

TURBO PASCAL 4.0 : beknopte handleiding

Citation for published version (APA):

Koumans, P. W. (1989). *TURBO PASCAL 4.0 : beknopte handleiding*. (TH Eindhoven. Afd. Werktuigbouwkunde, Vakgroep Produktietechnologie : WPB; Vol. WPA0711). Technische Universiteit Eindhoven.

Document status and date:

Gepubliceerd: 01/01/1989

Document Version:

Uitgevers PDF, ook bekend als Version of Record

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

88 446877

TURBO PASCAL 4.0

Beknopte handleiding

Ir. P.W. Koumans

WPA-rapport nr. 0711 april 1989

Alleen voor intern gebruik

TURBO PASCAL

Beknopte handleiding door P.W.Koumans

Rapport WPA nr. 0711 April 1989

1. Starten en hoofdmenu

Om te kunnen starten zijn de volgende programma's op harde schijf of floppy nodig:

TURBO.EXE
TURBO.HLP
TURBO.TPL
GRAPH.TPU
EGAVGA.BGI

en naar behoefte: GOTH.CHR LITT.CHR SANS.CHR TRIP.CHR {letter-tekens}

Start op het juiste directory met het typen: TURBO en <return>.
Op het scherm staat dan verkort weergegeven:

File	Edit	Run	Compile	Edit	Options
Line 1	Col 1	Insert	Indent		C:NONAME.PAS
<div style="border: 1px solid black; padding: 10px; width: fit-content; margin: 0 auto;">Turbo Pascal Version 4.0</div>					
Output					

F1-Help F2-Save F3-Load F5-Zoom F6-Edit F9-Make F10-Main menu

Bovenaan wordt het hoofdmenu gegeven. Het deel met een achtergrond blokje is in te schakelen met de toets <enter>. Men komt op het hoofdmenu met de toets <F10> en kan selekteren met toets <pijl links> of <pijl rechts>. File, Compile en Options hebben nog een aantal te selekteren mogelijkheden, te bereiken met pijl op of pijl neer en de enter toets.

File heeft de mogelijkheden:

- Load Laden van een programma, aan te duiden op DOS manier.
 PAS hoeft niet toegevoegd te worden.
- Pick Neemt een file uit een lijst van 8 vroeger geladen
 files in de Edit selectie.
- New Specificeert dat de file nieuw is. Startend in de Edit

is de startnaam NONAME.PAS

Save Slaat de file op, die in de Edit is.

Write to Schrijft de file onder een nieuwe naam of overschrijft een bestaande file.

Directory Geeft de directory en file set, zoals gewenst.

Change dir Geeft de huidige directory en maakt verandering mogelijk.

OS shell Verlaat TurboPascal tijdelijk en komt in DOS. Terug naar TurboPascal door intikken: exit

Quit Verlaat T.Pascal en komt in DOS op de aktieve directory.

Edit Hiermee kan een programma worden geschreven.

Run Dit kommando doet een programma werken. Als dit programma nog niet gecompileerd is, wordt dit eerst gedaan. Na afloop van het programma komt de aanwijzing: Press any key to return to Turbo Pascal. Soms moet hiervóór eerst de <enter> toets worden gebruikt.

Compile Dit menu is onderverdeeld in een aantal delen, o.a. compile, waarmee een programma kan worden gecompileerd.

Options is ook onderverdeeld in een aantal items, die hier niet worden behandeld.

2. Bijzondere toetsen

F1	Geeft een Help scherm met informatie.
F2	Bewaart de huidige file.
F3	Hiermee kan een file worden geladen.
F5	Maakt het aktieve deel van de scherm groter/kleiner.
Alt F1	Laat het laatste Help scherm zien.
Alt F3	Laat een file nemen om te laden.
Alt F5	Geeft het gesaved beeldscherm.
Alt F9	Compileert het programma.
Alt F10	Geeft weer de versie van het scherm.
Alt C	Hiermee komt het Compile menu.
Alt E	Hiermee komt de Edit mode.
Alt F	Hiermee komt het File menu.
Alt O	Hiermee komt het Options menu.
Alt R	Laat het programma lopen.
Alt X	Verlaat Turbo-Pascal en komt in DOS.
Ctrl F6	Volgende window.

