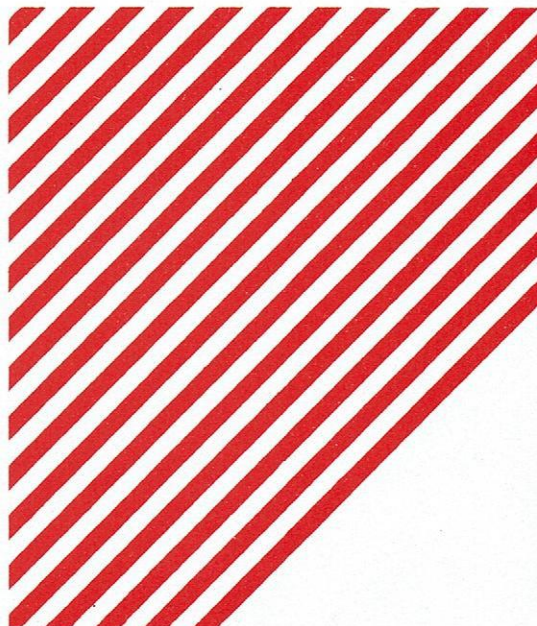
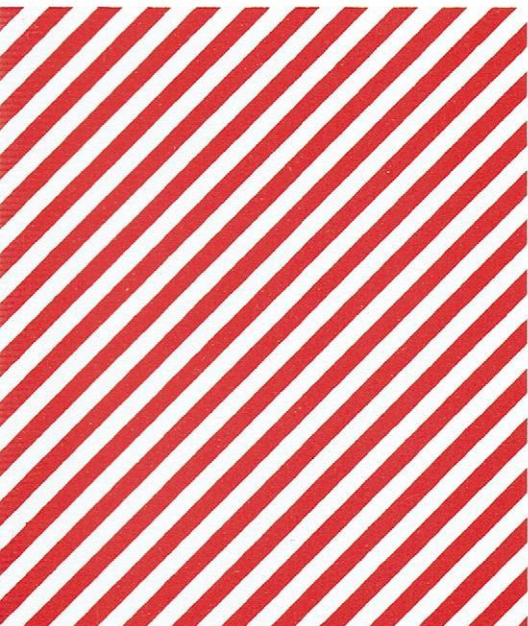




CPI-graf

Peter Ferdinand

En samling pascal procedurer
som gør det nemt at lave vinduer,
farver og grafik på skærmen.



CPI-graf

Peter Ferdinand

En samling pascal procedurer
som gør det nemt at lave vinduer,
farver og grafik på skærmen.

Forord

Anvendelsen af CPI-graf er fri. Det betyder at man frit kan inkludere hele programmet eller dele deraf i sine egne programmer. Man anmodes om at skrive følgende på en fremtrædende plads på disketten eller i brugervejledningen, som følger programmet:

CPI-graf fra Dansk Center for Pædagogik og Informatik er benyttet som en del af dette program.

CPI-graf er beskyttet af sædvanlige regler for ophavsrettigheder. Man kan derfor anvende det på samme måde, som man anvender en bog. Det betyder, at man kan udlåne CPI-graf til andre, ligesom man kan arbejde med det på andre datamater, blot man sikrer sig at det ikke benyttes samtidigt på to eller flere datamater. Man har også ret til at lave kopier af CPI-graf, men kun med det formål, at sikre sig imod tab af originaldisketten.

CPI-graf er udformet på en sådan måde at procedurenavne og procedurekald stort set svarer til de konventioner, som gælder for anvendelsen af grafik i Comal. Det betyder at man relativt enkelt vil være i stand til at benytte CPI-graf, hvis man kender til grafikken i Comal.

Denne brugervejledning giver dels nogle eksempler på anvendelsen af CPI-graf og dels en mere detaljeret beskrivelse af de procedurer, som brugeren oftest kommer i kontakt med. En række procedurer er kun beskrevet kortfattet. Hvis man ønsker at bruge disse henvises til kildeteksten.

Dansk Center for Pædagogik og Informatik har gjort hvad der er muligt for at programmet er fejlfrit. Hvis der imod forventning alligevel findes fejl, bedes de rapporteret med dokumentation af, hvor fejlen findes og under hvilke betingelser den er opstået.

Indhold

1. Indledning

1.1 Programmer på distributionsdisketten

2. Opbygning af skærbilleder

2.1 Det grafiske skærbillede - grafiktilstand

2.2 Tekstskærbilledet - teksttilstand

2.3 Vinduer på skærmen

2.4 Skalering af vinduer

2.5 Tegning af grafer

3. Oversigt over procedurer og konstanter

3.1 Globale konstanter og variable

3.2 Hjelpeprocedurer

3.3 Procedurer på skærmniveau

3.4 Procedurer på vinduesniveau

4. Appendix A Tegnsæt på Partner/Piccoline

1 Indledning

CPI-graf er en samling procedurer skrevet i Poly-pascal. De søger at løse et problem, som alle der har forsøgt at udvikle seriøse programmer, er stødt på. Hvis man vil udnytte moderne datamaters grafiske muligheder, oplever man at der er så store uoverensstemmelser mellem de forskellige typer af datamater, at flytning af programmer er særdeles tidsrøvende og vanskeligt. Problemet er, at der ikke findes en officiel eller generelt accepteret standard for grafisk udskrift til skærm eller printer. Hver maskinproducent har lavet egne regler, og kun få har tilsyneladende været interesserede i fælles regler for dette område. De forskellige forsøg der har været på at frembringe en standard, har desværre ikke været særlig succesfulde. CPI har derfor besluttet at udvikle grafiske procedurer, som kan anvendes på en del af de danske skoledatamater. Disse procedurer giver brugeren en ensartet grænseflade, samtidig med at procedurerne er rimeligt effektive, dvs at de arbejder hurtigt og præcist på de forskellige typer af datamater. CPI har foreløbigt installeret procedurerne på følgende typer af datamater:

IBM PC med Graphics Color Adapter (640 x 200 punkter)

IBM PC med Enhanced Graphics Adapter (640 x 300 punkter)

Olivetti M24 (640 x 400 punkter)

Regnecentralens Partner (720 x 352 punkter)

Regnecentralens Piccoline (560 x 256 punkter)

Procedurerne vil relativt simpelt kunne installeres på andre datamater. Det eneste krav er, at man fra Poly-pascal har direkte adgang til det grafiske skærbillede. I alle tilfælde udnyttes det grafiske skærbilledes største opløsning, og det er derfor ikke muligt med CPI-graf at anvende forskellige farver i de enkelte punkter, som billedet består af.

Udover de grafiske procedurer findes der en række procedurer til udskrift og indlæsning af tekster. Disse procedurer arbejder fuldstændigt ens i såvel tekst- som grafiktilstand. Der er fuld understøttelse af vinduer på skærmen, hvilket betyder at brugeren kan

arbejde med enten grafiske vinduer eller med vinduer som indeholder tekster på en helt ensartet måde.

