

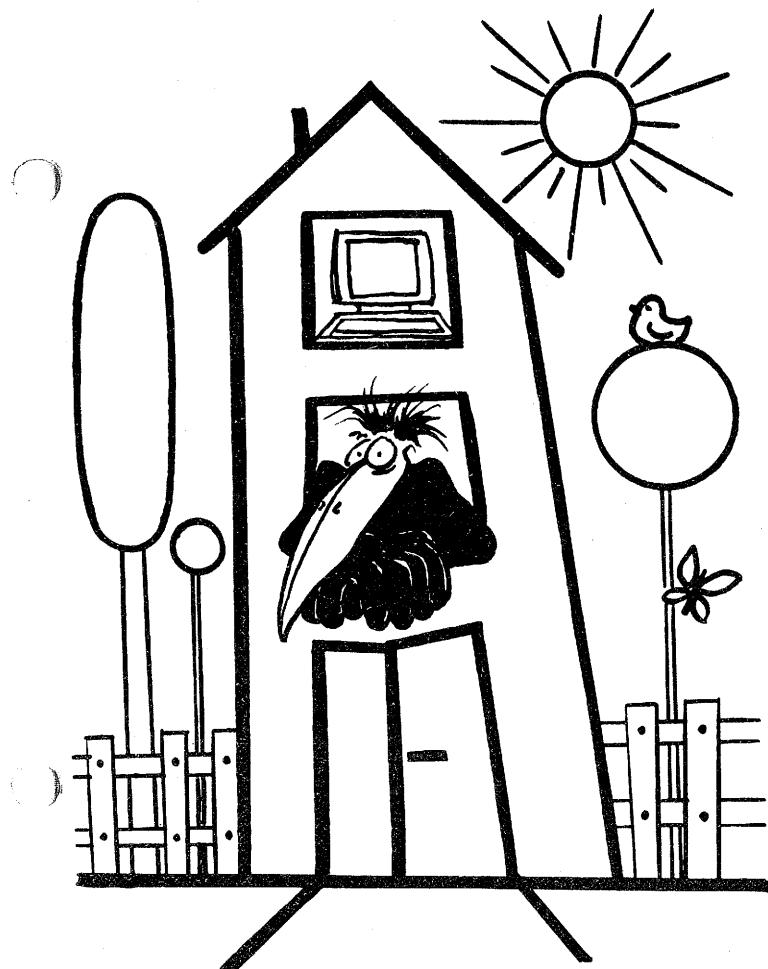
Obsah

1. Jednoduché obrázky	5
1.1. Grafika příkazu PRINT	7
1.2. Koncepce bodové grafiky	13
1.3. Vynášení bodů	17
1.4. Kreslení přímek	21
1.5. Kreslení obrázku a barvy	27
1.5.1 Barvy	27
1.5.2 Kreslení obrázku	29
2. Jednoduché grafy	45
2.1. Základy: grafy vývoje dat	46
2.1.1 Metoda příkazu print	47
2.1.2 Metoda bodové grafiky	51
2.2. Grafy s popisem	54
2.3. Sloupcové diagramy - barvy a stínování	59
3. Křivky	69
3.1. Kružnice	71
3.1.1 Algoritmus pro kreslení kružnice z přímkových úseků	73
3.1.2 Algoritmus kreslení kružnice vynášením bodů	75
3.2. Ostatní křivky	84
3.2.1 Eliptické křivky	84
3.2.2 Sinusové křivky	85
3.2.3 Polynomiální křivky	88
3.2.4 Normální křivky	92
3.3. Obrázky s křivkami	96
3.4. Grafy a kruhové diagramy	104
3.4.1 Grafy	104
3.4.2 Kruhové diagramy	107

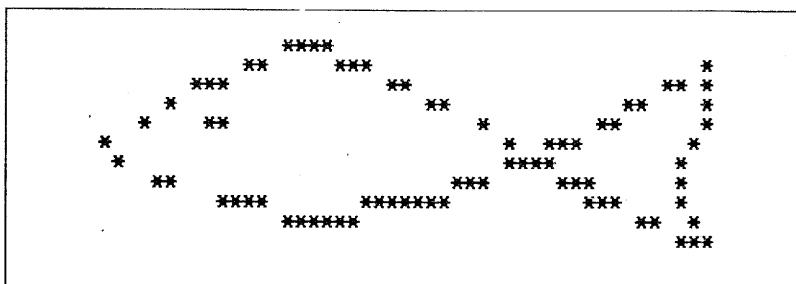
ZÁKLADY GRAFIKY

Začneme s nejzákladnějšími myšlenkami a metodami tvorby obrázků počítačem. Nejjednodušší metodou tvorby obrázků a grafů, tvořených počítačem, je použití standardních výstupních příkazů konkrétního programovacího jazyka. V případě jazyku BASIC je to mnohostraně využitelný příkaz PRINT. Ten můžeme používat např. pro sestavování grafických vzorů z písmen, číslic a ostatních symbolů. Na mnoha mikropočítačových systémech jsou však k dispozici mnohem výkonnější grafické techniky ve formě speciálních grafických příkazů. Pomocí těchto příkazů můžeme "kreslit" obrázky z bodů a přímek. V této části dálkového kursu Počítačová grafika se naučíme základní použití příkazu PRINT a ovládneme i práci se speciálními grafickými příkazy.

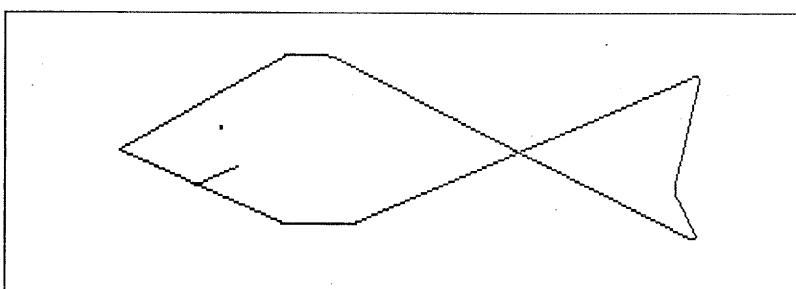
1. Jednoduché obrázky



Obrázky mohou být konstruovány uspořádáním znaků do tvaru požadovaných obrysů na displeji. Obr. 3-1 ukazuje takový jednoduchý obrázek, vytvořený z nesouvislých znaků. Tyto obrázky lze snadno vytvářet v jazyku BASIC právě pomocí příkazu PRINT. I když se jedná o primitivní obrázky s malou rozlišovací schopností pro řadu aplikací plně vyhovují. Obr. 3-2 ukazuje tentýž obrázek nakreslený z přímkových úseků pomocí speciálních grafických příkazů s kterými se setkáme později. V této kapitole prozkoumáme možnosti příkazu PRINT pro vytváření grafického výstupu a uvedeme si i grafické příkazy, které jsou typické pro mnohé mikropočítačové systémy, s přímou aplikací na počítačových systémech IBM PC/XT, AT.



Obr. 3-1 Grafický výstup sestavený ze znaků pomocí příkazu PRINT.



Obr. 3-2 Grafický výstup sestavený z přímek pomocí speciálních grafických příkazů.

1.1. Grafika příkazu PRINT

V běžném alfanumerickém režimu jsou příkazem PRINT zobrazovány znaky na řádcích obrazovky nebo tisknutý tiskárnou. Tak můžeme zobrazovat libovolná písmena, číslice, interpunkční znaménka a další symboly, které jsou obsaženy v množině znaků znakového generátoru zobrazovacího adaptéru mikropočítače. Například: příkaz v programu

```
( ) 10 PRINT "* * * *"
```

zobrazí na začátku řádku čtyři hvězdičky.

Obdélník z hvězdiček můžeme snadno vytvořit příkazy

```
10 PRINT "*****"
20 PRINT "*      *"
30 PRINT "*      *"
40 PRINT "*****"
```

nebo velká písmena pomocí příkazů.

```
10 PRINT "GGGGG RRRR    AAA    PPPP    H    H    I    CCCC SSSSS"
20 PRINT "G    G R    R A    A P    P H    H    I    C    S    "
30 PRINT "G    R    R A    A P    P H    H    I    C    SSS   "
40 PRINT "G    GG RRRR    AAAAAA PPPP    HHHHHH    I    C    SSS   "
50 PRINT "G    G R    R A    A P    H    H    I    C    S    "
60 PRINT "GGGGG R    R A    A P    H    H    I    CCCC SSSSS"
```

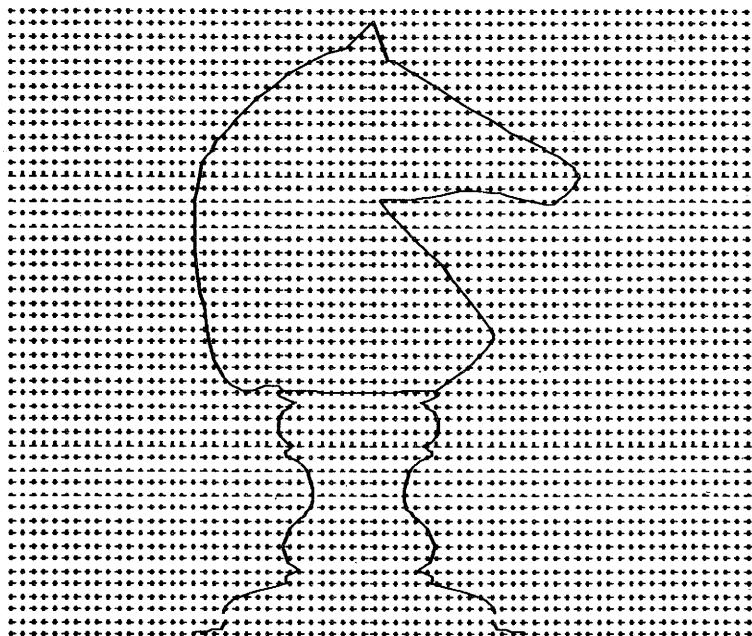
```
( ) 10 'PROGRAM 3-1. RYBA POMOCI PRIKAZU PRINT.
20 PRINT "          ****"
30 PRINT "          **      ***          **"
40 PRINT "          ***      **          **  **"
50 PRINT "          *          **          **  **"
60 PRINT "          *      **          *      **  **"
70 PRINT "          *          *      ***          **"
80 PRINT "          *          ***          **"
```

```
90 PRINT "      **          ***      ***      *"
100 PRINT "      ****      *****      ***      *"
110 PRINT "      *****      ***      **      **"
120 PRINT "          ***      ***"
130 END
```

Prog. 3-1 Vytvoření obrázku ryby.