3. Edit mode

Op de bovenzijde van het scherm staat:

Line 1 Col 1 Insert Indent C:NONAME.PAS

Hiermee wordt aangegeven dat de cursor op regel 1 en op colom 1 staat.

EDIT commando's

Cursorbewegingen

Karakter naar links	< <- >
Karakter naar rechts	< -> >
Woord naar links	<Ctrl < <- >>
Woord naar rechts	<Ctrl < -> >>
Regel omhoog	< pijl omhoog >
Regel omlaag	< pijl omlaag >
Scherm terug	< Pg Up >
Scherm verder	< Pg Dn >
Begin van de regel	< Home >

Eind van de regel	< End >
Bovenaan scherm	<Ctrl <Home>>
Onderaan scherm	<Ctrl <End>>
Begin van file	<Ctrl <Pg Up>>
Eind van de file	<Ctrl <Pg Dn>>

Andere commando's

Een regel opschuiven op scherm	<Ctrl <W>>
Een regel naar beneden op scherm	<Ctrl <Z>>
Invoegen aan/uit	< Ins >
Indent (inspringen) aan/uit	<Ctrl <Q><I>>
Nieuwe regel op plaats cursor	<Ctrl <N>>
Karakter wissen links cursor	< Backspace >
Karakter wissen boven cursor	< Del >
Woorddeel rechts en boven c	<Ctrl <T>>
Wissen vanaf cursor	<Ctrl <Q><Y>>
Wissen cursorregel	<Ctrl <Y>>
Markeer begin blok	< F7 >
Markeer einde blok	< F8 >
Cursor naar begin blok	<Ctrl <Q>>
Cursor naar einde blok	<Ctrl <Q><K>>
Verbergen/weergave blok	<Ctrl <K><H>>
Wissen blok	<Ctrl <K><Y>>
Copieert blok bij cursor	<Ctrl <K><C>>
Verplaatsen blok naar cursor	<Ctrl <K><V>>
Schrijft blok met tekst op disk, file naam opgeven	<Ctrl <K><W>>
Leest blok uit disk, filenaam opgeven	<Ctrl <K><R>>
Tab	< Tab >
Tab mode	<Ctrl <Q><T>>
Afbreken van commando	<Ctrl <U>>
Uit Edit naar hoofdmenu	<Ctrl <K><D>> of <Ctrl <K><Q>>
Uit Edit naar hoofdmenu (F,C of O) met resp. Alt F, Alt C, Alt O.	

4. Namen in Pascal te geven aan constanten, types, variabelen of functies worden indentifiers genoemd. Voorwaarden zijn:

- Beginnen met een letter.
- Gevolgd door letters of en cijfers, geen tekens of spaties, wel een scheidingsonderstreep.
- Lengte mag willekeurig, de eerste 63 charakters zijn bepalend.
- Er is geen onderscheid tussen hoofdletters en kleine letters.

Drie soorten namen:

- Gereserveerde namen, zoals CONST, BEGIN, FOR.
- Voorgedefinieerde namen, zoals SQRT, WRITE.
- Namen gedefinieerd door de gebruiker.

5. Constanten. Vier soorten:

- Booleanse: true en valse.
- Gehele getallen, bestaande uit cijfers en + of -, max : +32786, min : - 32786. Benaming : Integer.
- Reële getallen, bestaande uit cijfers, en eventueel uit + - . E.
Voorbeelden: -0.6 ; 5E-8 ; +5E2. Niet .6 schrijven, maar 0.6 .
- Strings. Dit zijn opeenvolgingen van tekens omsloten door een

accent. Bijv. 'dit is een string'. Binnen een string is wel onderscheid tussen hoofdletters en kleine letters.

