

## 25 Steuerung des Druckers

### Gliederung

25.1	Vorbemerkungen.....	2
25.2	Druckersteuerung in Turbo-Pascal .....	2
25.3	Einzelnadelansteuerung bei graphikfähigen Nadeldruckern .....	5
25.4	Die Seitenbeschreibungssprache HP-PCL.....	8

## 25.1 Vorbemerkungen

Zu den Begriffen CPI und LPI:

- CPI characters per inch. Schreibdichte bei Schriften mit festem Abstand (Courier, Pica, MS LineDraw, ...). Anzahl der Zeichen pro Zoll. 1 Zoll (1 inch) = 25.4 mm. Übliche Werte für CPI: 10, 12, 15, 17 und 20. Neben den Schriften mit fixer Teilung bieten alle modernen Drucker auch noch Proportionalschriften an, z.B. die Schrift Times New Roman. Neben den fest installierten Schriften stehen über geeignete Treiber die gesamten Schriften des Betriebssystems zur Verfügung, wie z.B. alle installierten TrueType-Schriften unter Windows.
- LPI lines per inch. Übliche Werte: 6 und 8. Bei 8 Zeilen/inch und üblicher Zeichengröße 12 pt schlechte Lesbarkeit.

Mit 10 CPI können bei A4-Papier bis zu 80 Zeichen in eine Zeile gedruckt werden. Durch den nichtbedruckbaren Rand gibt es aber bei Tinten- und Laserdruckern Einschränkungen.

Bei perforiertem Endlos-Tabellierpapier beträgt die Abschnittlänge 12 Zoll (= 304.8 mm) und damit ca. 8 mm länger als die A4-Höhe.

Drucker-Steueranweisungen sind nicht genormt, sondern orientieren sich an Industriestandards. Beispiele: Epson-Druckerstandard Esc/P, PCL, PreScribe, PostScript.

Mit geeigneter Software (Druckertreiber in Anwenderprogrammen wie z.B. Textverarbeitung) kann der Anwender die "Intelligenz" der Drucker nutzen, ohne sich um die Steueranweisungen kümmern zu müssen.

Moderne Matrix-Drucker sind grafikfähig. Bei diesen Druckern können durch geeignete Programmierung der Ansteuerung der Nadeln oder Düsen praktisch beliebige Zeichen dargestellt werden.

Laserdrucker besitzen sehr leistungsfähige Seitenbeschreibungssprachen wie z.B. »PreScribe«, Fa. Kyocera, PCL, Fa. Hewlett Packard oder »PostScript«, Fa. Adobe. Mit »PostScript« können über einen RIP (Raster Image Processor) auch geeignete Film- und Plattenbelichter angesteuert werden. Mit Ausnahme von PCL werden Seitenbeschreibungssprachen hier nicht behandelt. PostScript wird in DV II behandelt.

## 25.2 Druckersteuerung in Turbo-Pascal

Für Druckerausgaben ist in Turbo-Pascal grundsätzlich die Unit »PRINTER« zu deklarieren.

Steueranweisungen für den Drucker sind mit

```
Write(Lst, steueranweisung [, ...] );
```

abzusetzen, wobei »*steueranweisung*«: String-Ausdruck

Das folgende Beispiel zeigt die Steueranweisung, die bei einem Ducker mit der Esc/P-Druckersprache (Epson oder Kompatible) die Breitschrift einstellt:

**Schreibweise 1a:** `Write(Lst, Chr(27), Chr(87), Chr(1));`  
**Schreibweise 1b:** `Write(Lst, #27, #87, #1);`  
**Schreibweise 1c:** `Write(Lst, #27 + #87 + #1);`  
**Schreibweise 1d:** `Write(Lst, #27#87#1); { hier keine Blanks! }`

Die Zahlen stellen Ordnungsnummern nach ASCII dar.

Wenn der Ordnungsnummer ein Zeichen zugeordnet ist, das über die Tastatur erreichbar ist (siehe ASCII-Tabelle Kap. 13), dann kann auch unmittelbar das Zeichen selbst als eine String-Konstante in den String-Ausdruck eingesetzt werden.

Der Ordnungsnummer 87 ist laut ASCII-Tabelle das Zeichen 'W' zugeordnet. Somit kann die Steueranweisung auch wie folgt geschrieben werden:

**Schreibweise 2:** `Write(Lst, #27, 'W', #1);`

Die Ordnungsnummer war in diesem Beispiel dezimal zu verstehen. In den Drucker-Handbüchern sind die Codes oft hexadezimal angegeben. Eine Umrechnung in dezimal ist nicht notwendig, wenn man der Hex-Zahl die dafür vorgesehenen Turbo-Pascal-Kennung »\$« voranstellt. Somit kann man die gleiche Steueranweisung auch wie folgt formulieren:

**Schreibweise 3:** `Write(Lst, #$1B, #$57, #1);`

Kombinationen aller Schreibweisen sind zulässig. Hexadezimal-Buchstaben können groß oder klein eingegeben werden.

Im gezeigten Fall war die Steueranweisung eine sog. Escape-Sequenz, da das erste Zeichen das Steuerzeichen »ESC« war, siehe Kap. 13.

Diese Escape-Sequenz war insgesamt drei Zeichen lang, sie kann aber auch nur zwei Zeichen umfassen oder auch noch wesentlich länger sein, wie im Falle der Seitenbeschreibungssprache PCL. Daneben liegen oft noch Steueranweisungen vor, die nur ein Zeichen umfassen. In diesem Fall muß die Ordnungsnummer verständlicherweise unter 32 liegen. Außerdem gibt es auch noch Steueranweisungen, die länger als ein Zeichen sind und die nicht mit »ESC«, sondern mit einem anderen Zeichen beginnen, dessen Ordnungsnummer aber dann verständlicherweise unter 32 liegen muß.

Die folgende Tabelle zeigt eine Auswahl von Steueranweisungen für den 24-Nadel-drucker NEC P6. Eine vollständige Aufstellung ist im Handbuch enthalten.

Steueranweisungen (Auswahl)	Dez	Hex	Dez ggf. mit Zeichen
Schritt zurück, BS	008	08	008
Horizontal-Tabulator, HT	009	09	009
Zeilenvorschub, LF	010	0A	010

Vertikal-Tabulator, VT	011	0B	011
Seitenvorschub, FF	012	0C	012
Wagenrücklauf, CR (mit LF)	013	0D	013
Schmalschrift (17 CPI) ein: Schmalschrift aus:	015 018	0F 12	015 018
Breitschrift ein (1 Zeile): Breitschrift aus (1 Zeile):	014 020	0E 14	014 020
Unterstreichen ein: Unterstreichen aus:	027 045 001 027 045 000	1B 2D 01 1B 2D 00	027 '-' 001 027 '-' 000
Fettschrift ein: Fettschrift aus:	027 071 027 072	1B 47 1B 48	027 'G' 027 'H'
Hervorgehobene Schrift ein: Hervorgehobene Schrift aus:	027 069 027 070	1B 45 1B 46	027 'E' 027 'F'
hochgestellte Schrift ein: tiefgestellte Schrift ein: hoch/tief Schrift aus:	027 083 000 027 083 001 027 084	1B 53 00 1B 53 01 1B 54	027 'S' 000 027 'S' 001 027 'T'
Breitschrift ein (n Zeilen): Breitschrift aus (n Zeilen):	027 087 001 027 087 000	1B 57 01 1B 57 00	027 'W' 001 027 'W' 000
Zeichensatz 1: Zeichensatz 2:	027 054 027 055	1B 36 1B 37	027 '7' 027 '6'

Führende Nullen sind in der Tabelle nur zur besseren Unterscheidung der Dez-Codes (dreistellig) von den Hex-Codes (zweistellig) angegeben. Sie brauchen nicht, können aber eingegeben werden.

