prepared by B.D. Grainger for ICPUg and Commodore). Deze laatste versie wordt niet door Commodore verder ondersteund, in de eerste plaats omdat er op grote schaal ongeoorloofd kopiëren van zijn gemaakt; in de tweede plaats, omdat de versie niet voldoet aan de opgestelde COMVAL-60-standaard.

Bij mijn laatste bezoek aan Denemarken zag ik een huize van George Christensen een nieuwe CBM COMVAL-versie: 0.12, die geheel is toegesneden op de C-64.

De Commodore 64 is een 64k machine die beschikt over een aantal grafische mogelijkheden, waarvoor binnen COMVAL een aantal sleutelwoorden zijn opgenomen. Dit blad is zeker geen plaats om de mogelijkheden van de C-64 uiteen te zetten, echter waar COMVAL-60 van die specifieke eigenschappen gebruik maakt, past een enkel woord.

In de eerste plaats is het mogelijk op een ogenaamd high resolution scherm dat bestaat uit een raster van 300 x 190 punten elk punt van een kleur (C-64 staat 16 kleuren toe) te voorzien. Binnen COMVAL-60 zijn een aantal zogenaamde turtle-commando's en turtle-statements opgenomen (ook voorkomend in de programmeer taal LOGO), waarmee rasterficken op genoemd HR-scherm kunnen worden gemaakt. De turtle is een klein driehoekje, dat via commando/statements over het scherm kan worden bewogen en dat daarbij een spoor van gekleurde lijntjes achterlaat.

Wat de procedures en functies in COMVAL wordt bedoeld voor een niet-onaanvaardig toepassingsgebied voor COMVAL bereikbaar.

In de tweede plaats bestaat de mogelijkheid op het grafische scherm zelf-ontworpen liguurtjes, zogenaamde sprites, te laten

In C-64/COMVAL zijn alleen integer en feal functies mogelijk. Het gemis aan stringfuncties kan ook hier worden goedgehaakt door gebruik te maken van REF-variabelen.

Strings kunnen evenmin worden gebruikt als element in een PRINT USING statement.

C-64/COMVAL bevat geen CURSOR- of AT-state-ment. In plaats daarvan kunnen cursor-controle-tekens in strings worden opgenomen. Het gebruik daarvan bevoordert echter de leesbaarheid van het programma niet, omdat deze tekens als grafische tekens in listings verschijnen, waarbij in voorkomende gevallen zelfs een of meerdere oelen van de string wegvallen (alleen in de listing!).

Het betrekking tot het werken met RANDOM-files is er een aardige bijzonderheid. Indien een record bestaat uit twee (of meer) items, zeg een reel getal en een string (in deze volgorde), dan is het mogelijk een gedeelte van dit record (opnieuw) te beschrijven door de begin-positie op te geven na het recordnummer. Daarbij gaat het eerste gedeelte van het record niet verloren. Een gedeelte van het record kan op die manier ook worden gelezen.

Voorbeeld.

```
// DIM NAAMS OF 20
OPEN FILE 2, "VOOR8", RANDOM 50
RECORDS=3
A:=80
// een reel neemt 5 byte in een record
NAAMS="KOLLOUWEN"
WRITE FILE 2,RECORD: A, NAAMS
// we lezen nu vanaf positie 6 in het
// record:
READ FILE 2,RECORD,6: NAAMS
PRINT NAAMS
// we wijzigen record 3 vanaf positie 6:
NAAMS="TEKENS"
WRITE FILE 2,RECORD,6: NAAMS
// uitlezen gehele record:
READ FILE 2,RECORD: A, NAAMS
PRINT A; NAAMS
CLOSE
//
```

Het programma heeft als uitvoer:  
KOLLOUWEN  
80 TEKENS

COMVAL maakt gebruik van het BASIC Disk Operating System van de C-64 (dit DOS zit in ROM). Dit heeft tot gevolg, dat operaties op de diskette zich hiernaar richten. Daarom zijn aan COMVAL een aantal statements toegevoegd die kunnen aangeven of bij LOAD, DELETE en OPEN een file gevonden is

Het niet mogelijk in een programma gebruik te maken van dezelfde namen: komt TEST als (reel) variabele voor, dan zijn TEST\$ en TEST#, als mede TEST als procedure- of functienaam niet toegestaan.

Na RUN wordt een test op de juiste structuur van het programma gedaan. Bij de eerst ontkende onjuiste structuur verschijnt een foutboodschap.

