

## Kapitola 5

# Několik příkladů v Turbo Pascalu

V literatuře [3] se student mohl seznámit s podprogramy standardních jednotek Turbo Pascalu. V následujících dvou částech uvedeme ukázkou několika příkladů využívajících podprogramy standardní jednotky **Crt** pro práci s obrazovkou a standardní jednotky **Graph** pro práci s grafikou.

### 5.1 Použití podprogramů pro práci s obrazovkou

V jednotce **Crt** se nachází řada potřebných podprogramů pro práci s obrazovkou (a nejen s ní). Chceme-li pracovat s podprogramy jednotky CRT, pak na začátku části definic a deklarací musíme uvést odstavec **uses** s identifikátorem **Crt**.

Hodláme-li pak změnit barvu pozadí, na které budeme psát text, pak příkazem např.

`TextBackGround(blue)` resp. `TextBackGround(black)` nastavíme barvu pro pozadí, v našem případě modrou resp. černou. Barva se však neobjeví, dokud *něco* na obrazovku nenapišeme. Nastavíme-li barvu ke psaní textu příkazem např.

`TextColor(red)` resp. `TextColor(green)`, pak příkaz tisku např. `Writeln('Ahoj')` způsobí tisk červeného *Ahoj* na modré pozadí, resp. zeleného *Ahoj* na černé pozadí. Celý zbytek obrazovky zůstává nezměněn. Hodláme-li „vymodřit“ celou obrazovku pro případný další tisk, pak tak učiníme dvojicí příkazů

`TextBackGround(blue); ClrScr;` Po těchto příkazech je obrazovka prázdná (Clear Screen) a modrá. Hodláme-li nyní naše červené *Ahoj* posadit asi doprostřed obrazovky, pak tak můžeme učinit příkazy `TextColor(red); GoToXY(38,12); Write('Ahoj');` za předpokladu, že pracujeme ve standardním textovém režimu 25 řádků po 80 sloupcích.

Chceme-li náš tisk doprovodit akustickým signálem, použijeme příkazy `Sound(440); Delay(1000); NoSound;` které z počítače vyloučí tón *komorní a* (440Hz) po dobu 1s (1000ms). Pozor – opomenutí příkazu `NoSound` vede k neustálému „pískání“ počítače. Příkaz `Delay` slouží k pozastavení vykonávání dalších příkazů počítačem. Bez jeho přítomnosti ve výše uvedené trojici příkazů by byl generátor zvuku vypnut bezprostředně po jeho zapnutí tak rychle, že by uživatel žádný zvuk neslyšel.

Velmi potřebná je funkce typu `char ReadKey`, poskytující hodnotu znaku stisknutého na klávesnici. Za předpokladu, že proměnná `Z` je typu `char` ji můžeme použít v příkazu

`Z := ReadKey;` což vede ke stejnému efektu jako `Read(Z)` s tím rozdílem, že na příkaz `Read(Z)` musíme odpovědět stisknutím příslušné klávesy, např. `A` a tento potvrdit klávesou `enter` (poté je v proměnné `Z` znak 'A'), zatímco na příkaz `Z := ReadKey` odpovíme pouze stisknutím klávesy `A`.

Algoritmizaci opakovaně vyžadované odpovědi na posledním řádku obrazovky tak dlouho, dokud správně neodpovíme, můžeme provést např. ve tvaru:

```
repeat GoToXY(30,25);          { * přechod doprostřed posledního řádku obrazovky * }
      ClrEol;                  { * vymazání od nastaveného cursoru do konce řádku * }
      Write('Chceš pokračovat? Odpověz A)no - N)e:');
      Z := ReadKey;            { * možno použít Z:=UpCase(Z) * }
until Z in ['A','a','N','n'] { * a pak postačuje psát Z in ['A','N'] * }
```

Uvedený algoritmus by se ovšem obešel i bez deklarované proměnné `Z` (ta však v praktických aplikacích bývá potřebná, neboť je zpravidla dále testována). Jeho poslední dva řádky můžeme nahradit jedním ve tvaru

until ReadKey in ['A','a','N','n'] popř. until UpCase(ReadKey) in ['A','N']

## 5.2 Použití podprogramů pro práci s grafikou

Ve všech dále uvedených příkladech je pracováno jak s jednotkou Crt, tak s jednotkou Graph. Zvolené ukázky mají především podnítit zájem studentů o grafické aplikace. Použití jednotlivých podprogramů není zde rozebíráno. Syntaxe jejich použití je pro studenta FAST dostatečně popsána v [3].