6. Schema standaard Pascal programma

```

program naam(parameters) ;
label
    merktekens definities ;
const
    constanten definities ;
type
    typen definities ;
var
    variabelen definities ;
procedure of function
    procedure- en/of function beschrijvingen ;
begin
    aktie-beschrijvingen;
end.

```

Bij Turbo-Pascal is de volgorde van de delen niet verplicht. Alleen moet de programma-aanduiding eerst komen en het hoofdprogramma het laatst.

Een programma moet tenminste de woorden `program naam`, `begin` en `end` bevatten.

Kommentaar kan worden overal worden toegevoegd tussen `{....}` of `(*...*)`.

Regels worden niet genummerd, zoals bijv. bij Basic.

Iedere uitdrukking, kommando, deklaratie enz. wordt afgesloten met `;`. Achter `begin` geen `;` gebruiken omdat `begin` en `end` bij elkaar horen. Op een regel mogen meerdere uitdrukkingen staan.

Korte beschrijving van de delen van een programma

a. Het woord `program` moet steeds gevolgd worden door een naam. Daarachter kan tussen haakjes een reeks parameters gegeven worden waarmee het programma naar buiten communiceert. Dit afsluiten met `;`.

b. Het deel van definities moet steeds in de volgorde `label`, `const`, `type`, `var`, `procedure of function` zijn. Het is niet verplicht dat iedere groep aanwezig is.

c. `label` geeft een aantal merktekens waarnaar gesprongen kan worden met het commando `goto` in de vorm van getallen. Bijv. `label 3,18 ;`. Het maximum getal is 9999.

d. Definitie van constanten.

De algemene vormen zijn: `naam = constante;`
of `naam = - constante;`.

Een constante kan een getal zijn, een string of een gedefinieerde naam. Bijv.:

```

const   waar           = true;
        maximum       = 350;
        lettera       = 'a';

```

c. Definities van types.

De algemene vorm is: `naam = type;`.

Type kan verschillende vormen hebben zoals op blz.5 nader wordt uitgelegd.

d. Definities van variabelen.

De algemene vorm is: naam, naam, ...naam : type;

Bijv.: var A,B: boolean;
C,D: char;
K,L: integer;
R : real;

e. procedure of function structuren zijn eigenlijk kleine Pascal-programma's. Ze worden gebruikt om als stukken programma, die dikwijls voorkomen en hebben de mogelijkheid om op verschillende plaatsen in het programma opgeroepen te worden. Zie verder hoofdstuk 10.

7. Variabelen Er zijn vier basistypes:

a. Boolean uitdrukkingen hebben de waarde: True of False. Er worden 4 operatoren gebruikt: not, and, or, xor en de vergelijingsoperatoren: = , < , > , <> , <= , >= .

b. Integer : gehele getallen, positief of negatief. Hierbij 5 operatoren: *, div, mod, +, - . Div is het gehele getal na deling, Mod is de rest bij een deling. Zo is 7 div 3 : 2 en 7 mod 3 : 1.

Turbo-Pascal heeft de integer typen:

MaxInt	-32768	32768	2 bytes
Byte	0	255	1 byte
Shortint	-128	127	1 byte
Integer	-32768	32768	2 bytes
Word	0	65535	2 bytes
Longint	-214783648	214783647	4 bytes

c. Real: reële getallen, positief of negatief, uitgedrukt in getallen met een deel achter de punt (bijv. 3.14) of met exponenten. Turbo-Pascal definieert het type real als zijnde 6 bytes omvang met 11 significante digits en een exponentengebied van $2.9 \cdot 10^{E39}$ tot $1.7 \cdot 10^{E-38}$. De operatoren zijn *, /, en +, -.