Fraai is de mogelijkheid om een programma, dat geheel bestaat uit procedures en/of functies te runnen. Hierbij worden door het systeem de gebruikte namen onthouden, zodat ze ook gebruikt kunnen worden in commando's.

Daarbij is het prettig dat EXEC optioneel is. Via het commando 'SETEXEC\*' verschijnt EXEC voor de naam van de procedures in de procedure-toepen. Daardoor is bijgedragen aan de compatibiliteit tussen verschillende versies.

Als voorbeeldjes:

```
// STER; STER // meerdere procedure-roepen op
// // een regel mag hier !
// //
PROC STER
PRINT CHR$(147), // wisscheren,cursor home
FOR I:=1 TO 20 DO
FOR U:=1 TO 30 DO
PRINT "#",
PRINT U
ENDFOR I
ENDPROC STER
//
```

Dit programma vult een gedeelte van het scherm tweemaal (zie de beide procedure-roepen op deze regel) met een rechtehoek van 30 bij 20 sterretjes. Nadat het programma een keer heeft gelopen, kan STER dus ook als commando worden gegeven, met hetzelfde effect.

Mark op, dat in plaats van NEXT, logisch genoeg, ENDFOR gebruikt wordt. NEXT kan wel worden ingetoetst, maar wordt door het systeem vervangen door ENDFOR. De naam van de variabele na NEXT en ook na ENDFORPROC of ENDFUNC (einde van een functie) kan worden weggelaten. Bij eerste uitvoering van het programma worden de namen automatisch door het systeem toegevoegd.

Functies en procedures kunnen CLOSED worden gedefinieerd, waardoor zonnig lokale variabelen kunnen worden geteerd. Het IMPORT-(GLOBAL-)statement is echter niet geïmplementeerd.

niet (statement STATUS en STATUS\$). Ook het  
materiaal van nieuwe diskettes en het kopiëren  
n files op dezelfde diskette maar onder een  
dere naam geschiedt onder COMVAL (statement:  
SS).

Het uitlezen van een toets direct van het  
etsenbord verloopt via de string-functie KEYS.  
orbeeld.

```
//
DIM VERDERS OF 1
REPEAT
DOEN
VERDERS=KEYS
UNTIL VERDERS="J"
PRINT "op 'J' GEDRUKT"
//
PROC DOEN
PRINT "HE HERHALEN"
ENDPROC DOEN
//
```

zien op het scherm de regel

HE HERHALEN  
Ikkens afgedrukt, totdat op de J-toets wordt  
drukt.

Van de standaardfuncties ontbreken VAL en  
R\$.

De niet-standaard-functie SPCS(X) - geeft X  
aties - komt evenmin voor, maar deze kan als  
igt eenvoudig worden gedefinieerd.

```
//
DIM SPATIES OF 40
SPATIES(1:40)=" "
PRINT "n",SPATIES(1:5),"<"
//
```

eft als uitvoer

>  
it wordt veroorzaakt door het feit, dat, indien  
t een toekennning aan een substring de waarde  
chts uit minder tekens bestaat dan de substring  
lf, het tekort wordt aangevuld met spaties. Een  
ventueel overschot wordt weggelaten.

```
//
AS="HANDAPPEL"
AS(1:4)="AAR"
PRINT AS
AS(1:4)="DEWNE"
PRINT AS
//
```

eft daarom als uitvoer:

```
AAR APPEL
DENNAPPEL
```

De operator IN heeft een tweeledige werking:  
er vindt controle plaats of een string wel of geen  
substring is van een andere string en de logische  
uitdrukking waarin IN voorkomt krijgt de waarde  
FALSE of TRUE, waarbij TRUE (groter dan 0) de  
positie aangeeft van het eerst karakter in de  
onbezochte string.

Dus:

```
//
AS="HANDAPPEL"
P:="DAP" IN AS
PRINT P
//
heeft als uitvoer: 4.
Onderzoeken we
P:="ZES" IN AS
dan is de uitvoer: 0.
```

Als de te onderzoeken string de nulstring is ("")  
den is de waarde van P gelijk aan LEN(AS)+1 (hier-  
boven dus 10).

### 3. Turtle-grafieken

Zoals opgemerkt heeft C=64/COMAL een aantal  
statements, waarmee een turtle (pen) kan worden  
bediend.