**Příklad 5.2.1:** Vykreslení průběhu funkce  $y=\sin(x)$  v intervalu  $\langle -2\pi, 2\pi \rangle$ .

```

Program GRAF;                                     { * vykreslení y = sin(x) * }
  uses Graph,Crt;                                 { * připojení knihoven   * }
  var   I, GD, GM, Y1, Y2, POLX, POLY :integer;
        K1, K2, X, SIRKA, VYSKA      :real;
  begin SIRKA:=4*pi;                               { * X je od -2pi do +2pi * }
        VYSKA:=2;                                 { * Y v intervalu <-1,1> * }
        DetectGraph(GD,GM);
        InitGraph(GD,GM,'u:\compil\tp_5.5\graph');
        POLX:=GetMaxX div 2;
        POLY:=GetMaxY div 2;
        line(0,POLY,GetMaxX,POLY);               { * vykreslení osy x     * }
        line(POLX,0,POLX,GetMaxY);               { * vykreslení osy y     * }
        OutTextXY(GetMaxX-10,POLY+3,'x');        { * popis osy x         * }
        OutTextXY(POLX+5,1,'y');                 { * popis osy y         * }
        K1:=SIRKA/(GetMaxX+1);                   { * koeficienty pro převod * }
        K2:=(GetMaxY+1)/VYSKA;                  { * pixlu I na radiány X a * }
        SetColor(lightred);                       { * výsledku Y na pixly  * }
        for I:=0 to GetMaxX do
          begin X:=(I-POLX)*K1;
                putpixel(I,POLY-trunc(sin(X)*K2),lightblue);
          end;                                     { * vystup (zobrazení) bodu * }
        OutTextXY(20,10,'y=sin(x)');
        repeat until KeyPressed;
        closegraph;
  end.

```

\_\_\_\_\_ konec Příkladu 5.2.1 \_\_\_\_\_

\_\_\_\_\_ konec Příkladu 5.2.1 \_\_\_\_\_

**Příklad 5.2.2:** Ukázka nastavení výplňového vzoru a barvy ohraničené oblasti na úloze vykreslení terče.

```

Program TERC;
  uses Crt, Graph;
  var   GD, GM, R :integer;
  begin writeln('Zadej poloměr největší kružnice (maximálně 225): '); readln(R);
        DetectGraph(GD,GM);   InitGraph(GD,GM,'u:\compil\tp_5.5\graph');
        SetBkColor(Brown);    SetColor(Lightgreen);
        circle((GetMaxX-1) div 2, (GetMaxY-30) div 2,R);
        R:= R div 2;
        circle((GetMaxX-1) div 2, (GetMaxY-30) div 2,R);
        SetFillStyle(1,Lightred);
        FloodFill( (GetMaxX-1) div 2, (GetMaxY-30) div 2+R+2,Lightgreen);
        R := R div 2;
        circle((GetMaxX-1) div 2, (GetMaxY-30) div 2,R);
        SetFillStyle(10,LightBlue);
        FloodFill( (GetMaxX-1) div 2, (GetMaxY-30) div 2+R+2,Lightgreen);
        SetFillStyle(11,yellow);
        FloodFill( (GetMaxX-1) div 2, (GetMaxY-30) div 2 ,Lightgreen);
        OutTextXY(10,GetMaxY-10,' T E R C - ukazují pouze 5 sekund');
  end.

```

```

    delay(5000);
    closeGraph
end.

```

---

konec Příkladu 5.2.2

konec Příkladu 5.2.2

---

**Příklad 5.2.3:** Ukázka animace (pohybu) obrazce na úloze pohybu míčku v ohraničeném prostoru.

```

Program MIC;                                     { * ukázka animace          * }
uses Crt, Graph;                                 { * (pohyb míčku)         * }
var GD, GM, X, Y, DX, DY, R, COLORR : integer;
    VLEVO, VPRAVO, NAHORU, DOLU : integer;
Procedure INICIALIZACE;                          { * pro globální proměnné * }
begin
    VLEVO := 100;      VPRAVO := 500;
    NAHORU := 100;     DOLU := 300;
    R := 20;          COLORR := green;
    X := 200;         Y := 200;
    DX := 5;         DY := 5;
end;
Procedure KROK(var X, Y, DX, DY : integer);
begin
    if ((X+R+DX) >= VPRAVO) or ((X-R+DX) <= VLEVO) then DX := -DX;
    if ((Y-R+DY) <= NAHORU) or ((Y+R+DY) >= DOLU) then DY := -DY;
    SetColor(GetBkColor); Circle(X, Y, R);
    X := X + DX;          Y := Y + DY;
    SetColor(COLORR);    Circle(X, Y, R);
end;
Begin INICIALIZACE;
    DetectGraph(GD, GM);    InitGraph(GD, GM, 'u:\compil\tp_5.5\graph');
    SetColor(red);
    Rectangle(VLEVO, NAHORU, VPRAVO, DOLU);
    repeat KROK(X, Y, DX, DY);
    until Keypressed;
    CloseGraph
end.