**Beispiel:** Hochgestellte Schrift einschalten, in verschiedenen Schreibweisen

- 1) Write(Lst, #27, #83, #0);
- 2) Write(Lst, #27, 'S', #0);
- 3) Write(Lst, #\$1B, #\$53, #\$0);

Wenn in einem Programm sehr häufig Drucker-Steueranweisungen vorkommen (typisches Beispiel: Textverarbeitungsprogramme), ist es zweckmäßig, die Steueranweisungen nicht als einzelne Zeichen zu senden, sondern als eine deklarierte String-Konstante. Wenn das Programm für einen anderen Drucker angepaßt werden soll, ist lediglich die Konstanten-Deklaration zu modifizieren.

Das folgende Demo-Programm veranschaulicht diese Vorgehensweise. Das Beispiel gilt für den 24-Nadeldrucker NEC P6.

```

program Pas25021; { Steueranweisungen für den Drucker }
uses
    PRINTER;      { Für Druckerausgaben notwendig }
const           { Druckersteuerzeichen für Nadeldrucker NEC P6 }
    Schmal_Ein = #15;      { oder: #$0F }
    Schmal_Aus = #18;      { oder: #$12 }
    Fett_Ein   = #27#71;   { oder: #27'G', oder: #$1B#$47 }
    Fett_Aus   = #27#72;   { oder: #27'H', oder: #$1B#$48 }

```

```

Unter_Ein = #27#45#1; { oder: #27'-'#1, oder: #$1B#$2D#$1 }
Unter_Aus = #27#45#0; { oder: #27'-'#0, oder: #$1B#$2D#$0 }
Breit_Ein  = #27#87#1; { oder: #27'W'#1, oder: #$1B#$57#$1 }
Breit_Aus  = #27#87#0; { oder: #27'W'#0, oder: #$1B#$57#$0 }
FF         = #12;      { Form Feed, Seitenvorschub }
              { Keine Blanks zwischen aneinandergesetzten Zeichen! }

begin
  WriteLn(Lst, 'Das ist Normalschrift. ');
  Write( Lst, Schmal_Ein);
  Write( Lst, 'Das ist Schmalschrift. ');
  WriteLn(Lst, Unter_Ein, 'Das ist unterstrichene Schmalschrift. ');
  WriteLn(Lst, Schmal_Aus, Unter_Aus, 'Das ist Normalschrift. ');
  WriteLn(Lst, Fett_Ein, 'Das ist Fettschrift', Fett_Aus);
  WriteLn(Lst, 'Das ist wieder Normalschrift. ');
  WriteLn(Lst, Unter_Ein, 'Das ist Unterstreichung.', Unter_Aus);
  WriteLn(Lst, 'Das ist wieder Normalschrift. ');
  WriteLn(Lst, Breit_Ein, 'Das ist Breitschrift.', Breit_Aus);
  WriteLn(Lst, 'Das ist wieder Normalschrift. ');
  Write( Lst, FF);
end.

```

Hier die eingescannte und um ca. 40% vergrößerte Ausgabe des NEC P6:

```

Das ist Normalschrift.
Das ist Schmalschrift. Das ist unterstrichene Schmalschrift.
Das ist Normalschrift.
Das ist Fettschrift
Das ist wieder Normalschrift.
Das ist Unterstreichung.
Das ist wieder Normalschrift.
Das ist Breitschrift
Das ist wieder Normalschrift.

```

Wie das Programm zeigt, können die Steueranweisungen auch unmittelbar vor den zu druckenden Text gestellt werden und auch kombiniert werden.

Abschließend sei noch erwähnt, daß bei vielen Druckern einige Steuerfunktionen auch hardwaremäßig eingestellt werden können (Tasten am Bedienpult und/oder DIP-Schalter im Inneren der Geräte). Die Informationen sind den Handbüchern zu entnehmen.

## 25.3 Einzelnadelansteuerung bei graphikfähigen Nadeldruckern

Das folgende Programm zeigt beispielhaft das Drucken eines selbstdefinierten Sonderzeichens (Emblem der Fachhochschule München). Dazu ist ein graphikfähiger Drucker mit 24 Nadeln und mit Steuersequenzen notwendig, die denen des Druckers Epson LQ

1500 entsprechen. Die 24-Nadel-Drucker der NEC-Pinwriter-Serie (z.B. NEC P6) erfüllen diese Voraussetzung.

Beim Programmieren ist zu beachten, daß die 24 Nadeln in Gruppen zu je 8 Nadeln (oberes Zeichen, mittleres Zeichen und unteres Zeichen) angesprochen werden müssen. Die oberste Nadel jeder Gruppe hat den Wert  $2^7 = 128$ , die nächste den Wert  $2^6 = 64$  usw. Das lästige Abzählen der Wertigkeiten der zu setzenden Nadelpunkte kann man sich ersparen, wenn man eine "Bit-Map" des gewünschten Zeichens als String-Array in geeigneter Weise in das Programm schreibt und das Berechnen der Wertigkeiten auch dem Programm überläßt. Das Editieren des Zeichens kann bei dieser Vorgehensweise in komfortabler Weise visuell geschehen. Bei häufig benötigten Sonderzeichen empfiehlt es sich, die Bit-Map des Zeichens abzuspeichern.

```

program Pas25031;  { ***** Grafik für 24-Nadel-Drucker ***** }
{ Druckt mittels eines 24-Nadel-Druckers das Emblem der FH München.
  Druckersteuerzeichen nach Epson LQ 1500, z.B. auch für NEC P6

  Zum Graphik-Modus »Esc * (Mode) (n1) (n2)« bei 24 Nadeln
  Mode = 39: Dreifache Dichte, 180 DPI (dots per inch)
  Mode = 32: Einfache Dichte,   60 DPI
  Mode = 33: Doppelte Dichte,  120 DPI
  Mode = 38: CRT-II,           90 DPI
  n1  = 27   ┌ Anzahl der Spalten = n1 + n2*256
  n2  = 0   └ Hier: 27 + 0*256   = 27 Spalten

  Beschränkungen von SpalteMax, n1Max und n2Max in Abhängigkeit
  von DPI nach Druckerhandbuch "NEC PINWRITES P6/P7 Series,
  Technical Reference Guide", Seite 7-9, Table 7-5:

```

DPI	SpaltenMax	n2Max	n1Max
60	816	3	48
120	1632	6	96
180	2448	9	144
240	3264	12	192
360	4896	19	32

```

}

```

**uses**

CRT, PRINTER;

**const**

```

NadelMax    = 24;  { 24-Nadel-Drucker }
Mode        = 39;
n1          = 27;  { Grafikzeichen mit 27 Spalten }
n2          = 0;
GraphikMode = Chr(27) + '*' + Chr(Mode) + Chr(n1) + Chr(n2);
SpalteMax   = n1 + n2*256;

```

**var**

```

Spalte, Spur,
Oben, Mitte, Unten:  Byte;
Nadel:  array[1..NadelMax] of string[SpalteMax];
Zeichen: array[1..SpalteMax] of record
        OberesZeichen,