Počet řádek pro zobrazování obrázků jak jsme si již řekli závisí na použitém mikropočítáčovém systému. Rozměry obrazovky jsou zpravidla od 16 do 48 řádek a od 23 do 80 znaků na řádek. Pokud bychom chtěli slovo GRAPHICS umístit do středu obrazovky při rozměrech stínítka 40 znaků na řádek, 16 řádků, musíme umístit toto slovo do pozice 17 na řádku 8. To by mohlo být provedeno například tak, že řada příkazů PRINT by způsobila přechod na 8. řádek (tj. přeskocení 7 řádků), za nimiž by následoval příkaz PRINT TAB (17); "GRAPHICS".

Složité obrázky lze vytvářet tak, že se nejprve požadovaný tvar nakreslí na grafický čtverečkový papír. Podle tohoto nákresu můžeme určit pozice pro tisk znaků, které vytvoří požadovaný obraz nebo obrys. Při tisku pak každý horizontální řádek znaků na grafickém papíru odpovídá jednomu tiskovému řádku.



Obr. 3-3 Obrázek nakreslený na grafickém papíru (vytvořeném tiskem znaků plus) se používá při určení pozic znaků (Prog. 3-2).

Grafický papír je rozdělen na čtverečky, ale ploška, na které se znak zobrazuje na obrazovce nebo tiskne na tiskárně, je obdélníková - výška je obvykle poněkud větší než šířka. Proto je při použití standardního čtverečkovaného papíru pro určení tiskových pozic bude zobrazovaný obrázek značně zkreslen. Pro libovolný počítačový systém však můžeme vytvořit grafický papír natisknutím celé strany vyplněné znaménky plus "+", jak je uvedeno na Obr. 3-3. Program 3-2 tiskne siluetu obrázku nakresleného na Obr. 3-3.

```
10 'PROGRAM 3-2. SILUETA SACHOVE FIGURKY POMOCI PRINT.  
20 PRINT TAB(10); "+" #"
```

```
30 PRINT TAB(10); "# #####
40 PRINT TAB(10); "# #####"
50 PRINT TAB(10); "# #####"
60 PRINT TAB(10); "# #####"
70 PRINT TAB(10); "# #####"
80 PRINT TAB(10); "# #####"
90 PRINT TAB(10); "# #####"
100 PRINT TAB(10); "# #####"
110 PRINT TAB(10); "# #####"
120 PRINT TAB(10); "# #####"
130 PRINT TAB(10); "# #####"
140 PRINT TAB(10); "# #####"
150 PRINT TAB(10); "# #####"
160 PRINT TAB(10); "# #####"
170 PRINT TAB(10); "# #####"
180 PRINT TAB(10); "# #####"
190 PRINT TAB(10); "# #####"
200 PRINT TAB(10); "# #####"
210 PRINT TAB(10); "# #####"
220 PRINT TAB(10); "# #####"
230 PRINT TAB(10); "# #####"
240 PRINT TAB(10); "# #####"
250 PRINT TAB(10); "# #####"
260 PRINT TAB(10); "# #####"
270 PRINT TAB(10); "# #####"
280 PRINT TAB(10); "# #####"
290 PRINT TAB(10); "# #####"
300 PRINT TAB(10); "# #####"
310 PRINT TAB(10); "# #####"
320 PRINT TAB(10); "# #####"
330 PRINT TAB(10); "# #####"
340 PRINT TAB(10); "# #####"
350 PRINT TAB(10); "# #####"
360 PRINT TAB(10); "# #####"
370 PRINT TAB(10); "# #####"
380 PRINT TAB(10); "# #####"
390 PRINT TAB(10); "# #####"
400 PRINT TAB(10); "# #####"
410 PRINT TAB(10); "# #####"
420 PRINT TAB(10); "# #####"
430 PRINT TAB(10); "# #####"
```

```

440 PRINT TAB(10); " #####"
450 PRINT TAB(10); " #####"
460 PRINT TAB(10); " #####"
470 PRINT TAB(10); " #####"
480 PRINT TAB(10); " #####"
490 PRINT TAB(10); " #####"
500 PRINT TAB(10); " #####"
510 PRINT TAB(10); " #####"
520 PRINT TAB(10); " #####"
530 PRINT TAB(10); " #####"
540 END

```

Prog. 3-2 Silueta figurky (šachová figurka) pomocí příkazu PRINT.

Protože jsou řádky 370 až 420 v Prog. 3-2 identické, mohli bychom použít také cyklus obsahující jeden příkaz PRINT místo těchto šesti řádek. Pro každý obrázek obsahující opakované nebo symetrické vzory můžeme odvodit cykly vytvářející takový vzor. Např. pyramida na Obr. 3-4 může být zobrazena programem Prog. 3-3.



Obr. 3-4 Symetrické vzory, jako je tato pyramida vyvořená Prog. 3-3, mohou být programovány pomocí cyklů, které minimalizují počet příkazů PRINT.

```

10 'PROGRAM 3-3. PROGRAM PRO KRESLENI PYRAMIDY.
20 K=30
30 FOR N=1 TO 29 STEP 2
40     PRINT TAB(K);
50     FOR J=1 TO N

```

```
60      PRINT "A";
70      NEXT J
80      PRINT
90      K=K-1
100 NEXT N
110 END
```

Prog. 3-3 Symetrický vzor (pyramida) příkazem PRINT se smyčkami v programu.

Počet příkazů PRINT, nutných k zobrazení, by také mohl být redukován zakódováním tiskových pozic do příkazů DATA. Tato zakódovaná data mohou být z příkazů DATA čtena, dekódována a tisknuta. Existuje mnoho způsobů kódování tiskových řádek. Například, každý řádek obrázku může být zaznamenán jako dvojice čísel. Prvé číslo udává počáteční tiskovou pozici a druhé číslo udává kolik znaků se má na této řádce vytisknout. Program 3-4 tiskne siluetu uvedenou na Obr. 3-3 pomocí tohoto kódování.

Stínování obrázků se může provádět příkazem PRINT změnou typu znaků užitých při zobrazení. Hvězdičky, znaky "@" a ostatní plné symboly vytváří tmavší plochy než znaky "O" nebo "I". Na některých tiskárnách je možno získat stínování pomocí přetisku (overprint), tj. umístěním dvou nebo více znaků na stejnou tiskovou pozici.

Na některých mikropočítáčích jsou k dispozici dokonce i speciální grafické znaky poskytující další metodu obměny struktury obrázků. Tyto znaky se mohou být zadávány z klávesnice a lze je používat jako každý jiný znak. Na ostatních systémech můžeme zobrazovat speciální znaky příkazem PRINT CHR\$(A), kde A je kód požadovaného znaku. Obr. 3-5 ukazuje obraz vytvořený z grafických znaků.

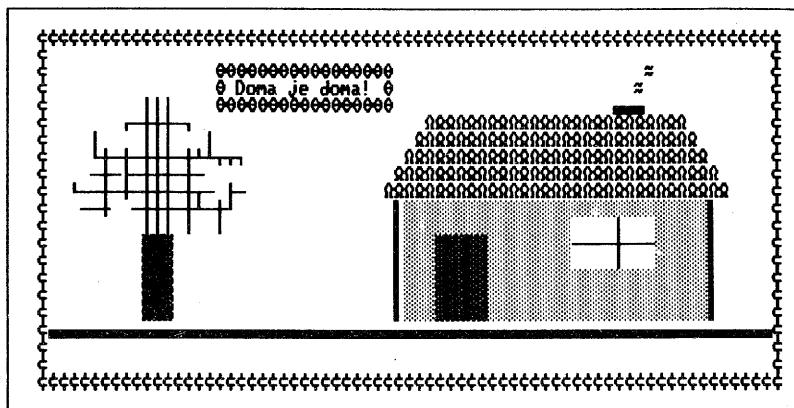
```
10 'PROGRAM 3-4. SILUETA SACHOVE FIGURKY Z KODOVANYCH DAT.
20     'KAZDY TISKOVY RADEK JE ULOZEN JAKO PAR CISEL.
30     'PRVE CISLO JE POZICE, KDE SE MA ZACIT TISKNOUT.
40     'DRUHE CISLO JE KOLIK ZNAKU SE MA TISKNOUT.
```

```

50      'VSTUP JE UKONCEN NACTENIM 0,0.
60      'ZNAK PRO TISK LZE ZMENIT NA RADKU 110.
70  READ P, N
80  IF P=0 AND N=0 THEN 200
90  PRINT TAB(P);
100 FOR K=1 TO N
110     PRINT "#";
120 NEXT K
130 GOTO 70
140 DATA 28,1,25,5,22,9,20,13,19,16,17,20,16,23,15,26,14
150 DATA 29,14,32,13,35,13,36,12,37,12,36,12,19,44,3,12
160 DATA 20,12,21,12,22,12,23,12,24,12,25,13,25,13,26,13
170 DATA 27,14,27,14,26,15,24,15,23,17,19,22,13,20,17,20
180 DATA 17,21,15,22,13,23,11,23,11,23,11,23,11,23,11,23
190 DATA 11,21,15,21,15,21,15,21,15,17,23,13,31,13,31,13
200 DATA 31,10,37,10,37,10,37,0,0
210 END

```

Prog. 3-4 Silueta šachové figurky příkazem PRINT a zakódovaná data.



Obr. 3-5 Obrázek vytvořený ze speciálních grafických znaků.