CPI-graf indsættes i et Poly-pascal program ved at indføje følgende sætning i starten af programmet:

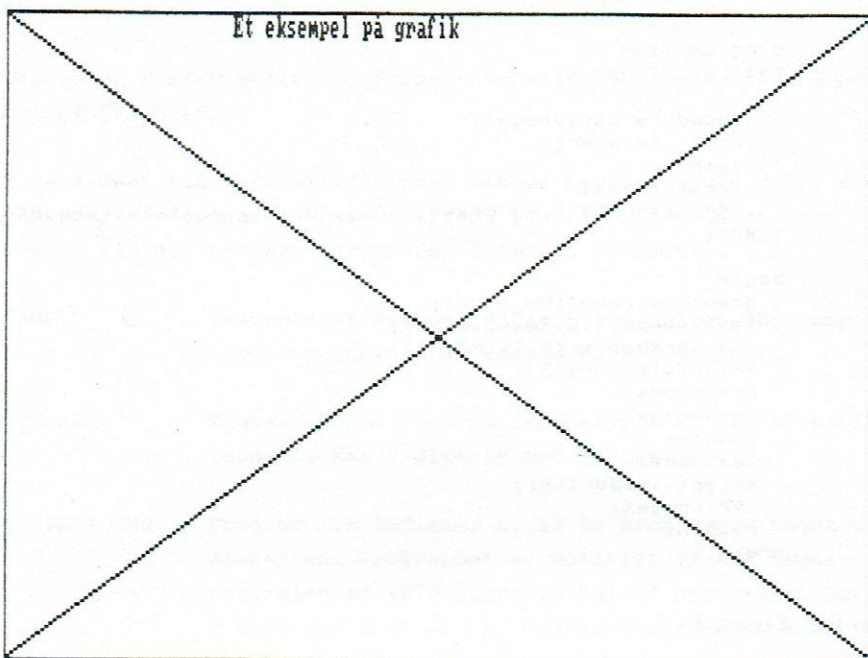
```
(*§I cpi-graf *)
```

I figur 1 er der vist et eksempel på, hvorledes et meget simpelt program konstrueres. Programmet kalder først proceduren GRAPHIC-SCREEN med en angivelse af, hvilken type skærm man arbejder med. I dette eksempel en IBM PC skærm med en opløsning på 640 x 200 punkter. Hele skærmen anvendes som et stort vindue og dette vindue skaleres med sætningen **window** til 0 - 100 og 0 - 100 for henholdsvis x- og y-aksen. Der tegnes en ramme rundt om vinduet med sætningen **drawframe**. De følgende fire sætninger tegner to streger fra hjørne til hjørne. Når der trykkes på retur-tasten vendes der tilbage til teksttilstand igen.

```
program figur1;
(*§I cpi-graf*)
const
  retur=@13;
var
  ch: char;
begin
  graphicscreen(ibm_high);
  graphmode;
  window(1,0,100,0,100);
  selectwindow(1);
  drawframe;
  moveto(0,0);
  drawto(100,100);
  moveto(0,100);
  drawto(100,0);
  drawstr(20,0,'Et eksempel på grafik');
  repeat read(kbd,ch) until ch=retur;
  textmode;
end.
```

Figur 1

Et PolyPascal program som viser grundprincipperne bag anvendelsen af CPI-graf.



Figur 2

Resultatet af kørslen med programmet som er vist i figur 1.

Det er muligt at anvende en række af procedurerne i både grafik- og teksttilstand. Programmet i figur 3 viser hvorledes man kan lave et vindue som består af linier, hvori man kan skrive en vilkårlig tekst.

Et eksempel på et program som laver et tekstvindue på skærmen i både grafik- og i teksttilstand og udskriver en tekst i det.

Der defineres et vindue på skærmen hvor øverste venstre hjørne er i kolonne 20 og linie 8 og nederste højre hjørne er i kolonne 45 og linie 18. Denne definition sker med sætningen **textviewport**. Vinduet navngives med sætningen **viewporthead** og vinduet indrammes med sætningen **writeframe**. Her udskrives navnet på vinduet ligeledes.

```

program figur3;
(*#i c:cpi-graf*)

  procedure skrivnoget;
  var i: integer;
  begin
    writeframe;
    for i:=0 to 8 do drawstr(0,i,'Grafik og tekstvindue');
  end;

begin
  graphicscreen(ibm_high);
  textviewport(2,20,45,8,18);
  viewporthead(2,'Vindue');
  selectviewport(2);
  graphmode;
  skrivnoget;
  readln;
  textmode;
  selectviewport(2);
  skrivnoget;
  readln;
end.

```

Figur 3

1.1 Programmer på distributionsdisketten

På distributionsdisketten ligger en del filer.

CPI-GRAF.PAS Dette er de egentlige procedurer skrevet i Poly-pascal.

8x8FONT.CHR Et tegnsæt i 8 x 8 punkter som bruges til tekster i grafikskærm-billedet. Tegnsættet svarer til det tegnsæt som findes til IBM PC. Dette tegnsæt skal altid være tilstede på den diskette, hvor CPI-graf anvendes. Det kaldes fra proceduren **GRAPHICSCREEN**, som automatisk indlæser det.

KIK.HER En tekstfil som indeholder beskrivelsen af de sidste ændringer.

DEMO.PAS Demonstrationsprogram

Derudover findes nogle programmer som viser eksempler på anvendelsen af CPI-graf.

I versionen til Piccoline/Partner findes desuden nogle filer som medfører, at datamaten kommer til at udskrive et tegnsæt, som svarer til det tegnsæt der findes i IBM PC datamater.

IBMPIC Tegnsætsdefinition af Piccoline til et IBM kompatibelt tegnsæt. Kan redigeres med RcFont.

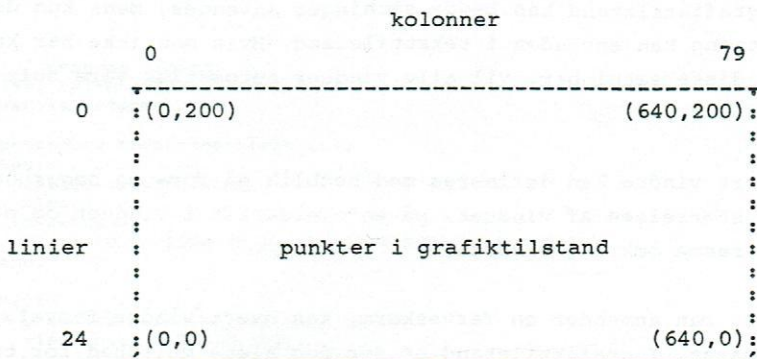
IBMPAR Tegnsætsdefinition af Partner til et IBM kompatibelt tegnsæt. Kan redigeres med RcFont.

CHARSET.CMD Program som indlæser et af de ovennævnte tegnsæt i datamaten. Programmet er udviklet af A/S Regnecentralen af 1979 og er en del af programmet RcFont. I appendix A er der en nøjere beskrivelse af, hvorledes programmet anvendes.

2 Opbygningen af skærbilleder

2.1 Det grafiske skærbillede - grafiktilstand

Det grafiske skærbillede er opbygget som en række punkter - pixels - der enten kan slukkes eller tændes. CPI-graf udnytter skærmens højeste opløsning og kan derfor alene repræsentere to "farver", sort eller hvid, svarende til om et punkt er slukket eller tændt. På en IBM PC med en ColorGraphics Adapter findes der horisontalt 640 punkter, dette kaldes fremover for x-aksen i skærbilledet. Der findes vertikalt 200 punkter, og dette kaldes for y-aksen. Koordinaterne kan vises som i figur 4. Hver gang der defineres et vindue på skærmen med sætningen **viewport** skal der refereres til denne opbygning af skærmen. Man skal dog være opmærksom på at andre datamater og skærme kan være opbygget anderledes. Der er derfor indlagt to variable som angiver skærmens maximale størrelse. De hedder **scr_maxx** og **scr_maxy**. Denne oplysning kan benyttes, hvis man fx ønsker at definere et vindue i øverste venstre halvdel af skærmen **viewport (1,0,scr_maxx div 2, 0, scr_maxy div 2)**. I grafisk tilstand er det muligt med CPI-graf at skrive tekst linievis, ligesom det er muligt at tegne teksten. Hvis teksten skrives med **drawstr** følges linierne, som angivet i figur 4, hvis teksten tegnes med **plottext** eller **plotchar** følges punkterne på skærmen. Man skal dog være opmærksom på at ikke alle datamater kan vise det samme antal tegn og linier på skærmen i grafisk tilstand. Tegnmatricen i grafik er 8 x 8 punkter, hvilket betyder, at det maximale antal tegn der kan vises er $((scr_maxx+1) \text{ div } 8)$. På en Partner er det maximale antal tegn pr linie derfor 90 tegn nemlig $720 \text{ div } 8$. Der er to variable som til stadighed viser antallet af linier og kolonner på skærmen i både grafik- og teksttilstand. Antal af kolonner ligger i variabelen **txt_maxx** og antal af linier i variabelen **txt_maxy**.