```

```

        MittleresZeichen,
        UnteresZeichen: Char;
    end;

function ZweiHoch(i: Byte): Byte;
var
    j, Temp: Byte;
begin
    Temp := 1;
    for j := 1 to i do
        Temp := Temp * 2;
    Zweihoch := Temp;
end;

begin
    {
        { Spalte: 000000000111111111122222222223333333333  Beschränkung }
        { Spalte: 123456789012345678901234567890123456  siehe oben }
        Nadel[01] := '.....'; { Oben, Wert: 128 }
        Nadel[02] := '.....***.....'; { Oben, Wert: 64 }
        Nadel[03] := '.....***.....'; { Oben, Wert: 32 }
        Nadel[04] := '.....***.....'; { Oben, Wert: 16 }
        Nadel[05] := '.....***.....'; { Oben, Wert: 8 }
        Nadel[06] := '....*****.....*****.....'; { Oben, Wert: 4 }
        Nadel[07] := '...*****.....*****.....'; { Oben, Wert: 2 }
        Nadel[08] := '...*****.....*****.....'; { Oben, Wert: 1 }
        Nadel[09] := '...***.....***.....***.....'; { Mitte, Wert: 128 }
        Nadel[10] := '...***.....***.....***.....'; { Mitte, Wert: 64 }
        Nadel[11] := '...*****.....***.....***.....'; { Mitte, Wert: 32 }
        Nadel[12] := '...*****.....***.....***.....'; { Mitte, Wert: 16 }
        Nadel[13] := '...*****.....***.....***.....'; { Mitte, Wert: 8 }
        Nadel[14] := '...***.....'; { Mitte, Wert: 4 }
        Nadel[15] := '...***.....'; { Mitte, Wert: 2 }
        Nadel[16] := '...***.....*****.....'; { Mitte, Wert: 1 }
        Nadel[17] := '...***.....*****.....'; { Unten, Wert: 128 }
        Nadel[18] := '...***.....*****.....'; { Unten, Wert: 64 }
        Nadel[19] := '.....***.....***.....***.....'; { Unten, Wert: 32 }
        Nadel[20] := '.....***.....***.....***.....'; { Unten, Wert: 16 }
        Nadel[21] := '.....***.....***.....***.....'; { Unten, Wert: 8 }
        Nadel[22] := '.....***.....***.....***.....'; { Unten, Wert: 4 }
        Nadel[23] := '.....'; { Unten, Wert: 2 }
        Nadel[24] := '.....'; { Unten, Wert: 1 }

    for Spalte := 1 to SpalteMax do
        begin
            Oben := 0; for Spur := 1 to 8 do
                begin
                    if Nadel[Spur, Spalte] = '*'
                    then Oben := Oben + ZweiHoch(8 - Spur);
                    Zeichen[Spalte].OberesZeichen := Chr(Oben);
                end;
            Mitte := 0; for Spur := 9 to 16 do
                begin
                    if Nadel[Spur, Spalte] = '*'
                    then Mitte := Mitte + ZweiHoch(16 - Spur);
                    Zeichen[Spalte].MittleresZeichen


```

```

:= Chr(Mitte);
end;
Unten := 0; for Spur := 17 to 24 do { 24 Nadeln }
begin
  if Nadel[Spur, Spalte] = '*'
  then Unten := Unten + ZweiHoch(24 - Spur);
  Zeichen[Spalte].UnteresZeichen := Chr(Unten);
end;
end; { von »for Spalte := ...« }
Write(Lst, ' Es folgt das Emblem der FH München: ');
Write(Lst, GraphikMode); { Hier kein »WriteLn«, sonst Verstümmelung }
for Spalte := 1 to SpalteMax do
  Write(Lst, Zeichen[Spalte].OberesZeichen,
        Zeichen[Spalte].MittleresZeichen,
        Zeichen[Spalte].UnteresZeichen );
WriteLn(Lst, ' Alles klar?');
end.

```

Nachstehend die eingescannte PCX-Grafik der Ausgabe des 24-Nadel-Druckers NEC P6.

Das Emblem der FH München:  Alles klar?

## 25.4 Die Seitenbeschreibungssprache HP-PCL

Der Druckerhersteller Hewlett & Packard hat für seine Tinten- und Laserdrucker eine Seitenbeschreibungssprache namens PCL (page control language) entwickelt, die sich mittlerweile neben PostScript zu einem Industriestandard entwickelt hat. Von fast allen anderen Druckerherstellern werden PCL-Emulationen angeboten.

PCL besteht aus einer großen Sammlung von mitunter sehr langen Escape-Sequenzen und ist sehr leistungsfähig. Im Gegensatz zu PostScript ist PCL statisch, d.h. nicht programmierbar. PostScript ist dagegen sowohl eine Seitenbeschreibungssprache, als auch eine Programmiersprache. Im Gegensatz zu den PostScript-Befehlen (genauer Operatoren) sind die PCL-Escape-Sequenzen nicht selbsterklärend. Das ständige Nachschlagen im Druckerhandbuch kann nerven!

PCL ist in verschiedenen Versionen implementiert. Bei HP-Tintendruckern z.B. als PCL-3, bei HP-Laserdruckern in der Version 5 oder 6 (1996). Die kleineren Versionen sind Untermengen der größeren Versionen.

Für Plotter hat HP zusätzlich die Plottersprache "GL/2" (Graphic Language) entwickelt. In der Version 5 von PCL kann GL/2 emuliert werden, wobei auf Laserdruckern naturgemäß keine "Farbstifte" zur Verfügung stehen.



Es ist an dieser Stelle nicht möglich, alle Möglichkeiten von PCL darzustellen. Die folgenden Demo-Programme sollen nur einen Überblick über einige Möglichkeiten von PCL und auch der GL/2-Emulation geben. Im Anhang sind Ausdrücke der einzelnen Programme auf einem HP-LaserJet 4 ausgegeben. Im Ausdruck ist der jeweilige Programmname angegeben.

```

program Pas25041; { Seitenbeschreibungssprache HP-PCL. kha }
  { HP = Hewlett & Packard, PCL = Page Control Language }
  { Hewlett Packard LaserJet/DeskJet und kompatible Drucker. }
  { PCL/5 nur für HP-Laserdrucker. HP-DeskJet besitzt nur PCL/3 }
  { PCL-Steuerbefehle beginnen mit Zeichen Escape = Chr(27) = #27}

uses
  CRT, PRINTER;

const
  Programm = 'Pas25041.PAS';
  LptVor = '1'; { '1' für Lpt1, '2' für Lpt2 }
  LJVor = '4'; { '3' für LJ III, '4' für LJ IV }

var
  Lpt, LJ: Char;
  i: ShortInt;
  iStr: string[3];
  LaserJet3,
  LaserJet4: Boolean;

procedure WriteXY(Sp, Ze: Byte; Meldung: string);
begin
  GotoXY(Sp, Ze);
  Write(Meldung);
end;

procedure DruckerInfos(var Lpt, LJ: Char);
begin
  TextColor(Yellow); TextBackground(Blue);
  ClrScr;
  WriteXY(10, 3, ' PCL-Demo Nr 1: Einfache Beispiele');
  WriteXY(10, 5, ' PCL/5-Drucker, z.B. HP-LaserJet');
  WriteXY(10, 6, '+-----' +
                                     '-----+');
  WriteXY(10, 7, '| 1 An Parallelschnittstelle LPT1 |' +
                                     ' 3 LaserJet III |');
  WriteXY(10, 8, '| 2 An Parallelschnittstelle LPT2 |' +
                                     ' 4 LaserJet IV |');
  WriteXY(10, 9, '| Esc Abbruch |' +
                                     ' Esc Abbruch |');
  WriteXY(10, 10, '| ----- |' +
                                     ' ----- |');
  WriteXY(10, 11, '| <<< Eingabe |' +
                                     ' <<< Eingabe |');
  WriteXY(10, 12, '+-----' +
                                     '-----+');
  WriteXY(12, 11, LptVor); WriteXY(48, 11, LJVor);
  GotoXY(12, 11);
repeat

```

[illegible]

```

    Str(i, iStr);
    WriteLn(Lst, #27 + '&a' + iStr + 'L',
            'Linker Rand ', i, ' Spalten');

    end;
    WriteLn(Lst, #27 + '&a0L');
    WriteLn(Lst, 'Wieder Standardeinstellung für den linken Rand.');
```