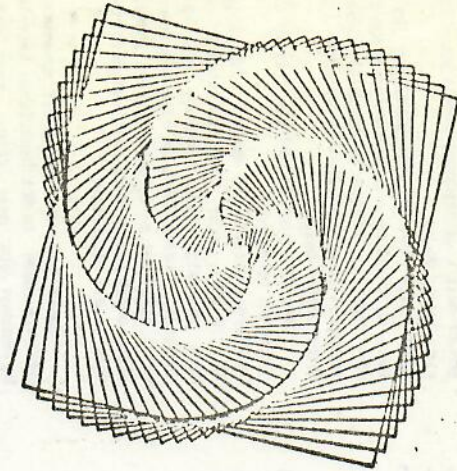
We geven van de mogelijkheden een korte beschrij-  
ving via een enkel programma.

```
0010 //T1
0020 STARTGRAF
0030 SHOW
0040 //
0050 PROC STARTGRAF
0060 SETGRAPHIC 0
0070 MOVE TO 160,100
0080 SETHEADING 0
0090 ENDPROC STARTGRAF
0100 //
0120 PROC SHOW
0130 AT=1
0140 FOR I=1 TO 5 DO
0150 AT=2*A
0160 VIERKANT(A)
0170 ENDFOR I
0180 ENDPROC SHOW
0190 //
0200 PROC VIERKANT(X) CLOSED
0210 FOR I=1 TO 4 DO
0220 FORWARD X
0230 LEFT 90
0240 ENDFOR I
0250 ENDPROC VIERKANT
0260 //
```

Een tweede voorbeeld.

```
0010 //T2
0020 STARTGRAF // zie programma T1
0030 SHOW
0040 //
0040 PROC SHOW2
0050 HOEK=89
0060 LENGTE:=4; DELTA:=2
0070 WHILE LENGTE<200 DO TEKEN(LENGTE)
0080 ENDPROC SHOW2
0090 //
0100 PROC TEKEN(X)
0110 FORWARD X
0120 LENGTE:=LENGTE+DELTA
0130 LEFT HOEK
0140 ENDPROC TEKEN
0150 //
```

De uitvoer van het programma is een figuur als  
onderstaand.



We behandelen waarnodig regel voor regel.

regel 20: aanroep van de procedure STARTGRAF.

regel 60: in plaats van het tekstscherf wordt het  
HR-scherf geïntialiseerd. Turtlegrafie-  
ken zijn nu mogelijk. De pen staat in  
het punt met coördinaten (0,0) - links  
onderhoek.

regel 70: de pen wordt verplaatst naar het punt  
(160,100), zonder te tekenen.

regel 80: de penkop krijgt de richting van de  
positieve x-as. Daarna volgt op

regel 30: aanroep van de procedure SHOW1.

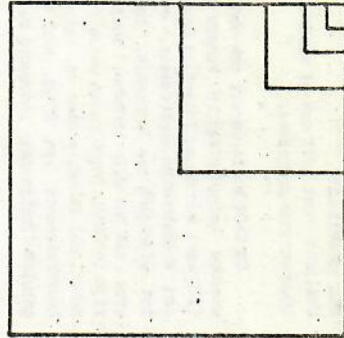
regel 140-180: met de telkens verdubbelde waarde  
van a wordt de procedure VIERKANT 5 keer  
aangeropen.

regel 210-240: 4 keer wordt een samengestelde  
beweging uitgevoerd:

regel 220: FORWARD 2. De pen gaat 2 posities  
(puntes) in de door de penkop aangege-  
ven richting en tekent daarbij een lijn-  
stukje. Daarna volgt:

regel 230: LEFT 90. De penkop draait over een hoek  
van 90 in tegenwijzerrichting. De proce-  
dure VIERKANT(X) tekent dus een vierkant  
waarvan de zijde gelijk is aan X.

De uitvoer van het programma is dus:



Op eenvoudige wijze kan een XY-assenstelsel  
met de oorsprong in het midden van het scherm -  
oorspronkelijke coördinaten (160,100), nieuwe  
coördinaten (0,0) - worden gedefinieerd.

```
PROC NAAR(X,Y)
Xs=160+X*16
Ys=100+Y*15
MOVE TO X,Y
DRAW TO X,Y
ENDPROC NAAR
PROC LIJN'NAAR(X,Y)
Xs=160+X*16
Ys=100+Y*15
DRAW TO X,Y
ENDPROC LIJN'NAAR
```

Beide bovenstaande procedures kunnen in zo'n stijl worden gebruikt. De COMAL-codetacht DRAUWTO X,Y,X tekent een lijn vanuit de huidige positie van de pen naar het punt met de (schemm-)coördinaten (X,Y). De formale parameters zijn coördinaten in het nieuwe assenstelsel.