1.2. Koncepcie bodové grafiky

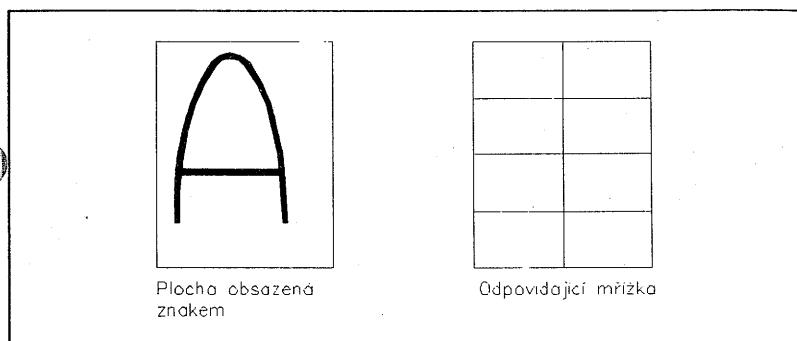
Každý znak, zobrazovaný na obrazovce monitoru lze nakreslit do malého obdélníku. Tuto oblast je na mnoha

mikropočítáčích možno rozdělit do mřížky několika menších obdélníčků. Tyto menší obdélníčky se nazývají obrazové prvky nebo jednoduše body. Počet bodů, na které se rozděluje obdélníček znaků, závisí na konkrétním systému. Zpravidla dva až osm bodů bývá horizontálně a od tří do dvaceti vertikálně. Obr. 3-6 ukazuje mřížku 2×4 odpovídající plošce jednoho znaku. Co výše uvedený systém dělení znaků přináší? Stínítko obrazovky, na němž se používá tento systém, bude obsahovat dvakrát tolik bodů jako je znaků na řádce a vertikálně čtyřikrát více bodů než řádek. Má-li takový systém 48 řádek po 80 znacích, můžeme vynášet horizontálně do jedné ze 160 pozic a vertikálně do jedné ze 192 pozic.

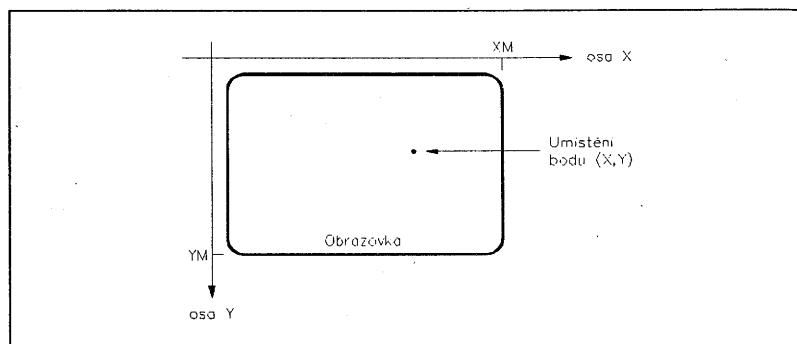
Body jsou přímo přístupné v grafickém režimu. Pro přechod do grafického režimu vyžadují některé systémy, abychom napsali speciální inicializační příkaz, jako je GR nebo SCREEN. Jiné systémy přecházejí automaticky do grafického režimu při uvedení některého speciálního grafického příkazu, jako je vynesení bodu, nebo nakreslení přímky.

Vynesení bodu na stínítku obrazovky znamená, že říkáme počítači, aby rozsvítil malý světelný obdélníček na příslušné pozici bodu. Jednotlivé pozice bodů jsou označovány souřadnicemi. To znamená, že musíme stanovit pozici bodu dvojicí celých čísel (X, Y). Prvé celé číslo X určuje horizontální a druhé číslo Y vertikální vzdálenost na stínítku obrazovky. Zpravidla jsou tyto vzdálenosti měřeny zleva doprava a shora dolů. To znamená, že počátek souřadného systému na stínítku obrazovky je v levém horním rohu, jak je uvedeno na Obr. 3-7. X. Horizontální souřadnice se může měnit zleva doprava po celých číslech 0, 1, 2, ... až do nějakého maxima XM. Y a vertikální souřadnice se může měnit shora dolů po celých číslech 0, 1, 2, ..., YM. Souřadnicemi je pozice bodu v levém horním rohu označena

jako $(0,0)$. Souřadnice bodu v pravém dolním rohu je (XM, YM) . Tento standard používají mikropočítače APPLE, TRS-80, IBM PC, ATARI. Všechny naše další grafické diskuse budou založeny na tomto souřadném systému.



Obr. 3-6 Plocha obsazená jedním znakem je rozdělena mřížkou na menší obdélníkové plošky (body).



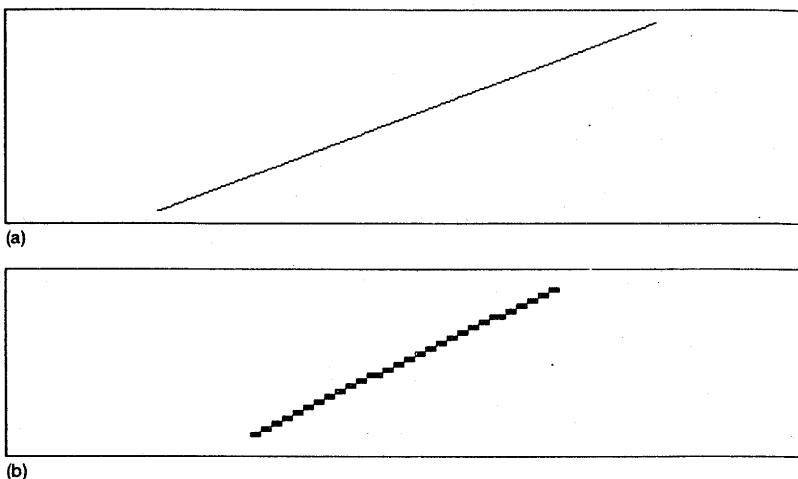
Obr. 3-7 Souřadný systém, používaný mnoha mikropočítači, má počátek v levém horním rohu obrazovky.

Některé mikropočítače užívají nám ze školy obvyklejší souřadný systém s počátkem v levém dolním rohu obrazovky. Pro tyto systémy má bod v levém dolním rohu souřadnice $(0,0)$ a v pravém horním rohu (XM, YM) . Z řady takto

koncipovaných grafických systémů jmenujme namátkou např. mikropočítače firem Hewlett-Packard, Tektronix a Intercolor. Naše příklady programů lze převést do tohoto souřadného systému nahradou každé hodnoty souřadnice Y vynášených bodů hodnotou YM - Y.

V praxi hodnoty souřadnic maximální šířky a výšky stínítka obrazovky, XM a YM, se mění od 40 do několika set. Větší hodnoty XM a YM umožňují kreslit přímky s hladším průběhem, protože každý bod zabírá menší oblast na stínítku. Počet bodů na úsečce se nazývá rozlišovací schopnost systému. Obr. 3-8 ukazuje vlivy vyšší a nižší rozlišovací schopnosti na vzhled úseček. Horizontální a vertikální úsečky budou hladké bez ohledu na rozlišovací schopnost. Diagonální přímky vypadají při nižší rozlišovací schopnosti jako hruběji stupňovité.

Rozlišovací schopnost se určuje mnohem přesněji počtem bodů na centimetr nebo na inch (dpi). Při vynášení bodů na větším stínítku obrazovky při stejném počtu bodů na jeho šířku jako na malé obrazovce bude rozlišovací schopnost menší (menší počet bodů na centimetr). U většiny mikropočítačů lze užít monitory s různě velkými stínítky obrazovky. Změna velikosti stínítka však nezmění počet bodů zobrazovaných počítačem. Tento počet je určen grafickými možnostmi příslušného systému, tj. tím, jak byl navržen a vyroben. Některé systémy nám umožní výběr z několika grafických režimů s různými rozlišovacími schopnostmi. Tyto režimy se obvykle označují jako nízká rozlišovací schopnost (low resolution, poskytující méně bodů), střední rozlišovací schopnost (medium resolution, poskytující více bodů) a vysoká rozlišovací schopnost (high resolution, poskytující nejvíce bodů).



Obr. 3-8 Přímky nakreslené s vyšší rozlišovací schopností (a) a nižší rozlišovací schopností (b).

1.3. Vynášení bodů

Chceme-li vynést bod na stínítku obrazovky, musíme použít příslušné grafické příkazy. Na rozdíl od příkazu PRINT nejsou grafické příkazy standardizovány. Různé mikropočítačové systémy užívají různé příkazy k provedení stejné operace. Protože jsou základní grafické operace stejné (přechod do grafického režimu, vynášení bodů, atp.), budeme pracovat se sadou grafických příkazů, které obsahuje GWBASIC na počítačích řady IBM PC. Vzhledem ke stavebnicové konstrukci hardware tohoto počítače budeme předpokládat konfiguraci s grafickým adaptérem typu CGA. V případě, že Váš počítač je vybaven jiným druhem grafického adapteru je třeba využít příslušných standardních simulačních residentních programů tak, aby byla zajištěna kompatibilita s grafickým adaptérem CGA.

Pro náš systém musíme uvádět grafický režim a požadovaný stav rozlišení. Naše sada příkazů bude předpokládat dva rozlišovací režimy s příkazem pro grafickou inicializaci:

SCREEN n – přechod do grafického režimu, kde n udává rozlišovací schopnost: n = 1 je rozlišení 320x200 (čtyři barvy), n = 2 je rozlišení 640x200 (dvě barvy)

Jakmile jsme v grafickém režimu, můžeme vynášet body příkazem:

PSET(X,Y) – zobrazení bodu na obrazovce o souřadnicích X,Y. X a Y jsou číselné konstanty nebo výrazy.

Příklad užití příkazů SCREEN a PSET je uveden v Prog. 3-5. Tento program vynáší bod na pozici určené příkazem vstupu INPUT.

Vynášení bodů za hranice stínítka obrazovky vyvolává chybu bud' běhu programu, nebo zkreslení pozice bodu díky efektu překlopení. K překlopení dochází je-li nějaký bod vynášen za hranice na jedné straně stínítka. "Překlopí se" a objeví se na druhé straně.

```
10 'PROGRAM 3-5. VYNASI JEDEN BOD.  
20 PRINT "UVED SOURADNICE BODU X,Y:"  
30 INPUT X, Y  
40 SCREEN 1  
50 PSET(X,Y)  
60 END
```

Prog. 3-5 Vynášení bodu.

Abychom vyloučili tyto problémy, musíme zajistit, že ani vertikální, ani horizontální souřadnice nenabudou hodnot záporných a současně nebudou větší než jsou příslušné

maximální hodnoty dané systémem. To lze zajistit zařazením testů do programu:

$$\begin{aligned} 0 <= X <= XM \\ 0 <= Y <= YM \end{aligned} \quad (3-1)$$

V těchto testech je XM povolená maximální hodnota horizontální a YM je povolená maximální vertikální souřadnice bodu. Prog. 3-6 ukazuje užití těchto testů pro obrazovku o velikosti 280 x 160 bodů:

```
10 'PROGRAM 3-6. TEST HRANIC OBRAZOVKY A VYNASENI
20 PRINT "UVED SOURADNICE BODU X,Y:"
30 INPUT X,Y
40 IF X < 0 OR X > 279 OR Y < 0 OR Y > 159 THEN 80
50 SCREEN 1
60 PSET(X,Y)
70 GOTO 90
80 PRINT "POKUS VYNESENI MIMO STINITKO !"
90 END
```

Prog. 3-6 Vynášení bodů a testy, zda nepadnou mimo stínítko obrazovky.