Figur 4

Opbygningen af det grafiske skærbillede på en IBM PC. Der vises både punktopbygningen for tegneoperationerne og linieopbygningen for skriveoperationerne.

2.2 Tekstskærbilledet - teksttilstand

Tekstskærbillede er opbygget af tegn og linier på skærmen. Hvert tegn kan vises i nogle forgrundsfarver som er tegnets farve, og nogle baggrundsfarver. De fleste datamater har idag mulighed for at vise 25 linier på hver 80 tegn på skærmen. Disse linier nummereres oppefra med fortløbende numre fra 0 og 24. Tegnene nummereres fra venstre med fortløbende numre fra 0 og 79. Da det ikke kan lade sig gøre at lave grafik i teksttilstanden, vil kun nogle få procedurer være relevante. I procedure-oversigten er disse mærket med 'teksttilstand'.

2.3 Vinduer på skærmen

Et vindue er et område på skærmen, som defineres særskilt med sætningerne **viewport** eller **textviewport**.

I grafiktilstand kan begge sætninger anvendes, mens kun den sidste sætning kan anvendes i teksttilstand. Hvis man ikke har kaldt nogle af disse sætninger, vil alle vinduer automatisk være hele skærmbilledet.

Hvert vindue kan defineres med henblik på for- og baggrundsfarven, på størrelsen af vinduet, på en overskrift i vinduet og på at tegne en ramme omkring vinduet.

Hvis man anvender en farveskærm, kan hvert vindue farvelægges separat. I grafiktilstand er der dog alene mulighed for to farver, hvid og sort. I teksttilstand kan der anvendes 8 forskellige farver både som for- og baggrundsfarve. I teksttilstand gøres det med sætningerne `viewportcolor`, `textviewportcolor` fx `textviewportcolor (1,yellowtxt, redtxt)`, hvis forgrundsfarven i vindue 1 sættes til gul og baggrundsfarven til rød.

Der laves en overskrift øverst i vinduet, hvis der er defineret en sådan med sætningen `viewportheader` fx `viewportheader (1,'vindue nr 1')`. Her indsættes en overskrift i vindue 1. Overskriften indskrives med sætningen `writeframe` i enten tekst- eller grafiktilstand eller med `drawframe` i grafiktilstand alene.

Når alle vinduer er defineret på skærmen, kan de gøres aktive ét efter ét. Der kan kun være ét vindue aktivt ad gangen. Alle følgende operationer er derfor gyldige for det aktive vindue. Det gøres med sætningen `selectviewport`, indtil et andet vindue bliver gjort aktivt med samme sætning. Figur 5 viser et eksempel på anvendelsen af to vinduer.

```

program figur5;
(**! cpi-graf*)
var i: integer;

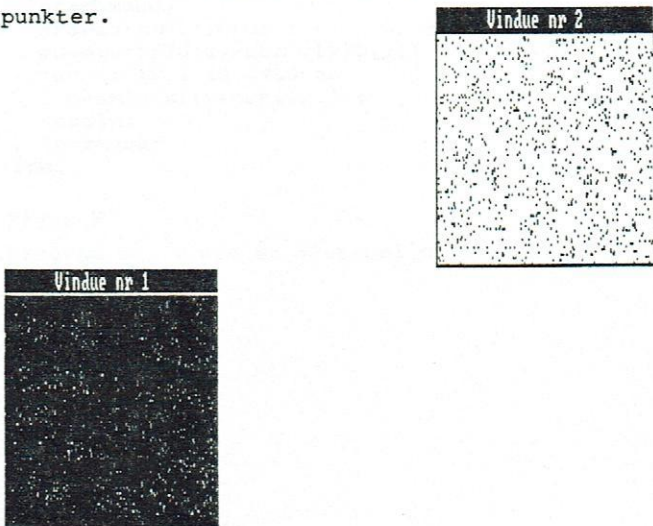
procedure rndskr(no: integer);
begin
  selectviewport(no);
  clear;
  drawframe;
  for i:=0 to 1000 do plot(random*2000,random*2000);
end;

begin
  graphicscreen(ibm_high);
  viewport(1,0,200,0,100);
  viewportcolor(1,black,white);
  viewport(2,400,600,100,200);
  viewportheader(1,'Vindue nr 1');
  viewportheader(2,'Vindue nr 2');
  graphmode;
  window(1,0,2000,0,2000);
  selectwindow(1);
  rndskr(1);
  rndskr(2);
  readln;
  textmode;
end.

```

Figur 5

Et program som udskriver to vinduer og fylder dem med tilfældige punkter.



Figur 6

En udskrift af programmet fra figur 5.

2.4 Skalering af vinduer

Hvert enkelt vindue på skærmen kan skaleres uafhængigt. Det gøres med sætningen **window**. Skaleringen vælges for alle følgende operationer med sætningen **selectwindow**. I figur 5 er der vist hvorledes hvert af de to vinduer skaleres med 0 til 2000 for både x- og y-aksen. Hvis man ikke har anvendt sætningen **window** vil skaleringen af det aktuelle vindue være skærmens størrelse.

2.5 Tegning af grafer

En procedure **drawaxis** er beregnet til at sætte tal og betegnelser på x og y-akserne og lave et "gitter" i tegningen. Proceduren arbejder altid i det aktive vindue. I figur 7 er der vist et eksempel på et lille program som tegner en graf over udviklingen i nogle data fra nationalregnskabsstatistikken. I figur 8 er der vist, hvorledes resultatet ser ud, når det udskrives på printer.

```

program figur7;
(*$! cpi-graf *)
var
  xykoor: array (.1970..1980.) of real;
  x: integer;

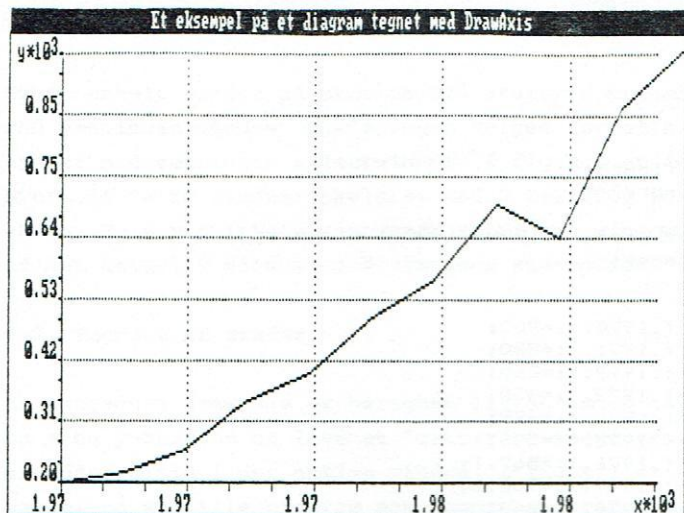
begin
  xykoor(.1970.):=200;
  xykoor(.1971.):=220;
  xykoor(.1972.):=260;
  xykoor(.1973.):=350;
  xykoor(.1974.):=399;
  xykoor(.1975.):=500;
  xykoor(.1976.):=567.1;
  xykoor(.1977.):=702.6;
  xykoor(.1978.):=643;
  xykoor(.1979.):=876;
  xykoor(.1980.):=978;

  graphicscreen(ibm_high);
  viewportheader(1,'Et eksempel på et diagram tegnet med DrawAxis');
  window(1,1970,1980,200,1000);
  selectwindow(1);
  selectviewport(1);
  graphmode;
  drawaxis(7,7,true);
  moveto(1970,xykoor(.1970.));
  for x:=1971 to 1980 do
    drawto(x,xykoor(.x.));
  readln;
  textmode;
end.