Voorgedefinieerde real-functies:

abs (R) {geeft de absolute waarde van R}
 sqr (R) {geeft R*R}
 sqrt (R) {geeft de wortel uit R}
 sin (R) {R in radialen}
 cos (R) {R in radialen}
 arctan (R) {resultaat in radialen}
 pi {type real, precisie afhankelijk van compiler}
 ln (R) {geeft de natuurlijke logaritme van R}
 exp (R) {geeft de exponentfunctie van R}
 int (R) {geeft het hele deel van R afgerond}
 frac (R) {geeft het deel van R achter de punt}
 frac (R) = R - int (R)
 trunc (R) {geeft het hele deel van R (zet deze waarde om tot een waarde van een type van integer: longint)}
 trunc (3.7) : 3
 round (R) {geeft de afronding van R (zet deze waarde om tot een waarde van een type integer: longint)}
 Als R >0 is round (R) = trunc (R+0.5); round (2.7) : 3
 Als R <0 is round (R) = trunc (R-0.5); round (-4.6) : -5

Andere real-functies moet men zelf maken en vastleggen bij het begin van het programma. Voorbeelden zijn in paragraaf 10.2.

d. Char Hiertoe behoort de ASCII karakterset met 128 karakters. De andere 128 waarden kunnen allerlei tekens geven. De aanduiding is `chr(geval)`, bijv `chr(65)`. Functies: `ord('i')` geeft het getal; bijv. `ord('A')` geeft 65.
`chr(K)` geeft het karakter dat overeenkomt met de waarde van de variabele K.
`pred(K)` geeft het karakter met een code 1 kleiner dan K.
`succ(K)` geeft het karakter met een code 1 groter dan K.
Ook hier kunnen vergelijkingsoperatoren gebruikt worden. Zo is `'a' > 'A'` true, immers `97 > 65`.

8. Aktiemogelijkheden.

Berekeningen vinden plaats van links naar rechts met prioriteiten:

1. Een uitdrukking tussen haakjes.
2. De (voorgedefinieerde) functie-oproepen.
3. not.
4. *, /, div, mod, and.
5. +, -, or, xor.
6. =, <>, <, <=, >, >=.

Toewijzing

variabele := uitdrukking; A:=5 betekent dat de variabele A de waarde 5 krijgt. Het type van de uitdrukking moet overeenstemmen met het type variabele. Uitzondering variabele type real en uitdrukking type integer.

Procedure read.

`read (variabele,... variabele)`

De variabelen moeten type real, integer, string of char zijn. De in te tikken gegevens moeten constanten zijn, gescheiden t.o.v. elkaar door tenminste 1 spatie.

`readln (variabele)` Deze lijkt op `read` met als verschil dat alle tekens tot aan het eerste teken van een nieuwe regel overgeslagen moet worden. `readln` zonder meer wordt gebruikt om het programma te laten stoppen en te kunnen starten door `<return>` in te drukken.

Procedure write.

`write (string of uitdrukking,... string of uitdrukking).`

De resultaten worden op een regel geschreven, de cursor blijft op het einde staan.

Een string is een tekst tussen accolades. `Write('tekst')`

De uitdrukkingen behoren tot een van de volgende types:

-type integer: gebruikt een veld ter breedte van de uitdrukking. `Write (12,10)` geeft: 1210. Een groter veld is te verkrijgen met `write(item:breedte)`. `Write(123:5)` geeft ..123

-type real: heeft standaard een veldbreedte van 17. Hierbij wordt de wetenschappelijke notatie gebruikt: het eventuele min-teken, een cijfer, een punt, 10 cijfers, de letter E, het teken en twee cijfers. `Write (23.50)` geeft : = 2.3500000000E+01
Andere notatie met `write (item:veldbreedte:cijfers achter komma)`. Hierbij is veldbreedte een integer uitdrukking. De velden worden vanuit rechts gevuld. `Write (23.50:8:1)` geeft23.5

- type boolean: gebruikt 1 kolom, T voor true en F voor false
- type string : gebruikt de gebruikte lengte als veldgrootte.
- type array of char: gebruikt de gedeklareerde lengte als veldgrootte.