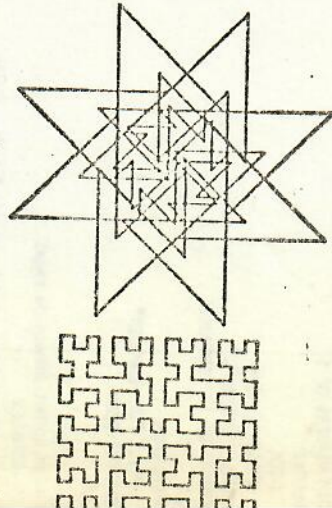
De gebruiken ze in de volgende procedure.

```
PROC CIRKEL(A,B,R)
  NAAR(A+R,B)
  FOR I=0 TO 360 STEP 10 DO
    FI:=I/180*3.14159
    X:=A+R*COS(FI)
    Y:=B+R*SIN(FI)
    LIJN NAAR(X,Y)
  ENDFOR I
ENDPROC CIRKEL
```

Plaatsen we de drie procedures in een programma dan kunnen we na RUN via onderstaande commando's een cirkel tekenen in het nieuwe coördinatenstelsel:

```
SETGRAPHIC 0
CIRKEL(1,-2,4)
```

Door gebruik te maken van recursieve procedures zijn plaatjes als onderstaand mogelijk.



Sprites en andere mogelijkheden

Het maken van zelfgedefinieerde beweegbare figuren (moveable objects, mobs, sprites) vergt, voordat via het op de systeemdisk staand programma SPRITESEICH het ontwerp op het (eventuele) van kleuren voorzien) scherm zelf kan worden samensteld, weinig moeite. Geven onderstaand in een programma echter direct een definitie van een sprite en laten via statements deze sprite over het scherm bewegen.

De gedaante van de sprite is de volgende:



De sprite (MOB) bestaat uit 21 lijnen elk van 3 karakters.

```
//
DIM MOB$ OF 63, DEEL$ OF 3
MOB$:=""
FOR I:=1 TO 21 DO
  MOB$(I)=MOB$(I)+DEEL$
ENDFOR I
MOB$(2)=CHR$(255)+CHR$(0)+CHR$(255)
MOB$(64)=CHR$(0)
// de sprite is nu opgebouwd.
// het laatste karakter bepaalt de soort.
DEFINE 7, MOB$
// de sprite krijgt een nummer (0-47); er
// kunnen zo in een programma 48 sprites
// worden gebruikt.
SETGRAPHIC 0
// initialisering HR-scherm.
IDENTIFY 1,7
// via het aan 7 (het nummer van de sprite)
// gekoppelde getal 1 (0-7 zijn mogelijk)
// kan de sprite worden verplaatst. Er
// kunnen zo maximaal 8 van de 48 beschik-
// bare op het scherm bewegen worden.
SPRITECOLOR 1,4
// de sprite krijgt een kleur.
// nu volgt de bewegings-
HEEN:=TRUE
REPEAT
  IF HEEN THEN BEGIN:=1; EIND:=300; H:=1
  FOR X:=BEGIN TO EIND STEP H DO
    Y:=2/3*X
    SPRITEPOS 1,X,Y
    // de linkerbovenhoek van de sprite
    // komt in punt (X,Y).
  ENDFOR X
  HEEN:=NOT HEEN
  IF NOT HEEN THEN BEGIN:=300; EIND:=1;
  // zo nodig weer terug
  UNTIL FALSE
//
```

Er zijn ook statements en functies waarmee kan worden vastgesteld of op het scherm bewegende sprites in botsing komen met andere op het scherm staande grafische informatie.

Een behandeling daarvan stijgt boven het doel van dit artikel uit. Wellicht moet dit ook al worden gezegd van het bovenstaande programma.

De C-84 bezit verder ook een 6581 "SI0" chip, waarmee geluidseffecten kunnen worden verkregen (SI0 komt van Sound Interface Device). De SI0 is een driestemmige synthesizer, met mogelijkheden tot wijziging van frequentie, toonkleur en volume voor elk der drie stemmen. Voor deze toepassing zijn in COMAL (nog?) geen extra faciliteiten opgenomen. Wel is het echter zo, dat de diverse structuurstatements van COMAL het programmeren van geluidseffecten zeer eenvoudig maakt.