```

Figur 7

Program der viser et eksempel på tegning af en graf.



Figur 8

Resultatet af kørslen med programmet som er vist i figur 7.

```

program figur9;
(**I cpi-graf *)
var
  ch,ch1:char;
  xykoor: array (.1970..1980.) of real;
  i,x,oldx: integer;
  fourx,foury: integer;
  wind2, wind3: wnd_type;

procedure column(x: Integer; xvalue, yvalue: real; color: byte);
var
  wndvalue: integer;
begin
  with wnd_ do begin
    wnd_:=wind3;
    selectviewport(1);
    drawstr(0,3,'x-akse:');drawreal(9,3,xvalue,10,2);
    drawstr(0,4,'y-akse:');drawreal(9,4,yvalue,10,2);
    wnd_:=wind3;
    wndvalue:=_wld_towindowy(yvalue);
    scr_fill(x+x1,y1+1,x+x1+24,wndvalue+y1+1,color);
    wnd_:=wind2;
    wndvalue:=_wld_towindowx(xvalue);
    wnd_.pencolor:=transp;
    wnd_drawline(wndvalue,0,wndvalue,maxy);
    wnd_drawline(wndvalue+1,0,wndvalue+1,maxy);
    wnd_drawline(wndvalue+2,0,wndvalue+2,maxy);
  end;
end;

```



```

begin
xykoo(.1970.):=200;
xykoo(.1971.):=220;
xykoo(.1972.):=260;
xykoo(.1973.):=350;
xykoo(.1974.):=399;
xykoo(.1975.):=500;
xykoo(.1976.):=567.1;
xykoo(.1977.):=702.6;
xykoo(.1978.):=643;
xykoo(.1979.):=876;
xykoo(.1980.):=978;

graphicscreen(ibm_high);
fourx:=scr_maxx div 4;
foury:=scr_maxy div 4;
viewporthead(1,'Et eksempel på en Graf');
window(1,1970,1980,200,1000);
viewport(2,2*fourx,scr_maxx,2*foury,scr_maxy);
viewport(3,0,2*fourx-8,8,3*foury-2);
viewportcolor(2,black,white);
viewportcolor(3,white,black);

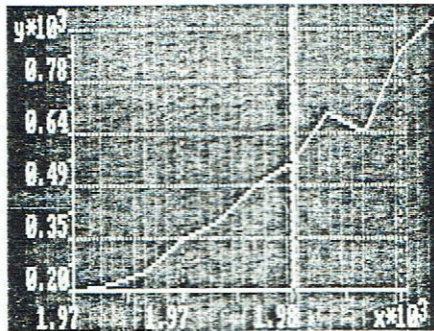
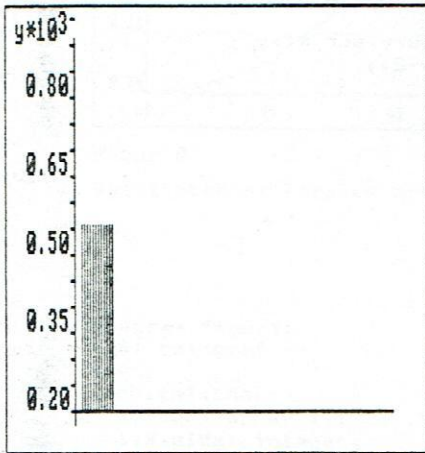
selectwindow(1);
selectviewport(2);
graphmode;
drawaxis(8,8,true);
wind2:=wnd_;
moveto(1970,xykoo(.1970.));
for x:=1971 to 1980 do
  drawto(x,xykoo(.x.));
selectviewport(3);
drawaxis(0,7,false);
wind3:=wnd_;
x:=1970;i:=8;
repeat;
  colomn(8,x,xykoo(.x.),%aa);
  oldx:=x;
  read(kbd,ch);
  if keypress then begin
    read(kbd,ch1);
    case ch1 of
      @75: if (x-1)>=1970 then x:=x-1;
      @77: if (x+1)<=1980 then x:=x+1;
    end;
    colomn(8,oldx,xykoo(.oldx.),0);
  end;
until ch=@13;
selectviewport(1);
wnd_drawtext(0,0,' - tryk <p> hvis du ønsker udskrift på printer');
read(kbd,ch);
if ch in (.'P','p'.) then scr_dump(ibm_graph);
textmode;
end.

```

Figur 9

Et program som viser, hvorledes man kan lave histogrammer og grafer.

x-akse: 1976.00
y-akse: 567.10



- tryk (p) hvis du ønsker udskrift på printer

Figur 10

Resultatet af kørslen med programmet som er vist i figur 9.

3 Oversigt over procedurer og konstanter.

CPI-graf arbejder på tre niveauer, som nærmere defineres nedenfor.

Skærmniveau:

Det er helt fundamentale procedurer, som arbejder i de absolutte skærmkoordinater. Procedurenavnene starter alle med `Scr_`
fx: `Scr_Plot`, `Scr_Drawline`.

Vinduesniveau:

Her arbejdes der i det til enhver tid aktive vindue, men stadig på skærmkoordinater, dog således at de starter fra vinduets nederste venstre hjørne med (0,0). Hvis der angives punkter udenfor vinduet vil disse ikke blive tegnet på skærmen. I alle procedurer af den art starter navnet med betegnelsen `Wnd_`
fx: `Wnd_Plot`, `Wnd_Drawline`.

Globale niveau (Brugerniveau):

Her arbejdes der ligeledes i det til enhver tid aktive vindue, men med skalerede koordinater. Koordinaterne transformeres ved hjælp af de variable som er angivet med sætningen `window`. Dette niveau kan også kaldes brugerniveauet, da det er disse procedurer, som brugeren oftest vil anvende. Det skal bemærkes, at de skaleringer som skal foregå er meget maskinkrævende og derfor nedsætter hastigheden af det udskrivne ganske betydeligt.

Mange procedurer vil være definerede på alle tre niveauer.

Dette afsnit vil nøjere gennemgå de procedurer, som findes i CPI-graf. Der er ikke tale om en fuldstændig programdokumentation, men blot en relativ kort beskrivelse af de vigtigste faciliteter, således at programmøren kan anvende programmet og ændre i forudsætningerne.

Ikke alle procedurer er defineret i detaljer. Det er kun tilfældet for de, som umiddelbart skal anvendes af brugeren. Dokumentationen er organiseret i fire afsnit.

- 1) **Maskinspecifikke procedurer** som brugeren normalt ikke vil anvende, men som er relevante, hvis CPI-graf skal flyttes til en anden datamat.
- 2) Procedurer på **skærmniveau**, hvor hele skærmen anvendes til udskrift, dvs at der arbejdes i absolutte koordinater.
- 3) Procedurer på **vinduesniveau** hvor der kun arbejdes i det aktive vindue, men stadig i absolutte skærmkoordinater.
- 4) Procedurer på **brugerniveau**. Der anvendes skalerede koordinater. Disse procedurer er de som er dokumenteret mest detaljeret.