Pro mnohé aplikace je výhodná možnost "vymazání" bodu, který byl předtím vynesen. To nám umožní další příkaz:

PRESET(X,Y) – vymazání bodu (bodu) vyneseného dříve na pozici X,Y. X a Y jsou numerické výrazy nebo konstanty. Nejsou-li celočíselné budou, obdobně jako tomu bylo u předchozí instrukce, zaokrouhleny.

Pokud není bod na pozici X,Y rozsvícen, nebude mít PRESET žádný viditelný efekt. Program 3-7 ukazuje, jak může být tento příkaz použit pro zhasnutí bodu po časovém zpoždění přibližně 2 sekund.

```
10 'PROGRAM 3-7. ROZSVICENI A ZHASNUTI BODU.
```

```
20 PRINT "UVED MAXIMALNI HORIZONTALNI A VERTIKALNI"
30 PRINT "HODNOTY PRO TENTO ROZLISOVACI REZIM XM, YM"
40 INPUT XM, YM
50 PRINT "UVED SOURADNICE BODU X, Y:"
60 INPUT X, Y
70 IF X < 0 OR X > XM OR Y < 0 OR Y > YM THEN 130
80 SCREEN 1
90 PSET(X,Y)
100 FOR J = 1 TO 1000: NEXT J
110 PRESET(X,Y)
120 GOTO 140
130 PRINT "POKUS VYNESENI BODU MIMO STINITKO !"
140 END
```

Prog. 3-7 Vynášení bodu a mazání.

Zpožďovací smyčky, jako jsou příkazy na řádce 100 v Prog. 3-7, mohou být vkládány do grafických programů v případě zobrazování blikajících bodů. Lze je také použít pro zpoždění smazání obrázku, tj. aby byl na stínítku obrazovky tak dlouho, dokud se nezobrazí následující obrázek. Tyto smyčky umožňují realizovat zpoždění přibližně 1 sekundu na každých 500 cyklů smyčky, v závislosti na rychlosti zpracování příslušného systému (IBM PC, PC-turbo, AT, AT-turbo, 386, a pod.).

Před vytvořením zvoleného obrázku nebo před výpisem programu potřebujeme často vymazat celou plochu stínítka obrazovky. K tomuto účelu použijeme následující příkaz:

CLS – vymazání stínítka a návrat kurzoru do levého horního rohu stínítka.

Tento příkaz může být použit buď "v" anebo i mimo režim GRAPHICS. Užití příkazu CLEARSCEEN je ilustrováno v Prog. 3-8, který vynáší náhodný vzor bodů. Pozor ! Funkce RND, užitá v tomto programu, má v různých verzích jazyka BASIC různé formy.

```

10  ' PROGRAM 3-8. VYNASENI NAHODNYCH BODU.
20  PRINT "UVED MAXIMALNI HORIZONTALNI A VERTIKALNI"
30  PRINT "HODNOTY PRO TENTO ROZLISOVACI REZIM XM, YM"
40  INPUT XM, YM
50  PRINT "UVED POSET BODU KTERE MAJI BYT VYNESENY:"
60  INPUT P
70  CLS
80  SCREEN 1
90  FOR K = 1 TO P
100    X = XM * RND(1)
110    Y = YM * RND(1)
120  PSET(X,Y)
130  NEXT K
140 END

```

Prog. 3-8 Vytvoření vzoru náhodným vynášením bodů.

1.4. Kreslení přímek

Příkaz PSET může být použit pro kreslení přímek. Má-li být nakreslena přímá čara, vynášíme příslušné body ležící na přímé spojnici dvou krajních bodů. Tyto body jsou vždy ob jednu vedle sebe, takže chceme-li nakreslit např. vertikální přímku, bude se souřadnice Y opakovaně zvětšovat o jedničku a souřadnice X přitom bude konstantní. Programy 3-9 a 3-10 uvádějí příklady kreslení vertikální a horizontální přímkы. Tyto programy vytvoří nepřerušované čary, protože každý vynášený bod je malý obdélníček, navazující na bod vynesený v předcházejícím kroku.

Chceme-li získat pomocí příkazu PSET diagonální přímku, musíme počítat hodnoty obou souřadnic bodů na přímce. Přitom použijeme rovnici /3-2/, která určuje vztah mezi souřadnicemi X a Y přímkы:

$$Y=M \cdot X + B$$

(3-2)

V této rovnici je M sklon /směrnice/ přímky, který může být kladný i záporný nebo nulový. Je-li $M = 0$, dostaneme horizontální přímku. Pro velmi velkou hodnotu M dostaneme téměř vertikálu. B je hodnota úseku na ose Y, kterou by mělo Y, kdyby bylo $X = 0$. Je-li $B = 0$, jedná se tedy o přímku, která prochází přímo počátkem souřadné soustavy.

```

10  'PROGRAM 3-9. PROGRAM PRO KRESLENI VERTIKALNI PRIMKY
20  CLS
30  PRINT "UVED MAXIMALNI HRIZONTALNI A VERTIKALNI"
40  PRINT "HODNOTY PRO TENTO ROZLISOVACI REZIM XM, YM:"
50  INPUT XM, YM
60  PRINT "UVED SOURADNICI X PRO VERTIKALNI PRIMKU:"
70  INPUT X
80  IF X >= 0 AND X <= XM THEN 110
90  PRINT "SOURADNICE X MIMO HRANICE STINITKA. ZNOVU"
100 GOTO 60
110 PRINT "UVED KONCOVE BODY Y1, Y2:"
120 INPUT Y1, Y2
130 IF Y1>=0 AND Y1<=YM AND Y2>=0 AND Y2<=YM THEN 160
140 PRINT "SOURADNICE Y MIMO HRANICE STINITKA. ZNOVU"
150 GOTO 110
160 CLS
170 SCREEN 1
180 FOR Y = Y1 TO Y2
190     PSET(X, Y)
200 NEXT Y
210 END

```

Prog. 3-9 Kreslení vertikální přímky vynášením bodů.

```

10  'PROGRAM 3-10. PROGRAM PRO KRESLENI HORIZONTALNI PRIMKY
20  CLS
30  PRINT "UVED MAXIMALNI HRIZONTALNI A VERTIKALNI"
40  PRINT "HODNOTY PRO TENTO ROZLISOVACI REZIM XM, YM:"
50  INPUT XM, YM
60  PRINT "UVED SOURADNICI Y PRO HORIZONTALNI PRIMKU:"
70  INPUT Y
80  IF Y >= 0 AND Y <= YM THEN 110
90  PRINT "SOURADNICE Y MIMO HRANICE STINITKA. ZNOVU"
100 GOTO 60

```

```

110 PRINT "UVED KONCOVE BODY X1, X2:"
120 INPUT X1, X2
130 IF X1>=0 AND X1<=XM AND X2>=0 AND X2<=XM THEN 160
140 PRINT "SOURADNICE X MIMO HRANICE STINITKA. ZNOVU"
150 GOTO 110
160 CLS
170 SCREEN 1
180 FOR X = X1 TO X2
190     PSET(X,Y)
200 NEXT X
210 END

```

Prog. 3-10 Kreslení horizontální přímky vynášením bodů.

Příkazem PSET můžeme také naprogramovat kreslení obecné přímky podle vztahu /3-2/. Program 3-11 ilustruje kreslení takové přímky na základě uvádění hodnot M a B. Tento program nejprve stanoví, zda může být požadovaná přímka na stínítku obrazovky nakreslena. Pokud nelze přímku nakreslit mezi hranicemi stínítka program zobrazí o této situaci zprávu. Pokud však může být nějaká část přímky nakreslena, pak Prog. 3-11 nakreslí viditelnou část přímky od jedné hrance stínítka ke druhé. Obr. 3-9 ukazuje výstup Prog. 3-11 pro případ M = 1. Uvedená přímka se objeví na druhé straně než bychom očekávali při zobrazení v konvenční souřadné soustavě, kde přímka s pozitivní směrnicí vede nahoru zleva doprava. K tomuto jevu dochází z tohoto důvodu, že náš počátek (0,0) je umístěn v levém horním rohu stínítka. Uvedený program můžeme modifikovat tak, abychom získali přímky s orientací shodnou s orientací v konvenčním souřadném systému násobením každé vstupní hodnoty směrnice -1 a záměnou každé vstupní hodnoty b za YM - B.

Program 3-11 nevytvoří nepřerušované čáry, pokud bude velikost směrnice větší než 1. V takovém případě budou mezi zobrazenými ploškami mezery. To lze opravit zvětšením souřadnice Y o 1 místo X, pokud ABS(M) > 1. Také by bylo

možno program modifikovat tak, aby přímku zobrazil uprostřed stínítka obrazovky. Příslušné metody budou diskutovány v Kapitole 4 při studiu grafů.