#### Tenslotte.

Opmerkelijk is het, dat het niet via een eenvoudig statement mogelijk is de catalogus van een disk af te drukken op de printer. Een omvangrijk, deels in assembler geschreven, programma is daarvoor nodig.

#### 5. COMAL-80gg en C-84

Nadat een tweetal leden van het bestuur van de COMAL-80gg in januari j.l. (eindejaar) de beschikking hadden gekregen over C84-versie 0.12 en hadden kunnen vaststellen, dat de faciliteiten die COMAL-80 op de C-84 biedt, aan de verwachtingen voldeden, is besloten contact op te nemen met Commodore Nederland om de versie wat gemakkelijker beschikbaar te krijgen dan nu: immers de bestellingen in Denemarken (tot nu toe twee) worden met een vertraging van twee maanden (!) afgehandeld.

Commodore Nederland heeft bereid C84-COMAL-80, versie 0.12, in Nederland in de rij van hun producten op te nemen, daarbij stellend, dat een goed handboek een eerste vereiste was. Tevens verzeevende Commodore aan de COMAL-80gg medewerking om de greep in staat te stellen zo'n handboek samen te stellen en om wat meer bekendheid te geven aan de nieuwe mogelijkheden op de C-84: op de jongstleden gehouden Teachip-COI-dag was versie 0.12 éénook voor het eerst te zien.

Op dit moment is ongeveer de helft van het handboek in eerste fase (nog niet gecorrigeerd) gereed en zijn alle programma's die op de Deense systeemdiskette staan, uit het Deens in het Nederlands vertaald en hier en daar aangepast (in een van de programma's zat zelfs een fout).

Ook de gedichten uit het Deense handboek die in het Nederlandse zullen worden opgenomen, zijn in eerste vertaling gereed, terwijl de rechten voor de vertaling van de Deense programmatuur en het handboek zijn verkregen.

Naar verwachting zal het handboek, dat naast een algemeen inleiding in het programmeren in COMAL-80 zelf, ook een volledig overzicht in de taal zal geven - elk COMAL-woord apart behandeld, voorzien van een toelichting op de werking en van een of meer voorbeeldprogramma's - in augustus a.s. gereed zijn.

Afhankelijk van de belangstelling in Nederland voor 0.12 zal de druk worden verzorgd door Commodore Nederland of door de gebruikersgroep. Hoe de diskettes (samen met boek of afzonderlijk) zal nog nader worden bezien.

Over de kosten in Nederland is op dit moment nog niets te zeggen. De Deense versie komt op Dkr. 350,00 (ongeveer f.l. 105,00), voor twee diskettes en een maget niet volledig. Deens handboek (voor Denen geen probleem, immers COMAL is daar overbekend). Opge-merkt zij, dat de beide diskettes inderdaad nodig zijn, omdat versie 0.12 niet kan worden gekopieerd naar andere diskettes!

Zoals bekend is er op dit moment slechts een COMAL-leerboek in het Nederlands beschikbaar (in Denemarken ook maar drie, maar dat schijnt vrijboven te zijn; hoewel boeken verschijnen er in Nederland over BASIC? : COMAL-80, een programma-taal (twee delen), geschreven door Soren Christensen, vertaald door Preet Potjeffort (Pret van der Poort) en uitgegeven door Foucaboek in Culmborg.

Met de aanpassing van de bij dit boek behorende programmatuur, tot op heden alleen voor de CP/M-versie van COMAL-80, Matonic COMAL-80, beschikbaar, zal binnenkort door de gebruikerstgroep een begin worden gemaakt. Bij de leerboeken zal dan ook een supplement moeten verschijnen in verband met de kleine verschillen tussen de CP/M- en de C84-versie.

Gezien de belangstelling op de Teachip-COI-dag en de telefonische verzoeken om inlichtingen voor opvolgende over de nieuwe COM-versie - de ingebouwde mogelijkheid voor turtle-geachtes sprak zeer aan (zeker bij mensen uit het basisschoolwjs) - zal de COMAL-80gg trachten om op zo kort mogelijke termijn de COMAL-80 versie 0.12 in Nederland beschikbaar te krijgen.

bij:

- Belangstellenden kunnen zich nu reeds melden
- Secretariaat van de COMAL-80gg,
- p.a. Stichting Pasca,
- Bezuldenhoutseweg 225,
- Den Haag, tel. 070-470261.