3.1 Globale konstanter og variable

Wnd_Max	Maximalt antal vinduer som kan være åbne på en gang. Standardværdien er 5.
Scr_Maxx	Maximalt antal skærm punkter på x-aksen. Afhængig af datamattype.
Scr_Maxy	Maximalt antal skærm punkter på y-aksen. Afhængig af datamattype.
Scr_segm	Segmentadressen på det grafiske skærbillede. Afhængig af datamattype.
Scr_aspect	En faktor, som anvendes ved cirkeltegning for at sikre at cirklen er rund på skærmen. Afhængig af datamattype.

Scr_yaddr og Scr_xaddr	Et array som indeholder oplysninger om hvert punkts fysiske placering i det grafiske skærbillede. Afhængig af datamattype.
Wnd_actual	Nummeret på det til enhver tid aktive vindue.
Scr_start	Segmentadressen på starten af tekstskærbilledet. Afhængig af datamattype.
Txt_maxx og Txt_maxy	Maximalt antal linier og kolonner som kan udskrives som tekstlinier. Man skal være opmærksom på at disse variable ændrer værdi afhængig af om datamaten er i grafik- eller teksttilstand.
Graph_type	Angiver den aktuelle type af datamat. Følgende muligheder findes i øjeblikket:
IBM_high:	IBM Color Graphics Adapter i 640 x 200 punkter
IBM_Enhan:	IBM Enhanced Color Adapter i 640 x 300 punkter
Oli_high	Olivetti M24 høj opløsningsgrafik i 640 x 400 punkter
PAR_high	Partner højopløsningsgrafik i 720 x 352 punkter
PIC_high	Piccolinens højopløsningsgrafik i 560 x 256 punkter

For Partner og Piccolinens vedkommende er det ligegyldigt hvilken parameter som anvendes her, da CPI-graf selv undersøger hvilken datamat der arbejdes med.

3.2 Hjælpeprocedurer

De følgende procedurer er alene beregnet til at udføre de mest simple operationer på skærbilledet. Flere af dem er maskinspecifikke. Dokumentationen er ikke fuldstændig, og hvis man ønsker at ændre dem, må man lave en udskrift af kildeteksten. Der gives

blot er overfladisk beskrivelse og en udskrift af den første linie i den rækkefølge de forekommer i programmet.

PROCEDURE init_graphics(typ: Scr_type);

Grafikskærmens initialisering af variable, indstilling af skærkoordinater og indlæsning af grafisk tegnfont fra '8x8font.chr'. (Maskinspecifik procedure)

PROCEDURE init_text;

Initialisering af tekstskærm. (Maskinspecifik procedure)

PROCEDURE writestr(col,row: INTEGER; str: str80);

Udskrivning af en tekst på tekstskærmen i de absolutte koordinater col,row. Farverne i de aktive vindue anvendes.(Maskinspecifik procedure)

FUNCTION spc(l: INTEGER; ch: CHAR):str80;

Funktion der returnerer en tekst med længden l og som indeholder tegnet i ch.

FUNCTION readstrtxt(x,y,l: INTEGER): str80;

Funktion der returnerer en tekst fra tekstskærmen i koordinaterne x,y, med længden l.

PROCEDURE swap(VAR a,b: INTEGER);

Ombytter værdierne i variablene a og b.

PROCEDURE scr_writedot(x,y: INTEGER);

Der udskrives et punkt i grafiskskærmen i de absolutte koordinater x,y. Punktets farve afhænger af vinduets farve.

PROCEDURE movescrtomem(x,y,xlen: INTEGER;k1,k2: INTEGER);

Der flyttes et antal bytes fra grafiskskærmen i koordinaterne x,y til en variabel med segmentadressen k1 og ofsetadressen k2. Antallet af bytes der skal flyttes er xlen.

PROCEDURE movememoscr(x,y,xlen: INTEGER;k1,k2:INTEGER);

Samme som **movescrtomem**, blot flyttes fra en variabel til grafiskskærm.

PROCEDURE fillvideoline(x,y,xlen,color: INTEGER);

En horisontal linie i grafiskskærmen med længden xlen fyldes med et bitmønster, som ligger i color.

FUNCTION baseaddr(x,y: INTEGER): INTEGER;

En funktion som returnerer den absolutte skærmadresse af punktet i x,y.

FUNCTION isqrt(arg: INTEGER):INTEGER;

En funktion der returnerer kvadratroden af et heltal.

FUNCTION convert(xx:real):INTEGER;

Konvertering af et decimaltal til et heltal. Kontrol med om heltallet ligger indenfor gyldighedsområdet af heltallet.

```
FUNCTION _Wld_ToWindowX(x : real) : INTEGER;
```

Omregning af x-aksen fra det skalerede brugerniveau til vinduesniveau. Det undersøges samtidigt at værdierne ligger indenfor det gyldige område for heltalsvariable.

```
FUNCTION _Wld_ToWindowY(y : real) : INTEGER;
```

Samme som `_Wld_ToWindowx` blot for y-aksen.

3.3 Procedurer på skærmniveau

Følgende procedurer er alle defineret på skærmniveau. Der sker ingen klipning ved skærmgrænserne og det er brugerens opgave at sørge for at holde sig indenfor skærmens fysiske ramme. Koordinaterne der angives er alle i absolutte skærmskoordinater.

```
FUNCTION Scr_ReadDot(x,y:INTEGER):BOOLEAN;
```

En funktion som returnerer værdien sand eller falsk afhængig af om punktet `x,y` i absolutte skærmskoordinater er lig forgrundsfarven eller ikke.

```
PROCEDURE Scr_fill(x1,y1,x2,y2:INTEGER; color: BYTE);
```

Et område på skærmen udfyldes med et bitmønster lagret i `color`. `x1,y1` er nederste venstre hjørne og `x2,y2` er øverste højre hjørne udtrykt i absolutte skærmskoordinater.

```
PROCEDURE Scr_Drawchar(x,y: INTEGER;ch: CHAR);
```

Tegnet `ch` tegnes i punktet `x,y` i absolutte skærmskoordinater. Værdien af `x` skal være et multiplum af 8. Der tegnes med forgrundsfarven fra det aktive vindue.


```
PROCEDURE Scr_drawLine(xx1,yy1,xx2,yy2: INTEGER);
```

Der tegnes en linie fra punktet xx1,yy1 til xx2,yy2 i forgrundsfarven fra det aktive vindue. Der foregår ingen klipning af linien.

```
PROCEDURE Scr_operation(opr: oprtype;xl, y1, x2, y2:INTEGER;
                        var buffer);
```

Der flyttes et område fra eller til skærmen. Området afgrænses af de absolutte skærmkoordinater xl,y1 for nederste venstre hjørne og x2,y2 for øverste højre hjørne. Området flyttes til skærmen fra en buffer, hvis variabelen opr er sat lig 'load' og omvendt, hvis opr er sat lig 'save'. Variablen buffer er en pointervariabel.

3.4 Procedurer på vinduesniveau

Følgende procedurer arbejder alle på et vinduesniveau, men i absolute vindueskoordinater, der starter med 0,0 i nederste venstre hjørne af vinduet. Hvis koordinaterne angives udenfor det aktive vindue vil tegningen ikke blive udført. Der tegnes med forgrundsfarven fra det aktive vindue.

```
PROCEDURE Wnd_WriteDot(x, y: INTEGER);
```

Det tegnes et punkt på skærmen i det aktive vindues forgrundsfarve. x,y angiver absolute vindueskoordinater.

FUNCTION Wnd_ReadDot(x, y : INTEGER) : BOOLEAN;

Det undersøges om punktet x,y er sat til forgrunds-
farven, og hvis det er tilfældet returneres værdien
'sand' ellers returneres værdien 'falsk'.