WriteLn(Lst,	{ Beispiel für vollständige Schriftdefinition: }
#27 + '(10U',	{ Zeichensatz: U: "10" = PC-8=IBM-Zeichensatz }
	{ U: "8" = HP Roman8 (ANSI ??) }
#27 + '(s1P',	{ Zeichenabstand P: "1" = proport., "0" = fix }
#27 + '(s12H',	{ Zeichendichte H: "12" = 12 cpi, auch F-Reals }
#27 + '(s10V',	{ Zeichengröße V: "10" = 10 pt, auch F-Reals }
#27 + '(s0S',	{ Schriftlage: S: "0" = gerade, "1" = kursiv }
#27 + '(s0B',	{ Strichstärke B: "0" = normal, "-7".."7" }
#27 + '(s4101T'	{ Schrifttyp T: "4101" = CG Times }
	{ T: "3" = Courier }
	{ T: "4148" = Univers (Helvetica) }
	{ T: "0" = Line Printer }

```

);

WriteLn(Lst, 'Schrift "CG Times", 10pt, mit allen ' +
            'Strichstärken von -7 bis +7:');
for i := -7 to 7 do
begin
    Str(i, iStr);
    WriteLn(Lst, #27 + '(s' + iStr + 'B', 'Strichstärke ', i);
end;
WriteLn(Lst);
WriteLn(Lst, #27 + '(s12V', { Schriftgröße 12pt }
        #27 + '(s0B', { Strichstärke 0 = normal }
        'Schrift "CG Times", 12pt, normale Strichstärke "0"');

Write(Lst, #12); { #12 = FormFeed }
Close(Lst);
end.
```

```

program Pas25042; { Seitenbeschreibungssprache HP-PCL. kha }
{ HP = Hewlett-Packard, PCL = Page Control Language }
{ Hewlett Packard LaserJet/DeskJet und kompatible Drucker. }
{ PCL-Steuerbefehle beginnen mit Zeichen Escape = Chr(27) = #27 }

uses
    CRT, PRINTER;

const
    Programm      = 'Pas25042.PAS';
    LptVor        = '1'; { '1' für Lpt1, '2' für Lpt2 }
    LJVor         = '4'; { '3' für LJ III, 'IV' für LJ IV }
    Esc = #27;
    PCL_Reset     = Esc + 'E';
    { Nicht Bezeichner "Reset" verwenden, da Pascal-Standardprozedur }
    Position00    = Esc + '*p000X' + Esc + '*p000Y';
    DPI300        = Esc + '*t300R';
    DPI600        = Esc + '*t600R';
    PCL_FF        = #12; { FF = Form Feed = Seitenauswurf }
```

```

LinkerRand10 = Esc + '&a10L';
LinkerRand0  = Esc + '&a0L';
{ Alle folgenden Schriften mit "V"ertikal = 12 pt,
  bei "fix" mit "H"orizontal = 10 cpi. IBM-Zeichensatz }

Schrift_Courier      = Esc + '(10U' + Esc + '(s0P' + Esc + '(s10H' +
                        Esc + '(s12V' + Esc + '(s0S' + Esc + '(s0B' +
                        Esc + '(s3T';
Schrift_Univers      = Esc + '(10U' + Esc + '(s1P' + Esc + '(s10H' +
                        Esc + '(s12V' + Esc + '(s0S' + Esc + '(s0B' +
                        Esc + '(s4148T';
Schrift_LinePrinter  = Esc + '(10U' + Esc + '(s0P' + Esc + '(s10H' +
                        Esc + '(s12V' + Esc + '(s0S' + Esc + '(s0B' +
                        Esc + '(s0T';
Schrift_CGTimes      = Esc + '(10U' + Esc + '(s1P' + Esc + '(s10H' +
                        Esc + '(s12V' + Esc + '(s0S' + Esc + '(s0B' +
                        Esc + '(s4101T';
CursorPositionSave   = Esc + '&f0S';
CursorPositionBack   = Esc + '&f1S';

Probetext  = 'Das ist ein Probetext mmm iii. ';
Probetext1 = '1. ' + Probetext + 'HP-PCL-Demo ' + Programm;
Probetext2 = '2. ' + Probetext;
Probetext3 = '3. ' + Probetext;

var
  Lpt, LJ:   Char;
  yR:        Real;
  i, x, y:   Integer;
  iStr:       string[3];
  VMIStr,
  HMIStr,
  yZeile, T,
  xSpalte:   string[80];
  LaserJet3,
  LaserJet4: Boolean;

procedure WriteXY(Sp, Ze: Byte; Meldung: string);
begin
  GotoXY(Sp, Ze);
  Write(Meldung);
end;

procedure DruckerInfos(var Lpt, LJ: Char);
begin
  TextColor(Yellow); TextBackground(Blue);
  ClrScr;
  WriteXY(10, 3, '          PCL-Demo Nr 2: Schriften. HMI und VMI');
  WriteXY(10, 5, '          PCL/5-Drucker, z.B. HP-LaserJet');
  WriteXY(10, 6, '+-----' +
                        '-----+');
  WriteXY(10, 7, '| 1   An Parallelschnittstelle LPT1 |' +
                        ' 3   LaserJet III |');
  WriteXY(10, 8, '| 2   An Parallelschnittstelle LPT2 |' +
                        ' 4   LaserJet IV  |');
  WriteXY(10, 9, '| Esc Abbruch                |' +

```

[illegible]

```

WriteLn(Lst, Schrift_CGTimes, Probetext1);
WriteLn(Lst);

for i := 13 downto 5 do
begin
  Str(i, iStr);
  WriteLn(Lst, Esc + '(s' + iStr + 'V', Probetext2,
    ' Schriftgröße ', iStr, 'pt. ');
end;
WriteLn(Lst); WriteLn(Lst);

for i := 13 downto 5 do
begin
  Str(i, iStr);
  Str(i*0.7:4:1, VMIStr);
  WriteLn(Lst, Esc + '(s' + iStr + 'V',
    Esc + '&l' + VMIStr + 'C', Probetext3,
    ' Schriftgröße ', iStr, 'pt. VMI = ',
    VMIStr, '/120 Zoll');
end;

VMIStr := Esc + '&l8C'; { Standard für 8/48 Zoll = 6 Zeilen / Zoll }
WriteLn(Lst, VMIStr, Schrift_CGTimes); WriteLn(Lst);

WriteLn(Lst, '4. HMI = Horizontal Motion Index');
WriteLn(Lst, 'Proportionalschriften ignorieren HMI,');
WriteLn(Lst, 'deshalb für HMI Schrift ',
  Schrift_Courier, '"Courier"');
for i := 12 downto 5 do
begin
  Str(i, iStr);
  Write(Lst, Esc + '&k' + iStr + 'H');
  for x := 1 to 13 do
    Write(Lst, #254);
  for x := 1 to 13 do
    Write(Lst, #179);
  WriteLn(Lst);
end;
Write(Lst, Esc + '&k12H'); { Standard-HMI 12/120" = 10 cpi }
Write(Lst, Esc + '*p1300Y',
  Esc + '*p1250X');

WriteLn(Lst, Schrift_CGTimes, '5. VMI = Vertical Motion Index');
for i := 38 downto 0 do
begin
  Str(i/10:5:1, iStr);
  Write(Lst, Esc + '*p1250X', Esc + '&l' + iStr + 'C');
  for x := 1 to 7 do
    Write(Lst, #196);
  for x := 1 to 7 do
    Write(Lst, #254);
  WriteLn(Lst);
end;
Write(Lst, Esc + '&l8C'); { Normaler VMI = 8/48" = 6 Zeilen/Zoll }
Write(Lst, Esc + '*p300X', Esc + '*p1900Y',