Procedure writeln.

writeln (string of uitdrukking,...string of uitdrukking)

De resultaten worden achter elkaar op een regel geschreven. Hierbij verschuift de cursor naar het begin van de volgende regel.

Zie verder : write.

Repeterende akties.

while uitdrukking do aktie

De uitdrukking moet het type boolean zijn, bij true aktie, na uitvoering wordt opnieuw de while-do lus doorlopen. Dit gaat net zo lang tot de uitdrukking false is. De aktie mag ingebed worden tussen begin ... end. Nesten mag.

Bijv. while n>0 do som := som + getal;

repeat aktie, ...aktie until booleaanse uitdrukking

De lus wordt net zo lang doorlopen totdat de uitdrukking true is. De akties worden tenminste eenmaal uitgevoerd.

for controle variabele := beginwaarde to eindwaarde do aktie
for controle variabele := beginwaarde downto eindwaarde aktie

De controle variabele moet boolean, char of integer zijn, geen real. De begin- en eindwaarde mogen uitdrukkingen zijn.

Nadat de lus is uitgevoerd wordt de beginwaarde verhoogd of verlaagd met de waarde, die volgt op die van de controle variabele. Dus bij een integerwaarde 1 hoger of 1 lager.

Voorwaardelijke akties

if booleaanse uitdrukking then aktie [else aktie]

De aktie na then wordt uitgevoerd bij true, na else bij false.

Niet voor else een ; plaatsen, na een complete aktie een ; gebruiken.

Nesten mag. else mag achterwegen blijven.

case uitdrukking of reeks constanten 1 : aktie;
reeks constanten 2 : aktie;
.
reeks constanten n : aktie
else : aktie

end;

Bijv. case i of

1,2,3 : x:= x+1;

4,5,6 : x:= x-1;

7,8,9 : x:= x/i Als i>9 of <1 dan is de case-aktie ongedefi-

nieerd.

Een case-label mag ook een enkel getal zijn, maar nooit een variabele.

Bij het case of commando is een else-clausule mogelijk.

Onvoorwaardelijke sprong

```
goto merkpunt;
merkpunt : aktie;
```

Deze merkpunten zijn vastgelegd in de label. Let op bij TURBO-PASCAL mag niet in een gestructueerd statement gesprongen worden, dat zijn samengestelde statements, voorwaardelijke statements of lussen. Men moet binnen een blok blijven (tussen begin en end).

9. Soorten typesString data type

Een string is een reeks van karakters. Het aantal te gebruiken karakters wordt gedefinieerd. Het maximum bedraagt 255.

```
Bijv. var name : string[30]
```

Het string type kan worden geschreven met write en worden ingelezen met read. Men hoeft niet de gehele ruimte van de string te vullen. Met Length(stringnaam) wordt de gebruikte lengte weergegeven.

De typen integer of real kunnen worden omgezet in een stringtype met een Str procedure :

```
Str(x[:veldbreedte[:aantal decimalen]],var s:string)
```

```
bijv.: function realstr(i:real):string
      var s: string[10]
      begin
        Str(i:6:4,s);
        realstr := s;
      end;
```

Array-typen

Een array is een reeks gegevens van hetzelfde type.

```
naam = array[ type 1 ] of type 2
```

Type 1 geeft het aantal elementen en soms de meervoudigheid van de array. Toegelaten zijn de typen boolean, char, scalair en interval. Type 2 geeft het type van de elementen. Hiervoor zijn alle typen toegestaan.

```
Bijv. tabel = array[1..10] of integer
```

```
Bijv. const n= 8; type woord = array[1..n] of char
```

```
Bijv. type kleur = (rood, oranje, geel, groen, blauw);
```

```
verf = array[ kleur ] of boolean
```

Selektie kan met de vorm : variabele [uitdrukking].

```
Bijv. var a: woord; a[5] is het vijfde element.
```