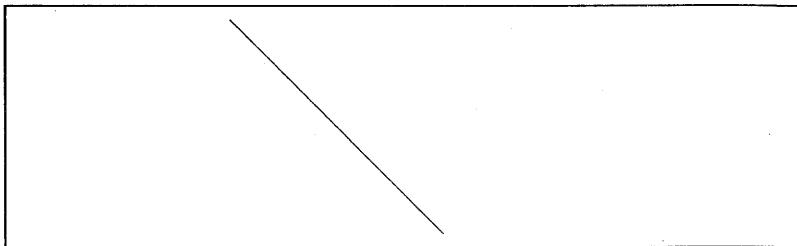
```
10  'PROGRAM 3-11. KRESLENI OBECNE PRIMKY POMOCI PSET
20  CLS
30  PRINT "UVED MAXIMALNI HRIZONTALNI A VERTIKALNI"
40  PRINT "HODNOTY PRO TENTO ROZLISOVACI REZIM XM,YM:"
50  INPUT XM,YM
60  PRINT "UVED HODNOTU SMERNICE A Y-USEKU M,B:"
70  INPUT M, B
80          'JE-LI SMERNICE NEGATIVNI A PRIMKA PROTINA
90          'OSU Y S HODNOTOU MENSI NEZ NULA, JE PRIMKA
95          'MIMO STINITKO
100 IF M < 0 AND B < 0 THEN PRINT "PRIMKA MIMO": GOTO 430
110          'JE-LI SMERNICE POSITIVNI ALE PRIMKA PROTINA
120          'OSU Y S HODNOTOU VETSI NEZ YM, JE PRIMKA
125          'MIMO STINITKO
130 IF M > 0 AND B < YM THEN PRINT "PRIMKA MIMO": GOTO 430
140          'JINAK ZJISTI NEJLEVEJSI BOD PRIMKY.
150          'JE-LI PRUSECIK Y MEZI 0 A YM PAK JE NEJLE-
160          'VEJSI BOD NA LEVEM OKRAJI STINITKA.
170 X1 = 0
180          'JE-LI SMERNICE NEGATIVNI A PRUSECIK Y MA
190          'HODNOTU VETSI NEZ YM, JE NEJLEVEJSI BOD NA
200          'DOLNIM OKRAJI KDE Y=YM, TAKZE X1=(YM-B)/M.
210 IF M < 0 AND B > YM THEN X1 = (YM - B)/M.
220          'MA-LI TENTO BOD X VETSI XM, JE PRIMKA MIMO.
230 IF X1 > XM THEN PRINT "PRIMKA MIMO": GOTO 430
240          'JE-LI SMERNICE POZITIVNI A PRIMKA PROTINA Y
250          'S HODN. MENSI NEZ 0, JE NEJLEVEJSI BOD NA
260          'HORNIM OKRAJI STINITKA, KDE Y = 0 TAKZE
265          'X1 = (O-B)/M NEBO X1 = O-B/M.
270 IF M > 0 AND B < 0 THEN X1 = - B/M
280          'JE-LI TENTO BOD ZA XM, JE PRIMKA MIMO
290 IF X1 > XM THEN PRINT "PRIMKA JE MIMO": GOTO 430
300          'JINAK JE PRIMKA ALESPOZ CASTECNE NA STINIT-
310          'KU ZACATEK NA NEJLEVEJSIM BODE X1. PRO ROS-
320          'TOUCI HODNOTY X POCITAT HODNOTY Y A VYNASET
330          'X,Y. POKRACOVAT, DOKUD NENI
```

```

335      '1. X > XM PRIMKA VYCHAZI ZE STINITKA VPRAVO
340      '2. Y < 0 PRIMKA VYCHAZI ZE STINITKA NAHORE
350      '3. Y > YM PRIMKA VYCHAZI ZE STINITKA DOLE.
360  CLS
370  SCREEN 1
380  FOR X = X1 TO XM
390      Y = M + X + B
400      IF Y < 0 OR Y > YM THEN 430
410      PSET(X,Y)
420  NEXT X
430  END

```

Prog. 3-11 Kreslení obecné přímky pomocí rovnice přímky a vynášení bodů.



Obr. 3-9 Přímka, kreslená programem 3-11, se směrnicí 1 a průsečíkem z y = 0.

Chceme-li vynášet přímku danou souřadnicemi dvou koncových bodů, musíme určit průsečík s osou Y a směrnicí, abychom mohli vypočítat všechny potřebné body přímky. Jsou-li dané koncové body (X_1, Y_1) a (X_2, Y_2) , pak pro M a B platí

$$M = \frac{Y_2 - Y_1}{X_2 - X_1} \quad (3-3)$$

$$B = Y_1 - M \cdot X_1$$

Program pro kreslení přímky na základě rovnic /3-3/ z daných souřadnic dvou koncových bodů vypočítá hodnoty M a

B a vynese jednotlivé body přímky mezi oběma koncovými body.

Mnoho programovacích jazyků standardně disponuje grafickými příkazy, které automaticky nakreslí přímku mezi dvěma danými koncovými body. Pro tuto funkci umožňuje následující příkaz.

LINE(X1,Y1)-(X2,Y2) – kreslí přímou čáru z pozice X1,Y1 do pozice X2,Y2. X1, Y1, X2, Y2 jsou numerické konstanty nebo výrazy.

Program 3-12 užívá příkaz LINE pro nakreslení přímky mezi libovolnými dvěma body.

```
10  'PROGRAM 3-12. KRESLENI PRIMKY PRIKAZEM LINE
20  CLS
30  PRINT "UVED MAXIMALNI HRIZONTALNI A VERTIKALNI"
40  PRINT "HODNOTY PRO TENTO ROZLISOVACI REZIM XM, YM:"
50  INPUT XM, YM
60  PRINT "UVED HODNOTU SOURADNICE PRVEHO BODU X1, Y1:"
70  INPUT X1, Y1
80  IF X1, >= 0 AND X1 <= XM THEN 110
90  PRINT "SOURADNICE X MIMO."
100 GOTO 60
110 IF Y1 >= 0 AND Y1 <= YM THEN 140
120 PRINT "SOURADNICE Y MIMO."
130 GOTO 60
140 PRINT "UVED HODNOTU SOURADNICE DRUHEHO BODU X2, Y2:"
150 INPUT X2, Y2
160 IF X2 >= 0 AND X2 <= XM THEN 190
170 PRINT "SOURADNICE X MIMO."
180 GOTO 140
190 IF Y2 >= 0 AND Y2 <= YM THEN 220
200 PRINT "SOURADNICE Y MIMO."
210 GOTO 140
220 CLS
230 SCREEN 1
240 LINE(X1, Y1)-(X2, Y2)
```

250 END

Prog. 3-12 Kreslení přímky příkazem pro kreslení přímek.

1.5. Kreslení obrázku a barvy

Příkazy pro vynášení bodů a kreslení přímek jsou základními nástroji pro zobrazování grafické informace. Těmito příkazy můžeme vytvářet obrazy, siluety, třírozměrná tělesa a složité scény. U mnoha grafických systémů můžeme naše obrázky doplnit o barvy.

1.5.1 Barvy

Možnost barevného zobrazení grafické informace má značný význam. Barvy můžeme použít pro vytváření srozumitelnějších obrazů. O tom, že složité grafy budou zpřehledněny užitím barev pro odlišení různých objektů nebo oblastí se není třeba přesvědčovat. Barvy jsou často používány pro zdůraznění nebo prosvětlení důležitých částí grafu, pro přiblížení skutečnosti. Můžeme je používat také pro vytváření atraktivnějšího obrázku či grafu

V naší sadě speciálních grafických příkazů přidáme následující příkaz:

COLOR F,B Tento příkaz nastavuje barvu bodů na barvu F (foreground-popředí) na stínítku obrazovky barvy B (background-pozadí). F a B jsou numerické konstanty nebo výrazy.

Např. pro grafický adaptér CGA jsou kódy barev pro F a B:

0 - černá, 4 - červená

1 - modrá, 5 - fialová

2 - zelená, 6 - červenohnědá

3 - světle modrá, 7 - bílá

Zvýšením výše uvedených konstant o 16 na grafickém adaptéru CGA způsobí blikání jednotlivých barevných bodů.

V jednom programu je možno použít několik příkazů COLOR. Například, následující posloupnost příkazů nastaví barvu pozadí obrazovky na světle modrou, pak zobrazí vzory v barvě červené, bílé a zelené.

COLOR 4,3

COLOR 1,3

COLOR 2,3

Nastavení pozadí a popředí na stejnou barvu ($F = B$) způsobí, že následující zobrazení je neviditelné. Nastavení F na stejnou barvu jako B a následující opakování kresby čáry, která je vidět na obrazovce, "vymaže" tuto čáru. Pokud není v programu použito žádného příkazu COLOR, předpokládá se,

že má systém standardní implicitní barvy nastavené pozadí na černou, popředí na bílou.

Některé systémy mají více možností volby barvy a jiné se omezují na stanovené kombinace barev. Někdy lze použít volbu barev vně grafického režimu pro nastavení barvy znaků.

Výběr barevných kombinací pro určité zobrazení by měl být pečlivě uvážen. Náhodný výběr může vyvolat kříklavý, nepříjemný dojem. Užití menšího počtu barev je obvykle lepší. Atraktivní může být volba barvy pozadí komplementární k jedné z barev popředí. Komplementární (doplňkové) barevné kombinace jsou červená s modrozelenou, modrá s oranžovou, žlutá s modrofialovou a zelená s červenofialovou (magenta).

Světlé pozadí (řekněme modré) je vhodné použít v kombinaci s tmavšími barvami popředí. Má-li se v nějakém zobrazení použít mnoho barev, je nejlepší šedé nebo neutrální pozadí. Černé obrysové čáry kolem oblastí jiných barev pomáhají snížit barevné kolize při použití mnoha barev.

1.5.2 Kreslení obrázku

Příklady užití příkazů PSET, LINE a COLOR při vytváření obrázků jsou uvedeny v následujících programech. V Prog. 3-13 kreslíme trojúhelník, pravoúhelník nebo obecný mnohoúhelník o libovolném počtu stran v různých barevných kombinacích.

```
10  'PROGRAM 3-13. KRESLENI OBECNEHO MNOHOUHELNika.  
20  'PROGRAM KRESLI MNOHOUHELNIK O MAX. 10 STRANACH.  
30  'VRCHOLOU SE UVADEJI V PORADI PROPOJENI.  
40  'MNOHOUHELNIK JE ZOBRAZEN V BAREVNÝCH KOMBINACÍCH  
50  'POPŘEDÍ A POZADI, S CASOVÝM ZPOZDENÍM MEZI  
60  'NASLEDUJICIMI displeji.
```

```

70   -----
80   CLS
90   DIM X(19), Y(10)
100  PRINT "UVED MAXIMALNI HODNOTY SOURADNIC XM, YM:"
110  INPUT XM,YM
120  PRINT "KOLIK VRCHOLU MA TVUJ MNOHOUEHNIK ?"
130  INPUT N
140  PRINT "UVED VRCHOLY V PORADI PROPOJENI X,Y:"
150  FOR K = 1 TO N
160    INPUT X(K), Y(K)
170    IF X(K) >= 0 AND X(K) <= XM THEN 200
180    PRINT "SOURADNICE X JE MIMO. ZKUS TO ZNOVU."
190    GOTO 160
200    IF Y(K) >= 0 AND Y(K) <= YM THEN 230
210    PRINT "SOURADNICE Y JE MIMO. ZKUS TO ZNOVU."
220    GOTO 160
230  NEXT K
240  SCREEN 1
250  FOR F = 0 TO 7
260    FOR B = 0 TO 7
270      COLOR B
280      CLS
290      FOR K = 1 TO N-1
300        LINE(X(K),Y(K))-(X(K+1),Y(K+1)),F
310      NEXT K
320      'SPOJENI POSLEDNIHO VRCHOLU S PRVYM
330      LINE(X(N),Y(N))-(X(1),Y(1)),F
340      PRINT "STISKNI NEJAKOU KLAVESU PRO DALSI
BAREVNOU KOMBINACI."
350      IF INKEY$ = "" THEN 350
360    NEXT B
370  NEXT F
380 END

```

Prog. 3-13 Kreslení obecného mnohoúhelníku a barva.