```

```
        '6. Cursorposition speichern >|',
        CursorpositionSave);
WriteLn(Lst, Esc + '*p300X',
        Esc + '*p2000Y', '7. Rechtecke');
Write(Lst, Esc + '*p300X', Esc + '*p2030Y',
        Esc + '*c300A', Esc + '*c100B',
        Esc + '*c0P');
Write(Lst, Esc + '*p300X', Esc + '*p2170Y',
        Esc + '*c300A', Esc + '*c100B',
        Esc + '*c30G',
        Esc + '*c2P');
Write(Lst, Esc + '*p300X', Esc + '*p2310Y',
        Esc + '*c300A', Esc + '*c100B',
        Esc + '*c10G',
        Esc + '*c2P');
Write(Lst, CursorpositionBack,
        '< 8. Zurück zur gespeicherten Cursorposition');
WriteLn(Lst, Esc + '*p750X',
        Esc + '*p2400Y',
        Esc + '&a90P',
        '9. PCL-Drehung nur');
WriteLn(Lst, Esc + '*p955X', 'in 90°-Schritten',
        Esc + '&a0P'); { Dann wieder auf 0-Grad }
Write(Lst, Schrift_Courier,
        Esc + '*p000X', Esc + '*p000Y',
        '|<HalloXX',
        Schrift_CGTimes);
WriteLn(Lst, Esc + '*p950X',
        Esc + '*p2000Y',
        '10. Cursorpositionierung mit Zeilen/Spalten');
WriteLn(Lst, Esc + '*p950X',
        'Schrift Courier, geringere HMI und VMI');
Write(Lst, Schrift_Courier, Esc + '&k4H',
        Esc + '&l1C');
for i := 0 to 94 do
    begin
        y := Round(30*Sin(i/10));
        Str(95 + i, iStr);    xSpalte := Esc + '&a' + iStr + 'C';
        Str(365 - y, iStr);  yZeile := Esc + '&a' + iStr + 'R';
        WriteLn(Lst, xSpalte, yZeile, '');
    end;
Write(Lst, PCL_FF);
Close(Lst);
end.
```

```

program Pas25043; { PCL/5, HP-LaserJet, kha }

uses
  CRT, PRINTER;

const
  Programm = 'Pas25043.PAS';
  Esc      = #27;
  PCL_Reset = Esc + 'E';
  PCL_FF    = #12; { FF = Form Feed = Seitenauswurf }
  DPI300    = Esc + '*t300R';
  DPI600    = Esc + '*t600R';
  LptVor    = '1';
  LJVor     = '4';

type
  Farbtyp  = (Weiss, Schwarz);
  Linientyp = (Voll, Gepunktet);

var
  Lpt, LJ:   Char;
  i:         Integer;
  x0, y0,
  B, H:      Word;
  xR, yR:    Real;
  xStr,
  yStr:      string;
  Farbe:     Farbtyp;
  Linienart: Linientyp;
  LaserJet3,
  LaserJet4: Boolean;

procedure WriteXY(Sp, Ze: Byte; Meldung: string);
begin
  GotoXY(Sp, Ze);
  Write(Meldung);
end;

procedure DruckerInfos(var Lpt, LJ: Char);
begin
  TextColor(Yellow); TextBackground(Blue);
  ClrScr;
  WriteXY(10, 3, '                PCL-Demo Nr 3: Grafik');
  WriteXY(10, 5, '                PCL/5-Drucker, z.B. HP-LaserJet');
  WriteXY(10, 6, '+-----' +
                                '-----+');
  WriteXY(10, 7, '| 1  An Parallelschnittstelle LPT1 |' +
                                ' 3  LaserJet III |');
  WriteXY(10, 8, '| 2  An Parallelschnittstelle LPT2 |' +
                                ' 4  LaserJet IV  |');
  WriteXY(10, 9, '| Esc Abbruch                        |' +
                                ' Esc Abbruch      |');
  WriteXY(10, 10, '| -----' +
                                ' ----- |');
  WriteXY(10, 11, '| <<< Eingabe                                |' +
                                ' <<< Eingabe    |');
  WriteXY(10, 12, '+-----' +
                                '-----+');

```



[illegible]

```

end;
begin
  case Linienart of
    Voll:      Modulo := 1;
    Gepunktet: Modulo := 15;
  end;
  case Farbe of
    Schwarz: FarbStr := Esc + '*c0P';
    Weiss:   FarbStr := Esc + '*c1P';
  end;
  Str(Dicke, DickeStr);
  DickeStrA := Esc + '*c' + DickeStr + 'A';
  DickeStrB := Esc + '*c' + DickeStr + 'B';

  if Abs(y2 - y1) > Abs(x2 - x1)
  then begin
    xyGetauscht := True;
    Tausche(x1, y1, x2, y2);
    if x1 > x2
    then Tausche(x1, x2, y1, y2);
    end
  else begin
    xyGetauscht := False;
    if x1 > x2
    then Tausche(x1, x2, y1, y2);
    end;
  end;

  for x := x1 to x2 - Dicke do
    if (x mod Modulo = 0) then
      begin
        y := Round(y1 + (y2 - y1)/(x2 - x1) * (x - x1));
        Str(x, xStr); if not xyGetauscht
          then xStr := Esc + '*p' + xStr + 'X'
          else xStr := Esc + '*p' + xStr + 'Y';
        Str(y, yStr); if not xyGetauscht
          then yStr := Esc + '*p' + yStr + 'Y'
          else yStr := Esc + '*p' + yStr + 'X';
        Write(Lst, xStr, yStr,
              DickeStrA,
              DickeStrB,
              FarbStr);
      end;
    end;
  end;
begin
  DruckerInfos(Lpt, LJ);
  case LJ of
    '3': begin LaserJet3 := True; LaserJet4 := False; end;
    '4': begin LaserJet3 := False; LaserJet4 := True; end;
  end;
  case Lpt of
    '1': Assign(Lst, 'Lpt1');
    '2': Assign(Lst, 'Lpt2');
  end;
  Rewrite(Lst);

```

```

Write(Lst, PCL_Reset);
if LaserJet3 then Write(Lst, DPI300);
if LaserJet4 then Write(Lst, DPI600);

WriteLn(Lst);
WriteLn(Lst, '
                ' +
                ' HP-PCL-Demo "' + Programm + '"');

x0 := 300; B := 600;
y0 := 0; H := 400;
Linie(x0, y0, x0 + B, y0, 1, Schwarz, Voll);
Linie(x0 + B, y0, x0 + B, y0 + H, 1, Schwarz, Voll);
Linie(x0, y0 + H, x0 + B, y0 + H, 1, Schwarz, Voll);
Linie(x0, y0, x0, y0 + H, 1, Schwarz, Voll);
Linie(x0, y0, x0 + B, y0 + H, 11, Schwarz, Voll);
Linie(x0, y0, x0 + B, y0 + 100, 2, Schwarz, Gepunktet);
Linie(x0, y0, x0 + 100, y0 + H, 1, Schwarz, Gepunktet);
Linie(x0, y0, x0 + B, y0 + 200, 2, Schwarz, Voll);
Linie(x0, y0, x0 + 200, y0 + 400, 2, Schwarz, Voll);
Linie(x0, y0, x0 + B, y0 + 300, 3, Schwarz, Voll);
Linie(x0, y0, x0 + 300, y0 + 400, 3, Schwarz, Voll);
Linie(x0 + B, y0 + H, x0, y0, 2, Weiss, Voll);
Linie(x0, y0 + H, x0 + 600, y0, 2, Schwarz, Voll);
Linie(x0, y0 + H, x0 + 600, y0, 3, Schwarz, Voll);

for i := 0 to 8 do
    Linie(300*i, 0, 300*i, 12*300, 2, Schwarz, Gepunktet);
for i := 0 to 12 do
    Linie(0, 300*i, 13*300, 300*i, 2, Schwarz, Gepunktet);

for i := -2000 to 2000 do
    begin
        xR := i/10;
        yR := Exp(-Sqr(xR/100)) * Cos(xR/50*Pi);
        if (i mod 10) = 0
            then begin
                Str(Round(600 + 2*xR - 15), xStr);
                xStr := Esc + '*p' + xStr + 'X';
                Str(Round(900 - 250*yR + 14), yStr);
                yStr := Esc + '*p' + yStr + 'Y';
                Write(Lst, xStr, yStr, '');
            end
        else begin
            Str(Round(600 + 2*xR), xStr);
            xStr := Esc + '*p' + xStr + 'X';
            Str(Round(930 - 250*yR), yStr);
            yStr := Esc + '*p' + yStr + 'Y';
            Write(Lst, xStr, yStr,
                Esc + '*c1A',
                Esc + '*c1B',
                Esc + '*c0P');
            Str(Round(960 - 250*yR), yStr);
            yStr := Esc + '*p' + yStr + 'Y';
            Write(Lst, xStr, yStr,
                Esc + '*c2A',