Multidimensionele array-types

```
Bijv. type vector = array[1..10] of integer;
```

```
matrix = array[1..10] of vector; {dit is een twee-
```

dimensionale array}.

```
Ook kan type matrix = array[1..10] of array[1..10]
```

```
Korter is: type matrix = array[1..10,1..10] of char
```

```
Bijv. type lijn = array[1..10] of char;
```

```
blad = array[1..50] of lijn;
```

```
boek = array[1..200] of blad;
```

Moeilijker te hanteren is:

```
type boek = array[1..200,1..50,1..100] of char
```

Selektie kan hieruit met var d: boek

1e char, 1e lijn, 1e bladzijde is te vinden met d[1][1][1] of verkort met d[1,1,1] .

Array typen kunnen worden geschreven met write, echter niet rechtstreeks worden ingelezen met read.

Een methode hiervoor is bijv.:

```

    var A : array[1..10] of char;
        x : char;
        i : integer;
begin
    for i := 1 to 10 do
    begin
        if EOLN then x := ' '
            else read(x);
        A[i] := x;
    end;
    writeln(A)
end.
```

Met dit programmaatje kan de array worden ook worden gebruikt voor teksten met minder dan 10 karakters. Teksten met meer dan 10 karakters wordt afgebroken na 10 karakters.

Er zijn nog andere types, zoals Record-types, Set-types ,die echter hier niet zullen worden behandeld.

10. Procedures en funkties

Procedures zijn zelfstandige programmadelen, die vanuit het hoofdprogramma kunnen worden opgeroepen. De structuur van zo'n deel is gelijk aan het hoofdprogramma. Tenminste moet het bevatten het hoofd: procedure <naam>, en een blok met begin en end; .

10.1 Parameters

Globale parameters

Dit zijn parameters, die voor het gehele programma gelden. Ze worden gedeclareerd voor het begin van het hoofdprogramma.

Lokale parameters

Deze parameters worden gebruikt binnen een procedure of een funktie en hebben alleen daar betekenis. Ze mogen in principe dezelfde naam voeren als een globale parameter. Ze moeten aan het begin van de procedure of funktie worden gedeclareerd.

Formele parameters

Dit zijn parameters, die binnen een procedure of funktie gelden en die worden aangeroepen met aktuele parameters vanuit het programma buiten de procedure of funktie en de waarde ervan krijgen. De deklaratie geschiedt in de hoofdaanduiding met naam.

In principe als volgt: procedure <naam> (par1, par2 :type).

In het programma wordt een procedure aangeroepen met:

<procedure-naam>. Een procedure met bijv. twee formele parameters met: <procedure-naam>(aktuele_parameter1,aktuele_parameter2).

Na het doorlopen van de procedure behouden de aktuele parameters hun waarde.

Variabele parameters

Deze worden ook gedeclareerd in de hoofdaanduiding met naam vooraf gaande met het woord var . dus bijv. als procedure <naam> (var par1, par 2:type). Deze parameters starten net als de formele parameters met de waarde van de aangeroepen aktuele parameters. Na doorlopen van de

procedure krijgen de aktuele parameters de waarde van variabele parameters.

10.2 Funkties

Bij een funktie wordt als output slechts één waarde afgegeven en wel via de funktie-naam. De verdere opbouw is als de procedure.