Souřadnice vrcholů se uvádějí v pořadí propojení a ukládají se do polí X a Y. Tento polygon je pak zobrazen kreslením úsečky z bodu X(1),Y(1) do bodu X(2),Y(2), pak z bodu X(2),Y(2) do bodu X(3),Y(3) atd.

Program 13-3 také ilustruje další formu časového zpoždění. V tomto případě se zpracování zastavuje na řádku 350. Program bude čekat na interakci s uživatelem, tj. na stisknutí nějaké klávesy na klávesnici. Funkce INKEY\$, dostupná v některých verzích jazyka BASIC, je inicializována systémem na prázdný řetězec. Při zpracování této funkce dostaneme hodnotu klávesy, která byla právě na klávesnici stisknuta. Tento druh časového zpoždění je užitečný zvláště když nějaký program vytváří řadu zobrazení, která má být užita ve spojení se zprávou nebo prezentací. V Prog. 3-13 je zařazena na řádku 340 zpráva STISKNI NEJAKOU KLAVESU PRO DALSI BAREVNUOU KOMBINACI, která upozorní na přerušení práce programu. Aby se tato zpráva nezobrazila přes náš obrázek, můžeme jednoduše tyto instrukce zobrazit jednou na začátku programu. Pak může program vymazat obrazovku, zobrazit obrázek a čekat na požadovanou akci.

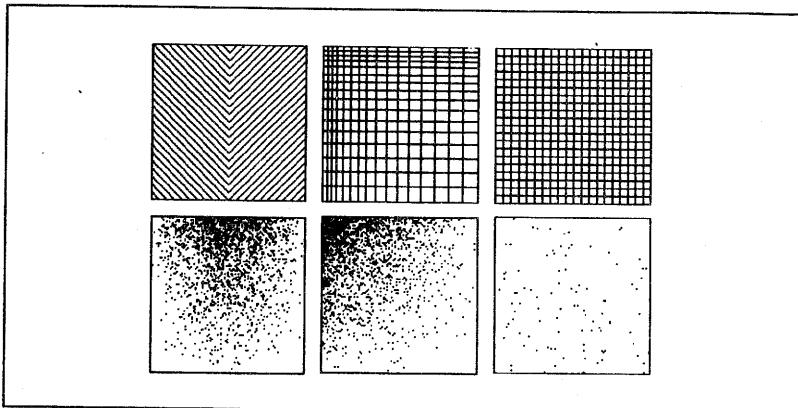
Technika zobrazování plných barevných ploch je ilustrována v Prog. 3-14. Některé systémy mají další grafické příkazy umožňující přímé vybarvování ploch. Avšak i jednoduché podprogramy pro kreslení přímek nám umožňují speciální efekty "vybarvení". V takových podprogramech se vždy postupně vybírá určitá pravoúhelníková oblast specifikací dvou protilehlých vrcholů.

Plné barevné plochy lze vytvářet kreslením horizontálních, vertikálních nebo diagonálních čar. Přitom můžeme měnit pořadí kreslených čar a výběr barvy, což vytváří různé zvláštní efekty. Pravoúhelníkovou oblast můžeme například vyplnit náhodně zbarvenými horizontálními čarami, kreslenými střídavě shora a zdola, které se setkají uprostřed.

Stínování na našich obrázcích lze získat částečným vyplněním oblastí každou druhou nebo třetí čarou. Také bychom mohli kreslit čáry oběma směry, přerušováním

vytvořené příčně šrafováné stínování. Další možností je měnit mezery mezi stínujícími čarami pro vytváření dojmu postupného zesvětlování (nebo tmavnutí) stínu. Tyto mezery mezi čarami by mohly být zvětšovány pomalu po 1 nebo 2 jednotkách, nebo můžeme zvětšovat mezery mnohem rychleji zdvojnásobením předcházející mezery. Obr. 3-10 ukazuje některé vzory stínování, vytvořené pomocí bodů a čar.

Program 3-14 vytváří na stínítku obrazovky čtverec vyplněním plochy stejným počtem bodů v horizontálním a vertikálním směru.



Obr. 3-10 Stínové vzory, vytvořené bodovou grafikou.

```
10  'PROGRAM 3-14. VYTVAŘI PLNY BAREVNY TROJUHESLNIK.  
20  CLS  
30  PRINT "UVED MAXIMALNI HORIZONTALNI A VERTIKALNI"  
40  PRINT "HODNOTY PRO TENTO ROZLISOVACI REZIM XM,YM:"  
50  INPUT XM,YM  
60  PRINT "UVED SOURADNICE LEVEHO HORNIHO ROHU X1,Y1:"  
70  INPUT X1, Y1  
80  IF X1>=0 AND X1<=XM AND Y1>=0 AND Y1<=YM THEN 110  
90  PRINT "SOURADNICE MIMO. OPAKUJ."  
100 GOTO 60
```

```

110 PRINT "UVED SOURADNICE PRAVEHO DOLNIHO ROHU X2,Y2:"
120 INPUT X2, Y2
130 IF X2>=0 AND X2<=XM AND Y2>=0 AND Y2<=YM THEN 160
140 PRINT "SOURADNICE MIMO. OPAKUJ."
150 GOTO 110
160 PRINT "UVED KOD BARVY POPREDI A POZADI F,B:"
170 INPUT F, B
180 IF F >= 0 AND F <= 7 AND B >=0 AND B <= 7 THEN 210
190 PRINT "CHYBNY KOD BARVY."
200 GOTO 160
210 CLS
220 SCREEN 1
230 COLOR B
240 FOR Y = Y1 TO Y2
250     LINE(X1,Y)-(X2,Y),F
260 NEXT Y
270 END

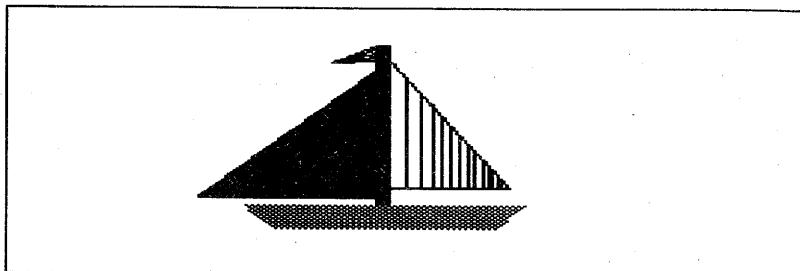
```

Prog. 3-14 Vybarvování plných trojúhelníků.

Jiná situace nastane u systémů, které mají výrazně odlišnou rozlišovací schopnost ve směru osy X od rozlišovací schopnosti ve směru osy Y. Například v případě systému, v němž 100 bodů v horizontálním směru měří celkem 6 cm, ale 100 bodů ve vertikálním směru měří 8 cm. Rozlišení X je tedy 100/6, zatímco rozlišení Y je 100/8. Plocha 100 x 100 bodů by vypadala na obrazovce jako obdélník s větší délkou než je jeho šířka. Aby vznikl čtverec, museli bychom bud' vynést větší počet bodů ve směru X, nebo menší počet bodů ve směru Y. Předpokládejme, že chceme změnit počet bodů ve směru Y. Počet bodů, potřebných ve směru Y pro vytvoření čtverce, můžeme určit násobením původní vertikální délky (100) poměrem rozlišení Y ku rozlišení X (100/8 děleno 100/6, tj. 3/4). Výsledkem je 75, takže musíme vynést ve směru osy Y 75 bodů, aby vznikl čtverec.

Pro určení rozlišovacího poměru (resolution ratio, někdy zvaného poměr stran - aspect ratio) určitého systému nakreslíme úsečku o délce 50 nebo 100 jednotek v obou

souřadných směrech a změříme jejich délky. Poměr délky X ku délce Y pak lze použít pro korekci počtu bodů ve směru osy Y. Poměr délky Y ku délce X lze použít pro korekci počtu plošek ve směru X. Korekce jednoho z obou směrů pro všechny body zobrazení nám umožní kreslit obrázky ve správných proporcích. Program 3-15 demonstruje korekci rozlišení, spolu s užitím stínování a barev při vytváření obrázku (Obr. 3-11).



Obr. 3-11 Obrazový výstup programu Prog. 3-15, se stínováním a korekcí rozlišení.

```

10  'PROGRAM 3-15. PROGRAM KRESLI PLACHETNICI PODLE NAVRHU
20  'NA GRAFICKEM PAPIRU (KOREKCE ROZLISENI X A Y).
30  'TAKE VYPLNUJE A STINUJE NEPRAVIDELNE TVAROVANE
40  'OBLASTI. BODY Z GRAF. PAPIRU JSOU CTENY PRIKAZEM
50  'DATA. HODNOTY Y JSOU UPRAVENY PODLE ROZLIS.
55  'POMERU NASOBENIM CISLEM R.
60  CLS
70  R = 5 / 6
80  PRINT "UVED MAXIMALNI HORIZONTALNI A VERTIKALNI"
90  PRINT "HODNOTY PRO TENTO ROZLISOVACI REZIM XM,YM:"
100 INPUT XM, YM
110 CLS
120 SCREEN 1
130  'KRESLENI STOZARU LODI
140 COLOR 6          'MODRY STOZAR NA SVETLEMODREM POZADI
150 READ XT, YT, XB, YB
160 YT = YT * R      'KOREKCE ROZLISOVACI SCHOPNOSTI
170 YB = YB * R
180 IF YT < 0 OR YT > YM OR YB < 0 OR YB > YM THEN 960