```

```

        Esc + '*c2B',
        Esc + '*c0P');
    Str(Round(990 - 250*yR), yStr);
    yStr := Esc + '*p' + yStr + 'Y';
    Write(Lst, xStr, yStr,
        Esc + '*c3A',
        Esc + '*c3B',
        Esc + '*c0P');

    end;

end;

Write(Lst, PCL_FF); { Seitenauswurf }
Close(Lst);
end.

```

Das folgende Programm demonstriert die GL/2-Emulation:

```

program Pas25044; { Plottersprache HP-GL/2.          kha      }
{ HP = Hewlett-Packard, PCL = Page Control Language
  GL = Graphic Language, Plottersprache

  Hier Demonstration der HP-Plottersprache "HP-GL/2".

  HP-GL/2 ist eine vektororientierte Plottersprache, die von
  Hewlett-Packard für ihre Plotter entwickelt wurde und die
  aber auch bei den HP-Laserdruckern zur Verfügung steht.
  Die Plottersprache kann nur unter PCL aufgerufen werden;
  die Seitenausgabe ist nur nach Rückkehr zu PCL möglich.
  Die Seite kann PCL- und GL/2-Elemente in beliebiger Reihen-
  folge und Anordnung aufweisen. Eine Beschränkung ist nur
  durch den Speicher des Druckers gegeben.

  Allgemeines zu HP-GL/2:

  Der Nullpunkt liegt standardmäßig links unten. Die x-Koordi-
  nate läuft dann von links nach rechts, y von unten nach oben.
  Eine Plottereinheit "plu" (plot unit) beträgt in absoluten
  Koordinaten exakt 0.025 mm, d.h. 1 mm hat 40 plu. Es kann
  aber auch ein benutzerspezifisches Koordinatensystem benutzt
  werden, wie im folgenden Beispiel gezeigt wird.

  Im Gegensatz zu PCL bestehen die Befehle in GL/2 nicht aus
  Escape-Sequenzen, sondern aus einer mnemotechnischen Bezeich-
  nung mit zwei Buchstaben (Schreibweise groß/klein beliebig)
  und in den meisten Fällen mit anschließenden numerischen
  Parameter, die aber als Strings an den Drucker geschickt
  werden müssen. Die Parameter sind mit Komma voneinander zu
  trennen. Der komplette Befehl ist mit einem Semikolon abzu-
  schließen. Leerzeichen in der Befehlsfolge sind zulässig,
  aber nicht notwendig.

  Beispiele:      IN;
                  PD 25, 10, 35, 55;
                  PD25,10,35,55;

  Bei vielen Befehlen können die numerischen Parameter auch aus
  Realtypen entstanden sein, allerdings nicht in Gleitkomma-

```

schreibweise, sondern in Festkommaschreibweise mit maximal vier Nachkommastellen. Für die Konvertierung in Strings ist in Pascal die Standardprozedur "Str(.., ..)" zu benutzen.

Beispiel: Str(y:5:1, yStr);  
Darin sei "y" ein numerischer Gleitkommaausdruck, der mit einer Nachkommastelle in die Stringvariable "yStr" konvertiert wird.

Das folgende Beispiel kann nur einen kleinen Einblick in HP-GL/2 gewähren. Für weitere Informationen wird auf das "Technical Reference Manual" von Hewlett-Packard verwiesen, das aber nicht im Lieferumfang des Druckers enthalten ist. Das HP-Benutzerhandbuch enthält keine Informationen über HP-GL/2.

}

**uses**

CRT, PRINTER;

**const**

```

Programm      = 'Pas25044.PAS';
LptVor        = '1'; { '1' für Lpt1, '2' für Lpt2 }
LJVor         = '4'; { '3' für LJ 3, '4' für LJ4 }
Esc = #27;
PCL_Reset     = Esc + 'E'; { Nicht "Reset", da Standardprozedur }
PCL_FF        = #12;      { FF = Form Feed = Seitenauswurf }
LinkerRand10  = Esc + '&a10L';
DPI300        = Esc + '*t300R';
DPI600        = Esc + '*t600R';
GL2_Beginn    = Esc + '%0B';
GL2_Init      = 'IN;';
PCL_Beginn    = Esc + '%0A';
Schrift_CGTimes = Esc + '(10U' + Esc + '(s1P' + Esc + '(s10H' +
                    Esc + '(s12V' + Esc + '(s0S' + Esc + '(s0B' +
                    Esc + '(s4101T';

```

**var**

```

i:           Integer;
y:           Real;
Lpt, LJ:     Char;
iStr, yStr,
GL_Befehl:   string;
LaserJet3,
LaserJet4:   Boolean;

```

**procedure** WriteXY(Sp, Ze: Byte; Meldung: string);

**begin**

```

  GotoXY(Sp, Ze);
  Write(Meldung);

```

**end;**

**procedure** DruckerInfos(var Lpt, LJ: Char);

**begin**

```

  TextColor(Yellow); TextBackground(Blue);
  ClrScr;
  WriteXY(10, 3, '                               PCL-Demo Nr 4: PCL und GL/2          ');

```



```

Write(Lst, 'Hallo, hier ist der HP-LaserJet ');
if LaserJet3 then Write(Lst, 'III, ');
if LaserJet4 then Write(Lst, 'IV, ');
WriteLn(Lst, 'zuerst im PCL-Modus.');
```