FUNCTION _Wnd_ClipLine(VAR x1, y1, x2, y2 : INTEGER) : BOOLEAN;

Funktionen returnerer værdien 'sand' eller 'falsk'
afhængig af om linien fra x1,y1 til x2,y2 ligger
indenfor det aktive vindue eller ikke. Variablene
x1,y1,x2 og y2 bliver justeret således at linie-
punkterne kommer til at være indenfor vinduet,
hvis det er muligt.

PROCEDURE Wnd_DrawLine(x1, y1, x2, y2: INTEGER);

Der tegnes en linie i det aktive vindue fra x1,y1
til x2,y2. Klipping af linien sker automatisk.

PROCEDURE Wnd_DrawBox(x1, y1, x2, y2 : INTEGER);

Der tegnes et parallelogram i det aktive vindue.
Klipping af firkanten sker automatisk.

PROCEDURE Wnd_DrawChar(x, y : INTEGER; c : CHAR);

Der tegnes et tegn c i det aktive vindue. Det sikres
automatisk at værdien af x-aksen er et multiplum
af 8.

PROCEDURE Wnd_DrawText(x, y : INTEGER; t :str80);

Der tegnes en tekst t i det aktive vindue. Det
sikres automatisk at værdien af x-aksen er et multi-
plum af 8.

```
PROCEDURE Wnd_circle(cx, cy, radius: INTEGER);
```

Der tegnes en cirkel i det aktive vindue med centrum i punktet cx, cy og med en radius på 'radius'.

3.5 Procedurer på brugerniveau

De følgende procedurer er beregnet til at brugeren umiddelbart kan anvende dem. De arbejder i skalerede koordinater i det aktive vindue. Dokumentationen er derfor relativt omfattende, hvilket betyder at alle parametre er forklaret, og at der gives eksempler på anvendelsen. Betegnelsen tilstand kan enten være grafik eller tekst. I grafiktilstand virker procedurerne kun når den grafiske skærm er gjort aktiv ved at kalde proceduren **graphmode**. I teksttilstand virker proceduren kun ved at tekstskeæmbilledet er gjort aktivt ved at kalde proceduren **textmode**. Hvis et navn er fremhævet med fed skrift betyder det at proceduren er beskrevet i detaljer i dette afsnit.

Definition	PROCEDURE circle(x, y, r: REAL);	
Niveau	bruger	
Tilstand	grafik	
Eksempel	circle(1.0, -2.0, 3);	
Parametre	x, y	Koordinaterne i det aktuelle vindue for cirkelns midtpunkt
	r	Cirkelns radius
Funktion	Tegner en cirkel i det aktuelle vindue. Cirkelns midtpunkt ligger i koordinatparret x, y . Der sker en afklipping, hvis cirklen går ud over vinduesgrænsen.	

Definition	Drawaxis(xdens, ydens: INTEGER; gitter: boolean);	
Niveau	vindue	
Tilstand	grafik	
Eksempel	Drawaxis(7,7,true);	
Parametre	xdens, ydens	Angiver hvor tæt skalamarkeringerne og y akse skal være. Gyldige værdier er 0 til 9. Hvis værdien er 0 udskrives ingen skalamarkering.
	gitter	En boolsk variabel som angiver om der skal tegnes et gitter i skaleringen eller ikke.
Funktion	Tegner et koordinatsystem i det aktive vindue med den aktive skalering indsat på de to akser.	
Definition	PROCEDURE drawframe;	
Niveau	bruger vindue	
Tilstand	grafik	
Eksempel	drawframe;	
Funktion	Tegner en ramme om det aktuelle vindue og indsætter overskrift, hvis man i f. d. har defineret en sådan med viewporthead . Vinduets tegneområde justeres derefter.	
Definition	PROCEDURE drawheader;	
Niveau	bruger vindue	

Tilstand	grafik	
Eksempel	drawheader;	
Parametre	ingen	
Funktion	Der tegnes en overskrift i invertschrift i det aktuelle vindue, hvis man i forvejen har defineret en sådan med viewporthead . Vinduets tegneområde justeres derefter.	
Definition	PROCEDURE drawreal(x,y: INTEGER; r: REAL; l, d: INTEGER);	
Niveau	bruger vindue	
Tilstand	grafik tekst	
Eksempel	drawreal(0,0,1.234567,4,1);	
Parametre	x,y	Række og kolonnennummer i det aktuelle vindue
	r	Et decimaltal som skal udskrives
	l	Antallet af pladser som skal afsættes totalt til tallet inkl minustegn og decimaltegn.
	d	Antal af pladser som skal afsættes til decimaler.
Funktion	Udskrives et decimaltal i det aktuelle vindue. Decimallet skrives højrejusteret.	
Definition	PROCEDURE drawstr(x, y: INTEGER; t: str80);	
Niveau	bruger	

Tilstand	grafik tekst
Eksempel	<code>drawstr(0,0,'En vilkårlig tekst');</code>
Parametre	<p><code>x,y</code> Række og kolonnenummer i det aktuelle vindue</p> <p><code>t</code> En tekst som udskrives på linien</p>
Funktion	Der udskrives en tekst i det aktuelle vindue. Teksten afkortes, hvis der ikke er plads til alle tegn i vinduet. Teksten udskrives linievis med koordinaterne 0,0 placeret i øverste venstre hjørne af vinduet.
Definition	PROCEDURE drawto(x, y: REAL);
Niveau	bruger
Tilstand	grafik
Eksempel	<code>drawto(200,500);</code>
Parametre	<p><code>x,y</code> Koordinater i det aktuelle vindue som er skaleret</p>
Funktion	Der tegnes en streg fra markørens aktuelle placering til et punkt som angives med koordinaterne <code>x,y</code> . Markørens placering bliver herefter punktet <code>x,y</code> .
Definition	FUNCTION getcolor(x, y: REAL): BOOLEAN;
Niveau	bruger
Tilstand	grafik
Eksempel	<code>IF getcolor(100,50) then ;</code>

Parametre	x,y	Koordinater i det aktuelle vindue som er skaleret.
Funktion		Funktion, som bliver sand, hvis punktet som den peger på, har samme farve som forgrundsfarven i vinduet. Ellers bliver den falsk.
Definition		PROCEDURE graphicscreen (typ:Scr_type)
tilstand		graphic
eksempel		graphicscreen (IBM_high);
parametre	typ:	typen af datamat eller grafikkort
	IBM_high:	IBM PC med COLOR GRAPHICS Adapterkort i farve eller sort/hvid 640 x 200 punkters opløsning
	IBM_enhan:	IBM PC med Enhanced colorgraphics Adapter i farver. 640 x 300 punkters opløsning
	OLI_high:	Olivetti M24 eller M21 i højopløsningsgrafik 640 x 400 punkters opløsning
	PAR_high:	Partner eller Piccoline med 60HZ farve eller sort/hvid skærm.720 x 352 punkters opløsning.
	PIC_high:	Piccoline med 50 HZ farve- eller sort/hvid skærm 560 x 256 punkters opløsning. For Partner og Piccoline gælder det at programmet selv indstiller sig til den korrekte skærmtpe.