Schematisch is de opbouw:

```
function <naam> (parameter1, parameter2 :type):type;
  deklaratie deel;
  begin
                                end;
```

Het aantal voorgedefinieerde funkties is gering (zie 7.c). Met function kunnen gemakkelijk gewenste wiskundige uitdrukkingen worden vastgelegd en worden opgeroepen vanuit het programma. Voorbeelden:

```
function arcsine (getal:real):real;   { arcsine(parameter) }
  begin
    if getal = 1 then arcsine := pi/2
    else   if getal = -1 then arcsine := -pi/2
    else arcsine := arctan(getal/sqrt(1-getal*getal));
  end;
```

```
function arccos (getal:real):real;   { arccos(parameter) }
  begin
    if getal = 0 then arccos := pi/2 else
    arccos := arctan(sqrt(1-getal*getal)/getal);
  end;
```

```
function tan(hoek:real):real;       { tan(hoek in radialen) }
  begin
    tan := sin(hoek)/cos(hoek);
  end;
```

```
function macht (getal,exponent:real):real; { machtsverheffen }
  begin
    macht := exp(exponent*ln(getal));
  end;
```

```
function log10(getal:real):real;
  begin
    log10 := ln(getal)/ln(10.0);
  end;
```

11. Weergave op het scherm

Er zijn een aantal kommando's waarmee de weergave op het beeldscherm kunnen worden geregeld. Enkele hiervan zijn:

```
ClrScr           { maakt beeldscherm of aktieve window schoon }

GotoXY(x,y)     { (type:byte), verplaatst cursor naar x,y waarbij x de
                  kolom is en y de rij. Linkerbovenhoek is (1,1) }

KeyPressed      { geeft true als een toets wordt ingedrukt }

Window(x1,y1,x2,y2) { (type:byte) geeft een window met x1,y1 als
                      linker bovenhoek en x2,y2 als rechter onderkant.
                      Default waarde (1,1,80,25) }
```



```

(x,y) als middelpunt en XRadius en YRadius als
resp. horizontale en verticale as.}
OutText(Textstring) { (type: string, zie string voor omzetting van real
of integer in type string) plaatst een string
tekens beginnend bij de grafische cursorpositie.
De string wordt afgebroken door de rand van het
beeldscherm of window. De cursor wordt over de
stringlengte verplaatst. }

OutTextXY(x,y,Textstring) { Als OutText, beginnend bij punt (x,y). De
cursor wordt niet verplaatst. }

SetLineStyle(Line Style, Pattern, Thickness) { (typen: word) }
SolidLn      = 0;   DottedLn      = 1;
CenterLn     = 2;   DashedLn      = 3;
UserBitLn    = 4;   hierbij Pattern opgeven als
                  BitPattern
NormWith     = 1;   ThickWith     = 3;

SetTextStyle(Font,Direction,CharSize) { (typen resp. word, word,
CharSizeType = 1..10) }
DefaultFont  = 0;   TriplexFont   = 1;
SmallFont    = 2;   SansSerifFont = 3;
GothicFont   = 4;
HorizDir     = 0;   (links naar rechts)
VertDir      = 1;   (van onderen naar boven)
NormSize     = 1;

SetTextJustify(Hor,Vert) { (typen: word), hiermee wordt de tekst op
de aangegeven wijze weergegeven t.o.v. de
cursor of aangegeven punt }
Horizontale instelling:
LeftText     = 0;   CenterText    = 1;   RightText   = 2;
Verticale instelling:
BottomText   = 0;   CenterText    = 1;   TopText      = 2;

GetMaxX      {(type word), geeft de maximum bereikbare waarde van x
voor het toegepaste beeldscherm }

GetMaxY      { analoog als GetMaxX }
Voorbeeld: Rectangle(0,0,GetMaxX,GetMaxY) tekent een
rechthoek, die het gehele scherm vult.

CloseGraph   { zet de grafische mode uit }

```

14. Literatuur

14.1. Boon, K.L.

Pascal voor iedereen
Kluwer Technische Boeken, 5e oplage, 1984

14.2 Borland International

Turbo Pascal 4.0 Owner's Handbook
Borland International, Scotts Valley, U.S.A., 1987

14.3 Cremers, M.K.P.

Programmeren in Pascal

Educaboek, Culemborg, 1984

- 14.4 Duntemann, J.
Turbo Pascal
Academie Service, 2e druk, 1986
- 14.5 Huybrechts, M.
Pascal
de Sikkel, 1981