Write(Lst, Esc + '*c2880X',	Esc + '*c2880Y',	{ PCL-Rahmen 4"*4" }
	{ 4*720 = 2880, 1" = 720 Dezipunkte }	
	{ 1 Zoll = 720 Dezipunkte }	
Esc + '*p300X',	Esc + '*p150Y',	{ Für PCL-Ankerpunkt }
Esc + '*c0T');		{ Ankerpunkt speichern }
		{ "*p##": 1/300 Zoll }

```

{ Jetzt umschalten auf GL/2-Modus: }
Write(Lst, GL2_Beginn, GL2_Init);
Write(Lst, 'SP 1;', { Select Pen 1. Beim LaserJet nur 1 }
'SC 0, 100, 0, 100;', { Set User Scaling mit x1, y1,
{ und x2, y2 }
'PW 0.8;', { Pen Width 0.8 mm }
'PD 100,0, 100,100, 0,100, 0,0;', { Pen Down, zeichne }
'PU 10, 95;', { Pen Up und bewege zum Punkt (x,y) }
'DT@;', { Definiere "@" als Text-Terminator }
'LBHallo, hier ist HP-GL/2@;',
'PU 10, 90;',
{ LB = Label Printing, mit Terminator }
'LBStandardschrift bei HP-GL/2@;',
{ SD = Standard Font Definition }
{ +--- 10pt }
'SD 1,21, 2,1, 4,10, 5,0, 6,3, 7,52;',
'SS;', { Select Standard Font }
'PU 10, 85;',
'LBAndere Schriften in GL/2: z.B. Univers fett 10pt@;',
'DI 1.0, 0.4;', { Direction, Tangens = 0.4/1.0 }
'PU 75, 65;',
'LBSchriftdrehung@;',
'PW 0.15;', { Pen Width 0.15 mm }
'PU 50, 50;',
'CI 25;', { Circle mit Radius 25 }
{ LT = Line Type }
{ Die eng gepunktete Linie "LT 0;" }
{ bricht bei kleinen Strichdicke aus }
'LT; PU 10, 4; PD 10, 4, 90, 4;',
'LT 0; PU 10, 6; PD 10, 6, 90, 6;',
'LT 1; PU 10, 8; PD 10, 8, 90, 8;',
'LT 2; PU 10, 10; PD 10, 10, 90, 10;',
'LT 3; PU 10, 12; PD 10, 12, 90, 12;',
'LT 4; PU 10, 14; PD 10, 14, 90, 14;',
'LT 5; PU 10, 16; PD 10, 16, 90, 16;',
'LT 6; PU 10, 18; PD 10, 18, 90, 18;',
'SC -50, 50;', { Set User Scaling mit x1, y1 }
'-50, 50;', { und x2, y2 }
'1;', { "1" bedeutet isotropes Scaling }
'PU 0, 0;', { Pen Up. Hier Cursor zu 0, 0 }
'CI 20;', { "CI r" = Circle mit Radius r }
'PU 0, 0;', { Pen Up }
```

```

        'PM 0;',          { Polygon Mode "0": aktivieren      }
        'PD -5, -10, 5, -10, 0, 0;', { Pen Down          }
        'PM 2;',          { Polygon Mode "2": schließen      }
        'FT 10, 25;',     { Fill Type: "10"=Raster, "25": F=25% }
        'FP;',            { Fill Polygon                    }
        'PU 0, 0; LT;',   { Zum Mittelpunkt, Standard-Linientyp }
        'PD -5, -10, 5, -10, 0, 0;', { Polygonrand nachzeichnen }
        'PU 35, -20;',    { Pen Up                          }
        'PM 0;',          { Polygon Mode aktivieren          }
        'CI -5;',         { Circle, r = 5, "-5": neg. Drehricht. }
        'PU 30, -15;',    { Pen Up                          }
        'CI 7;',          { Circle, r = 7, posit. Drehrichtung }
        'PM 2;',          { Polygon Mode beenden            }
        'FT 1;',          { Fill Type: "1": Vollton          }
        'FP;',            { Fill Polygon                    }
    );

Write(Lst, 'LT;',
      'PU -40, 0;',
      'PD 40, 0;');

if LaserJet3 then
    for i := -40 to 40 do
        begin
            Str(i, iStr);
            y := 25 * Exp(-Sqr(i/25)) * Cos(i/10*Pi);
            Str(y:6:1, yStr);
            if i = -40
                then GL_Befehl := 'PU'
                else GL_Befehl := 'PD';
            GL_Befehl := GL_Befehl + iStr + ',' + yStr + ';';
            Write(Lst, GL_Befehl);
        end;

if LaserJet4 then
    for i := -80 to 80 do
        begin
            Str(i/2:6:1, iStr);
            y := 25 * Exp(-Sqr(i/50)) * Cos(i/20*Pi);
            Str(y:6:1, yStr);
            if i = -80
                then GL_Befehl := 'PU'
                else GL_Befehl := 'PD';
            GL_Befehl := GL_Befehl + iStr + ',' + yStr + ';';
            Write(Lst, GL_Befehl);
        end;

{ ----- Umschalten auf PCL ----- }
Write(Lst, PCL_Beginn);
Write(Lst, Esc + '*p300X', Esc + '*p1500Y');
WriteLn(Lst, 'Hallo, hier ist wieder der PCL-Modus. ');
WriteLn(Lst, 'PCL und GL/2 sind kryptischer als PostScript. ');
WriteLn(Lst, 'meint der Autor. ');

Write(Lst, PCL_Reset); { Reset wirft auch eine bedruckte Seite aus }
Close(Lst);

```



**end.**

```

program Pas25045; { Dr. K. Haller, FHM, DR }
  { HP = Hewlett-Packard, PCL = Page Control Language }

uses
  CRT, PRINTER;

const
  Programm          = 'Pas25045.PAS';
  LptVor            = '1'; { '1' für Lpt1, '2' für Lpt2 }
  LJVor            = '4'; { '3' für LJ 3, '4' für LJ4 }
  Esc = #27;
  PCL_Reset        = Esc + 'E'; { Nicht "Reset", da Standardprozedur }
  PCL_FF           = #12;      { FF = Form Feed = Seitenauswurf }
  LinkerRand10     = Esc + '&a10L';
  DPI075           = Esc + '*t075R'; { 75, 100, 150, 200, 300, (600) }
  DPI100           = Esc + '*t100R';
  DPI150           = Esc + '*t150R';
  DPI200           = Esc + '*t200R';
  DPI300           = Esc + '*t300R';
  DPI600           = Esc + '*t600R'; { 600 dpi nur ab LaserJet IV }
  Schrift_CGTimes = Esc + '(10U' + Esc + '(s1P' + Esc + '(s10H' +
                        Esc + '(s12V' + Esc + '(s0S' + Esc + '(s0B' +
                        Esc + '(s4101T';

var
  Lpt, LJ:      Char;
  LaserJet3,
  LaserJet4:   Boolean;
  Ch:          Char;
  Zeile:       string;
  AW, EW:      Byte;
  i, iMax,
  j, jMax:     Word;
  jMaxStr:     string[5];
  DPI,
  xK, yK:      string[10];
  Lauf,
  LaufMax:     Byte;

procedure WriteXY(Sp, Ze: Byte; Meldung: string);
begin
  GotoXY(Sp, Ze);
  Write(Meldung);
end;

procedure DruckerInfos(var Lpt, LJ: Char);
begin
  TextColor(Yellow); TextBackground(Blue);
  ClrScr;
  WriteXY(10, 3, '                PCL-Demo Nr 5: Rastergrafik');
  WriteXY(10, 5, '                PCL-Drucker mit GL/2, z.B. HP-LaserJet');
  WriteXY(10, 6, '+-----' +
                        '-----+');
  WriteXY(10, 7, '| 1   An Parallelschnittstelle LPT1 |' +

```



```

for Lauf := 1 to LaufMax do
  begin
    case Lauf of
      1: begin DPI := DPI075; yK := Esc + '*p0300Y'; end;
      2: begin DPI := DPI100; yK := Esc + '*p0800Y'; end;
      3: begin DPI := DPI150; yK := Esc + '*p1200Y'; end;
      4: begin DPI := DPI200; yK := Esc + '*p1500Y'; end;
      5: begin DPI := DPI300; yK := Esc + '*p1800Y'; end;
      6: begin DPI := DPI600; yK := Esc + '*p2100Y'; end;
    end;
    Write(Lst, xK,
          yK,
          DPI,
          Esc + '*r0F',
          Esc + '*rlA');
    for i := 1 to iMax do
      begin
        Zeile := '';
        for j := 1 to jMax do
          begin
            Ch := Chr(AW + Random(EW - AW + 1));
            Zeile := Zeile + Ch;
          end;
        Write(Lst, Esc + '*b' + jMaxStr + 'W' + Zeile);
      end;
    Write(Lst, Esc + '*rB'); { Ende Rastergrafik }
  end;

  Write(Lst, PCL_Reset); { Reset wirft eine bedruckte Seite aus, }
  Close(Lst);
end.