Funktion	Denne procedure initialiserer vinduessystemet og det grafiske system til de korrekte størrelser. Grafiske- og tekstvinduer sættes til maximal størrelse. Vinduerne defineres med hvid baggrundsfarve og sort baggrundsfarve. Vinduennummeret vælges automatisk. Datamaten forbliver i teksttilstand.
Definition	PROCEDURE graphmode;
Niveau	bruger
Tilstand	tekst
Eksempel	graphmode;
Parametre	ingen
Funktion	Skærmen sættes i grafisk tilstand. Hvis den allerede er i denne tilstand sker der intet.
Definition	PROCEDURE inputstr(VAR s: str80; term: txt_charset; ch1, ch2: CHAR; VAR xx: INTEGER; x,y,l: INTEGER; VAR changed,inset: BOOLEAN);
Niveau	bruger
Tilstand	grafik tekst
Eksempel	inputstr(line,(.retur.),ch1,ch2,x,0,0,30,changed, inset);
Parametre	s Streng som tilbagefører den indlæste tekst. term Et 'set of char' som angiver hvilke tegn der stopper indlæsningen.

chl	Hvis værdien er <ESC> så er der sidst indlæst en udvidet tastaturkode, ellers er den et blanktegn.
ch2	Det sidst indlæste tegn.
xx	Kolonnen hvor markøren stiller sig ved indlæsning.
x,y	Tekstkoordinaterne i det aktive vindue, hvor teksten skal starte.
l	Den maximale tekstlængde
changed	Tilbageføres med sand, hvis der ikke er rettet i line ellers falsk.
inset	En 'kontakt' der angiver om redigering skal foregå i indsæt eller overskriv tilstand.
Funktion	<p>Proceduren indlæser en vilkårlig tekst fra et vindue på skærmen. Teksten kan rettet med markørtasterne. Hvis man trykker CTRL-J åbnes et vindue der viser de redigeringsmuligheder som findes i proceduren. Vinduet til dette bruger nummer 4 og man skal derfor være varsom med at anvende dette nummer til andet formål.</p>
Definition	PROCEDURE loadviewport(no x,y: INTEGER);
Tilstand	grafik, tekst
Eksempel	loadviewport(1, 0,0);
Parametre	<p>no Nummeret på det aktuelle vindue som skal gøres aktivt.</p> <p>Hvis no er et negativt tal, bliver stakken tømt fra den plads som netop er gældende.</p> <p>Hvis no er positivt, forbliver vinduet på stakken efter at det er hentet frem.</p>

x,y Er en værdi som angiver det antal af punkter/linier vinduet skal forskydes med i forhold til den oprindelige placering af vinduet.

Funktion Et vindue hentes frem som i forvejen er gemt med **saveviewport**. Vinduet bliver herefter til det aktive vindue på skærmen.

Definition **PROCEDURE moverel(x,y: REAL);**

Niveau bruger

Tilstand grafik

Eksempel moverel(1,1);

Parametre **x,y** Koordinater i det aktuelle vindue som er skaleret.

Funktion Flytter markøren til et nyt punkt uden at der tegnes i vinduet. Det nye punkt er relativt i forhold til markørens oprindelige placering.

Definition **PROCEDURE moveto(x, y: REAL);**

Niveau bruger

Tilstand grafik

Eksempel moveto(10,300);

Parametre **x,y** Koordinater i det aktuelle vindue som er skaleret.

Funktion	Markøren flyttes til punktet x,y uden at der tegnes i vinduet.
Definition	PROCEDURE plot(x, y: REAL)
Tilstand	grafik
Eksempel	plot(20,300);
Parametre	x,y Angiver et punkt i koordinatsystemet i det aktuelle vindue. Punktet skaleres automatisk.
Funktion	Sætter et punkt i det aktive vindue i den aktuelle forgrundfarve. Hvis koordinaterne ligger udenfor grænserne for vinduet vises punktet ikke. Skaleringen sker med sætningen WINDOW .
Definition	PROCEDURE plotchar(x,y: REAL; c: CHAR);
Niveau	bruger
Tilstand	grafik
Eksempel	plotchar(0,0,'A');
Parametre	x,y Koordinater i det aktuelle vindue som bliver skaleret. c Et ASCII tegn som skal udskrives.
Funktion	Der tegnes et tegn i det aktuelle vindue fra punktet x,y . Tegnet står 'ovenpå' punktet. Placeringen af tegnet er ikke linieorienteret, men det kan placeres et vilkårligt sted horisontalt. Vertikalt kan tegnet kun placeres i en x -koordinat, som er et multiplum af 8.

- Definition **PROCEDURE** `plottext(x,y: REAL; T: str80);`
- Niveau bruger
- Tilstand grafik
- Eksempel `plottext(10,20,'Dette er et tekst');`
- Parametre `x,y` Koordinater i det aktuelle vindue. Disse er skaleret
 `t` En tekst som skal tegnes i punktet
- Funktion Der tegnes en tekst i det aktuelle vindue fra punktet `x,y`. Teksten står 'ovenpå' punktet. Se bemærkningerne for `plotchar`.
-
- Definition **PROCEDURE** `saveviewport(no: INTEGER);`
- Tilstand grafik, tekst
- Eksempel `saveviewport;`
- Parametre `no` Nummeret på det vindue som skal gemmes på stakken.
- Funktion Hele det aktive vindue på skærmen bliver gemt sammen definitionen af vinduets størrelse. Vinduet kan hentes frem igen med `loadviewport(1)`. Vinduet bliver gemt på en dynamisk stak i datamatens lager. Hvis der ikke er tilstrækkelig plads i lageret, stoppes programmet med en fejlmeddelelse.

Definition	PROCEDURE Scr_Dump(mode: prn_type);
Niveau	skærm
Tilstand	grafik
Eksempel	Scr_dump(IBM_graph);
Parametre	mode Angiver den grafiske printer som er tilkoblet datamaten. Rc_603 og IBM_graph.
Funktion	Udskriver et skærbillede på en tilkoblet grafik-printer.

Definition	PROCEDURE selectviewport(no: integer);
Tilstand	grafik, tekst
Eksempel	selectviewport(1);
Parametre	no Nummeret på vinduet som skal gøres aktivt.
Funktion	Det angive vindue gøres aktivt på skærmen. Alle følgende operationer uanset om det er tekst eller grafik har alene betydning for dette vindue indtil man vælger at gøre et andet vindue aktivt med en ny sætning eller med loadviewport . Vinduet skal i forvejen være defineret med viewport eller text-viewport .

Definition	PROCEDURE selectwindow(no: integer);
Tilstand	grafik
Eksempel	selectwindow(1);

Parametre no Nummeret på den skalering som skal anvendes i det aktive vindue.

Funktion Det aktive vindue blive skaleret med den definition, som er gemt under det angivne nummer. I forvejen skal man have defineret skaleringen med **window**.

Definition **PROCEDURE textmode;**

Niveau bruger

Tilstand grafik

Eksempel textmode;

Parametre ingen

Funktion Skærmen sættes i teksttilstand. Hvis den allerede er i teksttilstand sker der intet.