```

```

190      'UDELEJ STOZAR O TLOUSTCE 5 CAR
200  FOR K = 0 TO 4
210      LINE(XT+K,YT)-(XB+K,YB),2
220  NEXT K
230      'CTI SOURADNICE LEVE STRANY
240      'UDELEJ PLNY SPODEK LODI
250  COLOR 6
260  READ XL1, YL1, XL2, YL2
270      'KOREKCE SOURADNIC Y
280  YL1 = YL1 * R
290  YL2 = YL2 * R
300  IF YL1 < 0 OR YL1 > YM OR YL2 < 0 OR YL2 > YM THEN 960
310      'URCI SMERNICI A PRUSECIK S OSOU Y PRO LEVOU STRANU
320  ML = ( YL2 - YL1 ) / ( XL2 - XL1 )
330  BL = YL1 - ML * XL1
340      'CTI SOURADNICE PRAVE STRANY
350  READ XR1, YR1, XR2, YR2
360      'KOREKCE SOURADNIC Y
370  YR1 = YR1 * R
380  YR2 = YR2 * R
390  IF YR1 < 0 OR YR1 > YM OR YR2 < 0 OR YR2 > YM THEN 960
400      'URCI SMERNICI A PRUSECIK S OSOU Y PRO PRAVOU STRANU
410  MR = ( YR2 - YR1 ) / ( XR2 - XR1 )
420  BR = YR1 - MR * XR1
430      'POMOCI SMERNIC A PRUSECIKU S OSOU Y VYPOCITEJ
440      'SOURADNICI X LEVE A PRAVE STRANY A NAKRESLI CARU
450  FOR Y = YL1 TO YL2
460      XL = ( Y - BL ) / ML
470      XR = ( Y - BR ) / MR
480      LINE(XL,Y)-(XR,Y),3
490  NEXT Y
500      'UDELEJ PLNOU VELKOU PLACHTU
510  COLOR 6
520  READ X1, Y1, X2, Y2
530      'KOREKCE SOURADNIC Y
540  Y1 = Y1 * R
550  Y2 = Y2 * R
560  IF Y1 < 0 OR Y1 > YM OR Y2 < 0 OR Y2 > YM THEN 960
570  LINE(X1,Y1)-(X2,Y2),0
580  LINE(X2,Y2)-(X1,Y1),0
590  M = ( Y2 - Y1 ) / ( X2 - X1 )      'URCI M A B DIAGONALY

```

```

600 B = Y1 - M * X1
610      'VYPOCITEJ BODY HORNÍ HRANY PLACHTY
620 FOR X = X1 TO X2 STEP -1
630      Y = M * X + B
640      LINE(X,Y)-(X,Y2)
650 NEXT X
660      'NAKRESLI MALÝ PRAPOREK
670 COLOR 6
680 READ X1,Y1,X2,Y2
690      'KOREKCE SOURADNÍC Y
700 Y1 = Y1 * R
710 Y2 = Y2 * R
720 IF Y1 < 0 OR Y1 > YM OR Y2 < 0 OR Y2 > YM THEN 960
730 FOR Y = Y1 TO Y2 STEP 2
740      LINE(X2,Y2)-(X1,Y),3
750 NEXT Y
760      'UDELEJ MALOU STINOVANOU PLACHTU
770 C = 5
780 READ X1, Y1, X2, Y2
790      'KOREKCE SOURADNÍC Y
800 Y1 = Y1 * R
810 Y2 = Y2 * R
820 IF Y1 < 0 OR Y1 > YM OR Y2 < 0 OR Y2 > YM THEN 960
830 LINE(X1,Y1)-(X2,Y2),3
840 LINE(X1,Y2)-(X2,Y2),3
850 M = ( Y2 - Y1 ) / ( X2 - X1 )      'URCI M A B DIAGONALY
860 B = Y1 - M * X1
870 X = X1 + C
880      'VYPOCITEJ BODY HORNÍ HRANY PLACHTY
890 Y = M * X + B
900 LINE(X,Y)-(X,Y2),3
910 C = C - C * .1
920 X = X + C
930      'DOSAZENO PRAVEHO OKRAJE PLACHTY ? POKUD ANO STOP
940 IF X <= X2 THEN 890
950 GOTO 1020
960 PRINT "SOURADNICE Y PO KOREKCI MIMO STINITKO."
970 DATA 140,10,140,110
980 DATA 95,110,105,125,190,110,180,125
990 DATA 140,25,80,105
1000 DATA 140,10,125,20

```

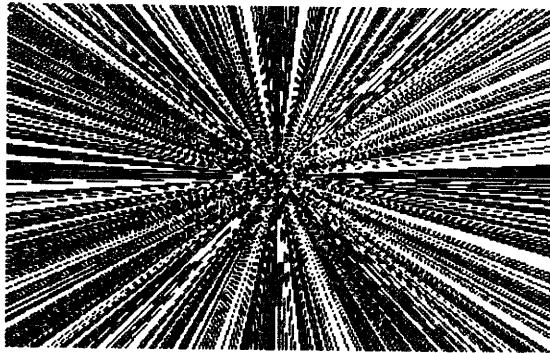
```
1010 DATA 145,20,185,100  
1020 END
```

Prog. 3-15 Nakreslení plachetnice se stínováním a korekcí rozlišovací schopnosti.

Programem Prog 3-16 vytváříte na stínítku obrazovky barevný vzor. Výsledek tohoto programu je uveden na Obr. 3-12. Kombinováním kreslení čar, barev a generátoru náhodných čísel lze vytvářet různé barevné vzory.

```
10  'PROGRAM 3-16. BAREVNE VZORY.  
20  CLS  
30  PRINT "UVED MAXIMALNI HORIZONTALNI A VERTIKALNI"  
40  PRINT "HODNOTY PRO TENTO ROZLISOVACI REZIM XM,YM:"  
50  INPUT XM,YM  
60  XC = XM / 2  
70  YC = YM / 2  
80  CLS  
90  SCREEN 1  
100 FOR X = 0 TO XM  
110     C = INT ( RND(1) * 4 )  
120     COLOR 0  
130     LINE(XC,YC)-(X,0),C  
140 NEXT X  
150 FOR Y = 0 TO YM  
160     C = INT ( RND(1) * 4 )  
170     COLOR 0  
180     LINE(XC,YC)-(XM,Y),C  
190 NEXT Y  
200 FOR X = XM TO 0 STEP -1  
210     C = INT ( RND(1) * 4 )  
220     COLOR 0  
230     LINE(XC,YC)-(X,YM),C  
240 NEXT X  
250 FOR Y = YM TO 0 STEP -1  
260     C = INT ( RND(1) * 4 )  
270     COLOR 0  
280     LINE(XC,YC)-(0,Y),C  
290 NEXT Y  
300 END
```

Prog. 3-16 Náhodné barevné vzory.



Obr. 3-12 Grafický vzor, vytvořený programem Prog. 3-16.

Obrázky mohou být tvořeny také interaktivně. Pro načrtnutí obrázku je přitom možno používat klávesnici, jako v programu Prog. 3-17. Tento program vynáší body ve směru určeném vstupem z klávesnice. Jako vstup můžeme využít i jiná vstupní zařízení, jako je světelné pero, digitalizační tabulka (tablet) nebo pákový ovladač (paddle). Tato zařízení jsou detailněji diskutována v kapitole 11. Interaktivní metoda pro konstruování obrázků z přímkových úseků je ilustrována v Prog. 3-18.

```
10      'PROGRAM 3-17. INTERAKTIVNI NACRTNUTI OBRAZKU.
20      CLS
30      PRINT "TENTO PROGRAM UMOZNUJE INTERAKTIVNE KRESLIT"
40      PRINT "OBRAZKY POMOCI KLACESNICE. PROGRAM SLEDUJE"
50      PRINT "KLAVESNICI A CEKA NA VSTUPNI SIGNAL. POKUD"
60      PRINT "ZJISTI, ZE BYLA STISKNUTA KLAVESA, NASTAVI"
70      PRINT "SOURADNICE.
75      PRINT -----
80      PRINT
90      PRINT "POUZITE KLAVESY A JEJICH VYZNAM:"
100     PRINT "      KLAVESA D      ZNAMENA KRESLIT DOLU,"
110     PRINT "                  U                      NAHORU,"
120     PRINT "                  R                      VPRAVO,"
```

```

130 PRINT " L VLEVO, "
140 PRINT " S STOP."
150 PRINT
160 PRINT "UVED MAXIMALNI HORIZONTALNI A VERTIKALNI"
170 PRINT "HODNOTY PRO TENTO ROZLISOVACI REZIM XM, YM:"
180 INPUT XM, YM
190 PRINT
200 PRINT "UVED POCATECNI SOURADNICE X,Y:"
210 INPUT X, Y
220 CLS
230 SCREEN 2
240 IF X < 0 OR X > XM OR Y < 0 OR Y > YM THEN 330
250 PSET(X,Y)
260 A$ = INKEY$: IF A$ = "" THEN 260
270 IF A$ = "D" THEN Y = Y + 1: GOTO 240
280 IF A$ = "U" THEN Y = Y - 1: GOTO 240
290 IF A$ = "R" THEN X = X + 1: GOTO 240
300 IF A$ = "L" THEN X = X - 1: GOTO 240
310 IF A$ = "S" THEN 340
320 GOTO 240
330 PRINT "SOURADNICE JE MIMO STINITKO."
340 END

```

Prog. 3-17 Interaktivní náčrtkek.

```

10 'PROGRAM 3-18. INTERAKTIVNI KONSTRUKCE OBRAZU.
20 'INTERAKTIVNI KRESLENI CAR UMOZNUJE UZIVATELI
30 'UVADET SOURADNICE POCATECNICH A KONCOVYCH BODU
40 'KAZDE CARY STEJNE JAKO BARVU TETO CARY. KRESLI
50 'OBRAZEK A PAK UMOZNI UZIVATELI ZACHOVAT NEBO
60 'SMAZAT KAZDOU CARU.
70 -----
80 DIM X1(20), Y1(20), X2(20), Y2(20), C(20)
90 CLS
100 PRINT "UVED MAXIMALNI HORIZONTALNI A VERTIKALNI"
110 PRINT "HODNOTY PRO TENTO ROZLISOVACI REZIM XM, YM:"
120 INPUT XM, YM
130 T = 1      'T JE POSET RADEK
140 CLS
150 SCREEN 1
160 PRINT "( UVED -1, -1 PRO UKONCENI )"
170 'VSTUP SOURADNICE KAZDE CARY