```

```

program Pas25046; { Kap. 25, Drucker HP-LaserJet, PCL-5 }
                  { Rechtecke, Füllungen und Zeichenfläche }
                  { Dr. K. Haller, FH München, Stg DR }

uses
  PRINTER;

const
  Programm = 'Pas25046.PAS';
  x0      = 0.0; { Nullpunkt X }
  y0      = 0.0; { Nullpunkt Y }
  xMax    = 197.8; { Nutzbare Breite in mm }
  yMax    = 279.9; { Nutzbare Höhe in mm }
  KL      = 2.54; { Kantenlänge kleines Quadrat }
  yNegT   = -25.4/6; { = -4.23 mm, Text }
  yNegG   = yNegT - 25.4/10; { = -6.77 mm, Grafik }

var
  i, Phi: Byte;
  xMaxStr,
  yMaxStr,
  yNegTStr,
  yNegGStr: string;

```

```
function MoveXmm(x: Real): string;
var
  xStr: string;
begin
  x := x/25.4*720; { In "decipoints = 1/720 inch }
  Str(x:7:2, xStr); { Bis 2 Dezimalstellen }
  MoveXmm := #27 + '&a' + xStr + 'H';
end;

function MoveYmm(y: Real): string;
var
  yStr: string;
begin
  y := y/25.4*720; { In "decipoints = 1/720 inch }
  Str(y:7:2, yStr); { Bis 2 Dezimalstellen }
  MoveYmm := #27 + '&a ' + yStr + 'V';
end;

function KastenXmm(x: Real): string;
var
  xStr: string;
begin
  x := x/25.4*720; { In "decipoints = 1/720 inch }
  Str(x:7:2, xStr); { Bis 2 Dezimalstellen }
  KastenXmm := #27 + '*c' + xStr + 'H';
end;

function KastenYmm(y: Real): string;
var
  yStr: string;
begin
  y := y/25.4*720; { In "decipoints = 1/720 inch }
  Str(y:7:2, yStr); { Bis 2 Dezimalstellen }
  KastenYmm := #27 + '*c' + yStr + 'V';
end;

function FuellenSchraffur(Muster: Byte): string;
  { Schraffurmuster: 1 waagrecht, 2 senkrecht, 3 rechts, }
var { 4 links, 5 Kreuz0Grad, 6 Kreuz45Grad }
  MusterStr: string[1];
begin
  Str(Muster, MusterStr);
  FuellenSchraffur := #27 + '*c' + MusterStr + 'G' +
    #27 + '*c3P';
end;

function FuellenRaster(Phi: Byte): string;
  { Phi: 1... 100. Ausgefuehrt werden aber nur 8 Stufen }
var
  PhiStr: string;
begin
  Str(Phi, PhiStr);
  FuellenRaster := #27 + '*c' + PhiStr + 'G' +
    #27 + '*c2P';
end;
```

```

begin
  Write(Lst, #27 + 'E'); { Drucker-Reset }
  Write(Lst, MoveXmm(x0), MoveYmm(y0),
    'Nullpunkt-Standard, x0 = 0.00 mm, y0 = 0.00 mm',
    KastenXmm(KL), KastenYmm(KL),
    MoveXmm(x0), MoveYmm(y0),
    FuellenRaster(100));
  Write(Lst, MoveXmm(30.0), MoveYmm(70.0),
    'Wenn bei Cursorbewegungen negative Werte entstehen');
  Write(Lst, MoveXmm(30.0), MoveYmm(75.0),
    'oder bei Konstanten ein Plus- oder Minuszeichen');
  Write(Lst, MoveXmm(30.0), MoveYmm(80.0),
    'angegeben ist, dann werden statt absoluten Cursor-');
  Write(Lst, MoveXmm(30.0), MoveYmm(85.0),
    'bewegungen relative Cursorbewegungen ausgeführt. ');
  Write(Lst, MoveXmm(30.0), MoveYmm(93.0),
    'Im Bedarfsfall ist der Cursor auf Null zu setzen. ');
  Str(yNegT:4:2, yNegTStr);
  Str(yNegG:4:2, yNegGStr);
  Write(Lst, MoveXmm(x0), MoveYmm(y0), { Cursor auf Null ! }
    MoveXmm(x0), MoveYmm(y0 + yNegT),
    'Nullpunkt-Minimum, x0 = 0.00 mm, y0 = ', yNegTStr,
    ' mm (Text), y0 = ', yNegGStr, ' mm (Grafik)',
    MoveXmm(x0), MoveYmm(y0),
    MoveXmm(x0), MoveYmm(y0 + yNegG),
    KastenXmm(KL), KastenYmm(KL),
    FuellenRaster(100));
  Write(Lst, MoveXmm(x0), MoveYmm(y0),
    MoveXmm(x0), MoveYmm(y0 + yNegG),
    KastenXmm(xMax), KastenYmm(0.1),
    FuellenRaster(100));
  Write(Lst, MoveXmm(0.0), MoveYmm(0.0),
    KastenXmm(210.0), KastenYmm(0.1), { "210.0" ist zu viel }
    FuellenRaster(100)); { wird aber ignoriert }
  Write(Lst, MoveXmm(0.0), MoveYmm(0.0),
    KastenXmm(0.1), KastenYmm(297.0), { "297.0" ist zu viel }
    FuellenRaster(100)); { wird aber ignoriert }

  Str(xMax:5:1, xMaxStr);
  Str(yMax:5:1, yMaxStr);

  Write(Lst, MoveXmm(0.0), MoveYmm(yMax),
    'yMax = ' + yMaxStr + ' mm, ' +
    'xMax = ' + xMaxStr + ' mm ' +
    '(y für Unterlängen zu tief!)' +
    KastenXmm(xMax), KastenYmm(0.1),
    FuellenRaster(100));
  Write(Lst, MoveXmm(xMax), MoveYmm(0.0),
    KastenXmm(0.1), KastenYmm(yMax),
    FuellenRaster(100));
  Write(Lst, MoveXmm(xMax - KL), MoveYmm(yMax - KL),
    KastenXmm(KL), KastenYmm(KL),
    FuellenRaster(100));
  for i := 1 to 6 do

```

```
Write(Lst, MoveXmm(10.0 + i*20.0), MoveYmm(5.0),  
      KastenXmm(15.0),      KastenYmm(15.0),  
      FuellenSchraffur(i));  
for i := 1 to 8 do  
  begin  
    case i of  
      1: Phi := 2; { Nur ganzzahlige Werte }  
      2: Phi := 7; { ca.-Mittelwerte der Spannweite }  
      3: Phi := 15;  
      4: Phi := 28;  
      5: Phi := 46;  
      6: Phi := 68;  
      7: Phi := 90;  
      8: Phi := 100;  
    end;  
    Write(Lst, MoveXmm(10.0 + i*20.0), MoveYmm(25.0),  
          KastenXmm(15.0),      KastenYmm(15.0),  
          FuellenRaster(Phi));  
  end;  
  
Write(Lst, MoveXmm(30.0), MoveYmm(130.0),  
      'Demo-Programm "' + Programm + '" für HP-LaserJet. kha');  
Write(Lst, MoveXmm(30.0), MoveYmm(140.0),  
      'Rechtecke, Standard-Füllungen und Zeichenfläche');  
  
Write(Lst, #12); { Seitenauswurf }  
end.
```