Definition **PROCEDURE textviewport(no, xx1, xx2, yy1,yy2: Integer);**

Niveau bruger

Tilstand grafik, tekst

Eksempel textviewport(1,0,79,0,2);

Parametre no nummeret på vinduet som skal defineres
 xx1 venstre kolonne i vinduet
 xx2 Højre kolonne i vinduet
 yy1 Øverste linie i vinduet
 yy2 Nederste linie i vinduet

Funktion	Proceduren definerer et bestemt område på skærmen som et vindue. Definitionen er tekstorienteret, men der kan i grafiktilstand også tegnes i vinduet. Vinduet er defineret ensartet uanset om det anvendes i tekst- eller grafiktilstand.
Restriktioner	xx2 skal være større end xx1 og yy2 skal være større end yy1. Hvis linierne eller kolonnerne ligger udenfor det fysiske skærbillede vil de automatisk blive afkortede. Man kan bruge to variable til at angive maximale antal af kolonner og linier. TXT_MAXX angiver maximale antal af kolonner og TXT_MAXY det maximale antal linier, der kan anvendes i det fysiske skærbillede.
Definition	PROCEDURE textviewportcolor(no, txtcolor, bckcolor: integer);
Niveau	bruger
Tilstand	tekst
Eksempel	textviewportcolor(1, whitetxt, bluetxt);
Parametre	no Nummeret på vinduet som skal defineres txtcolor Farven på teksten - forgrundsfarven bckcolor Baggrundsfarven Gyldige farver er: blacktxt, bluetxt, greentxt, cyantxt, redtxt, magentatxt, yellowtxt og whitetxt.
Funktion	Der defineres for- og baggrundsfarver i det valgte vindue.
Definition	PROCEDURE viewport(no, xx1, xx2, yy1, yy2: INTEGER);

Niveau	bruger										
Tilstand	grafik										
Eksempel	<code>viewport(1,0,200,0,50);</code>										
Parametre	<table> <tr> <td>no</td> <td>Nummer på vindue som skal defineres</td> </tr> <tr> <td>xx1</td> <td>Venstre x-koordinat</td> </tr> <tr> <td>xx2</td> <td>Højre x-koordinat</td> </tr> <tr> <td>yy1</td> <td>Nederste y-koordinat</td> </tr> <tr> <td>yy2</td> <td>Øverste y-koordinat</td> </tr> </table>	no	Nummer på vindue som skal defineres	xx1	Venstre x-koordinat	xx2	Højre x-koordinat	yy1	Nederste y-koordinat	yy2	Øverste y-koordinat
no	Nummer på vindue som skal defineres										
xx1	Venstre x-koordinat										
xx2	Højre x-koordinat										
yy1	Nederste y-koordinat										
yy2	Øverste y-koordinat										
Funktion	Der defineres et bestemt område på skærmen som et vindue. Definitionen gælder alene i grafisk tilstand.										
Restriktioner	x-koordinaterne skal være et multiplum af 8, hvis det ikke er tilfældet bliver de automatisk justeret. Hvis koordinaterne ligger udenfor det fysiske skærbillede bliver de automatisk tilpassede. Skærbilledets størrelse er defineret med to variable: SCR_MAXX og SCR_MAXY for henholdsvis største antal punkter horisontalt og vertikalt i det fysiske skærbillede.										
Definition	PROCEDURE viewportcolor(no, fcolor, bcolor: INTEGER);										
Niveau	bruger										
Tilstand	grafik										
Eksempel	<code>viewportcolor(1, white, black);</code>										
Parametre	<table> <tr> <td>no</td> <td>Nummeret på vinduet som skal defineres</td> </tr> <tr> <td>fcolor</td> <td>Forgrundsfarven</td> </tr> <tr> <td>bcolor</td> <td>Baggrundsfarven</td> </tr> </table> Gyldige farver er:	no	Nummeret på vinduet som skal defineres	fcolor	Forgrundsfarven	bcolor	Baggrundsfarven				
no	Nummeret på vinduet som skal defineres										
fcolor	Forgrundsfarven										
bcolor	Baggrundsfarven										

black, white, transp.

Funktion	For- og baggrundsfarven defineres i det angivne vindue. I grafiktilstand kan der kun vælges to "farver" enten sort eller hvid. Hvis farven sættes til transp. vil punkterne blive sorte hvis de var hvide i forvejen og omvendt.
Definition	PROCEDURE viewportheader(no: integer; head: str80);
Niveau	bruger
Tilstand	grafik, tekst
Eksempel	viewportheader(1, 'Dette er en overskrift');
Parametre	no Nummer på vinduet som skal forsynes med overskrift head Overskriften
Funktion	Der defineres en overskrift på de vinduer som udskrives på skærmen med sætningerne writeframe eller drawframe .
Definition	PROCEDURE window(no: integer; xleft, xright, ydown, yup: REAL);
Tilstand	grafik, tekst
Eksempel	window(1, -100.0, 1E3, 0, -2);
Parametre	no Nummeret på skaleringen som skal defineres. xleft, xright Skalering af x-aksen i koordinatsystemet ydown, yup Skalering af y-aksen i koordinatsystemet

Funktion Proceduren definerer skaleringen for x og y-aksen i et koordinatsystem. Skaleringen bruges i grafiktilstand til alle tegneoperationen på brugerniveau. Hvis **window** ikke er blevet defineret er skaleringen identisk med de absolutte skærmmkoordinater.

Definition **PROCEDURE writeframe;**

Niveau bruger vindue

Tilstand grafik tekst

Eksempel writeframe;

Parametre ingen

Funktion Tegner en ramme med ASCII linietegning omkring det aktuelle vindue. I øverste linie indsættes en overskrift, hvis en sådan er defineret med **viewport-header**. Vinduets tegneområde justeres derefter. Der benyttes et IBM standard tegnsæt til denne procedure.

Definition **PROCEDURE writehelp(no, w, l: integer; var st);**

Niveau bruger

Tilstand grafik tekst

Eksempel const
 help: string(.40.) =
 'Hjælpetekst som udskrives i vindue';
 begin
 textviewport(4,0,42,0,3);
 writehelp(4,40,1,help);

Parametre	no	Nummeret på vinduet som skal anvendes.
	w	Længden af teksten.
	l	Antal linier
	st	Tekstvariablen som skal stå i vinduet.

Funktion Udskriver en tekst som i forvejen skal være defineret i et vindue på skærmen. Der skrives en ramme omkring teksten. Det er vigtigt at længden på strengvariable st er nøjagtig den samme som værdien af variabelen w.

4. Appendix A

Definition af tegnsæt

For at få en ensartet måde at benytte CPI-graf på, er det muligt at omdefinere tegnsættet i Partner/Piccolinedatamaten, således at det bliver identisk med det, som findes på en IBM PC. Til det formål er der udviklet to tegnsætsdefinitionsfiler:

IBMPIC	Piccolinetegnsæt
IBMPAR	Partnertegnsæt

som begge ligger på disketten.

A/S Regnecentralen af 1979 har udviklet et program CHARSET, som er i stand til at omdefinere tegnsættet. Det gøres på følgende måde for Piccolinens vedkommende

```
A > CHARSET std = IBMPIC
```

og for Partner:

```
A > CHARSET std = IBMPAR
```

Programmet er offentligt tilgængeligt og findes på disketten til Partner/Piccoline.



CIP-graf er en samling procedurer skrevet i Polypascal som giver programmøren mulighed for på en nem og smertefri måde at lave vinduer på skærmen. Vinduer kan laves både i grafik- og teksttilstand. De grafiske vinduer kan individuelt skaleres, og de sædvanlige grafiske sætninger fra fx Comal kan ligeledes anvendes.

Pris 480,- kr. incl. moms

CPI er en selvejende institution under tilsyn af Undervisningsministeriet, oprettet i 1984 på grundlag af en bevilling fra Egmont Fonden. CPI har til formål at fremme en hensigtsmæssig anvendelse af informatik i undervisningen, primært i folkeskolen og gymnasiet.

Det er en vigtig arbejdsopgave for CPI at indhente og sprede information. CPI gennemfører også uddannelsesaktiviteter, fx kurser, workshops, seminarer eller konferencer. På relevante faglige områder iværksætter CPI forsknings- og udviklingsarbejde, dels ved hjælp af egne medarbejdere, og dels ved hjælp af specialister og forskere udefra.

En særlig udviklingsopgave består i at støtte produktionen af programmel til undervisning, og at udvikle den nødvendige pædagogik hertil.

CPI kan yde teknisk og pædagogisk vejledning inden for de nævnte områder.

Dansk Center for Pædagogik og Informatik
Vesterbrogade 72, 1620 København V
Telf. 01-23 71 11