```

---

konec Příkladu 5.2.3

konec Příkladu 5.2.3

---

**Příklad 5.2.4:** Ukázka vykreslování sloupcového a kruhového histogramu.

Dále uvedeny programy HISTOGRAM a KOLAC vedou k vykreslování sloupcového a kruhového histogramu. Praktická využitelnost je čtenáři jistě známa (vyhodnocování výsledků voleb apod.).

```

Program HISTOGRAM;
uses Graph, Crt;
var GD, GM, X1, Y1, X2, Y2 : integer;
    PHJ, D, I : word;
    P : array[1..31] of real;
    TEXT : array[1..31] of string[5];
    MAX : real;
Begin ClrScr;
    Write('Zadej pocet hodnocenych jednotek:'); ReadLn(PHJ);
    I:=1; MAX:=0;
    while I<=PHJ do
    begin
        Write('Vysledky ', I, '. jednotky v %: '); ReadLn(P[I]);
        Write('Nazev ', I, '. jednotky [max. 5 znaku]:'); ReadLn(TEXT[I]);
        if P[I]>MAX then MAX:=P[I];
        I:=I+1;
    end;
    DetectGraph(GD, GM);    InitGraph(GD, GM, 'u:\compil\tp_5.5\graph');
    ClearDevice;

```

```

D:= 50; {trunc(GetMaxX/PHJ) }
X1:=0; Y1:= GetMaxY; I:=1;
while I<=PHJ do
begin X2:=X1+D;
      Y2:=trunc(GetMaxY*(1-P[I]/100)); { (1-P[I]/MAX) }
      SetFillStyle(1,I);
      Bar3D(X1,Y1,X2,Y2,10,true);
      OutTextXY(X2-trunc(D/2),Y2-25,TEXT[I]);
      I:=I+1; X1:=X2;
end;
while not KeyPressed do;
CloseGraph;
end.

```

Program KOLAC;

```

uses Graph, Crt;
var GD, GM, X, Y      : integer;
    PHJ,D,I, R, U1, U2 : word;
    P      : array[1..31] of real;
    TEXT   : array[1..31] of string[5];
    MAX, S : real;
Begin ClrScr;
  Write('Zadej počet hodnocených jednotek:');ReadLn(PHJ);
  I:=1; MAX:=0;
  while I<=PHJ do
begin Write('Výsledky ',I,'. jednotky v %: ');      ReadLn(P[I]);
      Write('Název ',I,'. jednotky [max. 5 znaku]:'); ReadLn(TEXT[I]);
      if P[I]>MAX then MAX:=P[I];
      I:=I+1;
end;
S:=0;          { * Převod na %, pakliže není zadání do součtu 100 * }
for I:=1 to PHJ do S:=S+P[I];
for I:=1 to PHJ do P[I]:=100*P[I]/S;
DetectGraph(GD,GM);      InitGraph(GD,GM,'U:\compil\tp_5.5\graph');
ClearDevice;
SetColor(0);
X:=(GetMaxX-1) div 2; Y:= (GetMaxY-30) div 2; I:=1;
R:=100; U1:=0; line(X,Y,X+R,Y);
while I<=PHJ do
begin SetFillStyle(1,I);
      if I=PHJ then U2:=360
      else U2:=U1 + trunc(360*P[I]/100);
      Sector(X,Y,U1,U2,R,R);
      Bar((I-1)*70,GetMaxY-50,(I-1)*70+50,GetMaxY);
      OutTextXY((I-1)*70+10,GetMaxY-25,TEXT[I]);
      I:=I+1; U1:=U2;
end;
{ * Porovnání procedury Sector a procedury Arc
sector(100,100,0,30,100,100);
line(300,300,400,300); arc(300,300,0,30,100);
line(300+round(100*cos(PI*6)),300-round(100*sin(PI/6)),300,300); * }
while not KeyPressed do;
CloseGraph;
end.

```

# Literatura

- [1] Wirth, N.: *Algoritmus + Data Structures = Program*. Prentice-Hall, Englewood Cliffs, New York, USA 1973
- [2] Jensen, K.—Wirth, N.: *Pascal—User Manual and Report*. Springer Verlag, New York 1974
- [3] Motyčka, A.—Novotná, H.: *Výpočetní technika a algoritmizace (Vybrané pasáže z Turbo Pascalu)*. VUT FAST, Brno, 1992
- [4] Rábová, Z.—Češka, M.—Honzík, J.M.—Hruška, T.: *Počítače a programování*. ES VUT Brno, 1980
- [5] Bajgar, L.—Tyc, O.: *Základy programování. PASCAL. Průvodce Turbo-Pascalem*. ES VUT Brno, 1992
- [6] Rybička, J.: *Programové vybavení počítačů (algoritmizace)*. ES VŠZ v Brně, 1992
- [7] Jinoch, J.—Müller, K.—Vogel, J.: *Programování v jazyku Pascal*. SNTL Praha, 1985
- [8] Cimrman, J.: *Počátkové programování v Čechách a na Moravě. VN v Liptákově, někdy před sto lety*
- [9] Honzík, J.: *Programovací techniky*. ES VUT v Brně, 1985
- [10] Motyčka, A.—Rybička, J.: *Výpočetní technika a algoritmizace (Ovládání prostředí Turbo – Pascalu)*. VUT FAST, Brno, 1992
- [11] Hruška, T.: *Pascal pro začátečníky*. SNTL Praha, 1989
- [12] Erbes, J.—Motyčka, A.—Motyčková, L.: *Úvod do programování v jazyce Pascal*. ES VŠZ Brno, 1986