```

```
180 PRINT "UVED SOURADNICE X,Y PRVEHO BODU CARY:"
190 INPUT X, Y
200 IF X = -1 AND Y = -1 THEN 650
210 IF X >= 0 AND X <= XM AND Y >= 0 AND Y <= YM THEN 240
220 PRINT "PRVE SOURADNICE JSOU MIMO. OPAKUJ."
230 GOTO 180
240 X1(T) = X
250 Y1(T) = Y
260 PRINT "UVED SOURADNICE X,Y DRUHEHO BODU CARY:"
270 INPUT X, Y
280 IF X = -1 AND Y = -1 THEN 650
290 IF X >= 0 AND X <= XM AND Y >= 0 AND Y <= YM THEN 320
300 PRINT "DRUHE SOURADNICE JSOU MIMO. OPAKUJ."
310 GOTO 260
320 X2(T) = X
330 Y2(T) = Y
340 'VYBER BARVY CARY
350 PRINT "UVED KOD BARVY CARY:"
360 INPUT S
370 IF S >= 0 AND S <= 15 THEN 400
380 PRINT "KOD BARVY MUSI BYT Z INTERVALU 0 - 15."
390 GOTO 350
400 C(T) = S
410 GOSUB 570      'KRESLI OBRAZEK
420 'MA SE PONECHAT, NEBO SMAZAT ?
430 PRINT "PONECHAT, NEBO SMAZAT POSLEDNI CARU (K / E):";
440 INPUT A$
450 IF A$ = "K" OR A$ = "E" THEN 480
460 PRINT "UVADEJ POUZE K NEBO E."
470 GOTO 430
480 IF A$ <> "K" THEN 530
490 'PONECHAT CARU
500 T = T + 1      'PRIPRAVA NA DALSI CARU
510 GOTO 160
520 'SMAZAT CARU
530 T = T - 1
540 GOSUB 570      'NAKRESLI OBRAZEK BEZ POSLEDNI CARY
550 T = T + 1      'PRIPRAVA NA DALSI CARU
560 GOTO 160
570 'NAKRESLI OBRAZEK
580 CLS
```

```
590 FOR K = 1 TO T
600     COLOR 0      'NASTAV POPREDI NA ZADANOU BARVU
610     LINE(X1(K),Y1(K))-(X2(K),Y2(K)),C(K)
620 NEXT K
630 COLOR 0          'NASTAV POPREDI A POZADI NA BILOU
NA CERNE
640 RETURN
650 GOSUB 570
660 END
```

Prog. 3-18 Interaktivní návrh obrazu pomocí čar.

PROJEKTY PROGRAMOVÁNÍ

- 3-1. Napište program, který při použití příkazu PRINT vymaže stínítko obrazovky a zobrazí slovo "HELLO" (nebo libovolné jiné slovo, které si vyberete) velkými písmeny do středu stínítka obrazovky.
- 3-2. Zobrazte následující vzory pomocí programových smyček a příkazu PRINT.

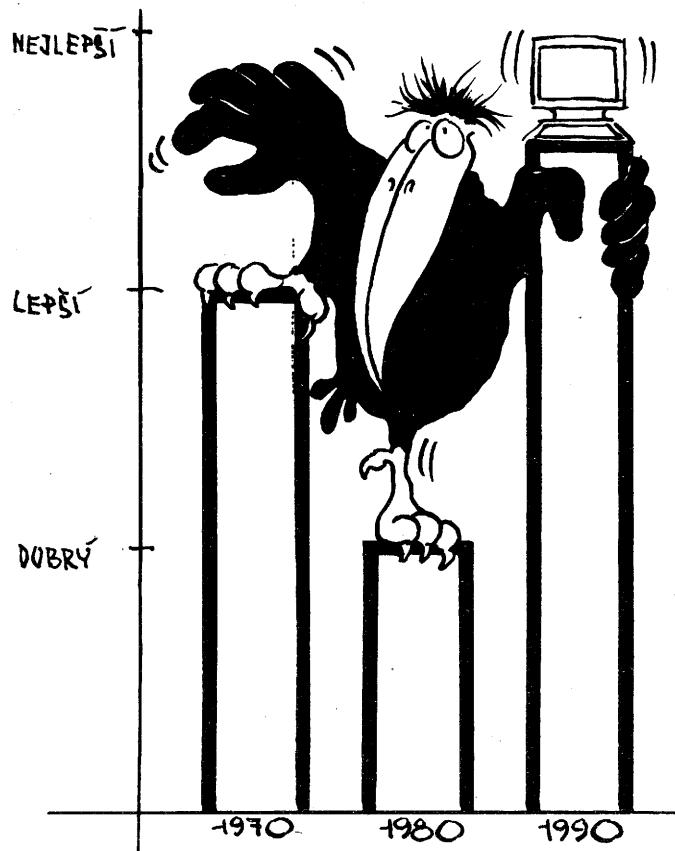
(a)	(b)
*****	@
****	@@
***	@@@
*	@@@@
***	@@@@@
****	@@@@@
*****	@@@
*****	@

- 3-3. Nakreslete na grafickém papíru návrh nějaké figury nebo scény. Vyplňte tento návrh nějakým znakem a zobrazte vzniklou siluetu na obrazovce pomocí příkazu PRINT.
- 3-4. Vytvořte siluetu z Prog. 3-3 zakódováním dat jako v Prog. 3-4.
- 3-5. Napište program pro vytvoření scény podle Obr. 3-5 pomocí speciálních znaků a příkazu PRINT.
- 3-6. Nakreslete příkazem PRINT obrázek z dat zakódovaných v příkazu DATA, který bude umožňovat několik různých znaků na jednom řádku. Data pro každý řádek mají obsahovat následující informace: (1) ASCII kód pro každý znak na řádku, (2) počáteční pozici znaku na každém řádku a (3) počet sousedních pozic obsazených každým znakem. Uvedený ASCII kód bude pro vytisknutí příslušného znaku překódován funkcí CHR\$

- 3-7. Pomocí bodové grafiky napište program, který vymaže stínítko obrazovky a nakreslí slovo velkými písmeny doprostřed obrazovky.
- 3-8. Upravte Prog. 3-11 tak, aby kreslil plnou čáru pro všechny hodnoty směrnice.
- 3-9. Upravte Prog. 3-11 tak, aby kreslil přímku s danou směrnicí a průsečíkem s osou Y, z libovolného specifikovaného bodu k okraji stínítka obrazovky.
- 3-10. Napište program, při užití podprogramu pro vynášení bodů a rovnic /3-3/, který nakreslí přímku mezi dvěma danými koncovými body.
- 3-11. Pomocí podprogramu pro kreslení přímky napište program pro zobrazení pravoúhelníku podle vstupních dat, která určují střed pravoúhelníku (XC,YC), šířku (W) a výšku (H). Pravoúhelník se má nakreslit se stranami rovnoběžnými se souřadnými osami.
- 3-12. Upravte program Prog. 3-11 tak, aby kreslil pravoúhelník o specifikované barvě a zobrazoval libovolný počet barevných, částečně se překrývajících pravoúhelníků.
- 3-13. Upravte program Prog. 3-11 tak, aby vyplnil pravoúhelník body s mezerami mezi body specifikovanými ze vstupu. Dále tento program upravte tak, aby zobrazoval libovolný počet překrývajících se, stínovaných pravoúhelníků.
- 3-14. Napište program pro zobrazení stínovaných vzorů, uvedených na Obr. 3-10.

- 3-15. Navrhnete na grafickém papíru figurku nebo scénu a napište program pro zobrazení tohoto návrhu pomocí příkazu pro kreslení přímek a stínovacích vzorů.
- 3-16. Upravte Prog. 3-17 tak, aby umožňoval interaktivní kreslení horizontálních, vertikálních a diagonálních čar.
- 3-17. Upravte Prog. 3-17 tak, aby interaktivně kreslil čáry o specifikované délce.

2. Jednoduché grafy



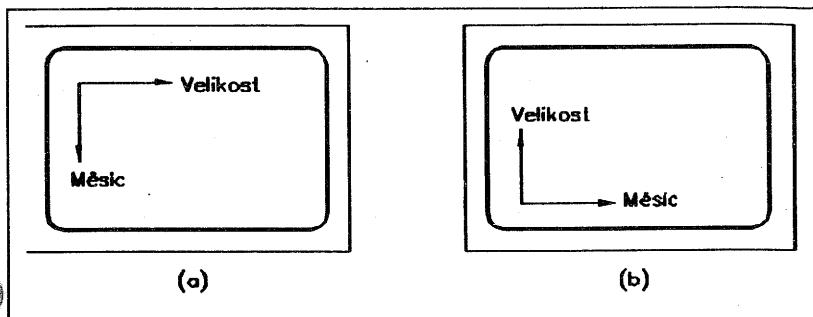
Data, uváděná v tabulce, jsou obvykle méně srozumitelná, než tatáž data vyjádřená grafickou formou. Grafy nám umožňují rychlejší pochopení informačního obsahu množiny čísel. Mohou zretelněji ukazovat různé vztahy mezi daty, které by na jednoduchém seznamu čísel (v tabulce) bylo často obtížné zjistit. Grafy můžeme konstruovat pomocí příkazu PRINT nebo pomocí speciálních grafických příkazů, definovaných v kapitole 1. Metoda, užívající příkaz PRINT, je použitelná pouze v některých případech. Grafické příkazy umožňují vytvářet podrobnější grafy s větším informačním obsahem.

2.1. Základy: grafy vývoje dat

Elementárním typem grafu je graf, který ukazuje obecný vývoj nebo "tvar" dat, s malým nebo žádným popisem. Vývojový graf pro data uvedená na Obr. 4-1 poskytuje celkový obraz obchodních změn během roku. Tento graf je možno orientovat tak, že velikost prodeje by byla reprezentována buď horizontálně nebo vertikálně.

Měsíc	Počet prodaných položek	Měsíc	Počet prodaných položek
leden	210	červenec	410
únor	150	srpen	390
březen	99	září	300
duben	250	říjen	651
květen	183	listopad	724
červen	352	prosinec	516

Obr. 4-1 Příklad tabulky obchodních dat.



Obr. 4-2 Hodnoty obchodních dat z příkladu na Obr. 4-1 mohou být vynášeny do grafu horizontálně (a) nebo vertikálně (b).

2.1.1 Metoda příkazu print

Graf vývoje dat, uvedených na Obr. 4-1, můžeme konstruovat tisknutím znaků na stínítku obrazovky v pozicích, které odpovídají velikostem hodnot. Předpokládejme, že chceme znázornit velikosti dat horizontálně na obrazovce, která má 25 řádek po 80 sloupcích (tedy formát IBM PC). Pozice znaků na každé řádce jsou počítány 1 až 80 zleva doprava, řádky jsou počítány 1 až 25 shora dolů. Měsíce bude představovat každý druhý řádek shora dolů a pozice 25 až 75 na každé řádce budou znázorňovat hodnoty prodeje. Maximální hodnota bude tedy odpovídat pozici 75 na 21. řádce shora (listopad) a minimální hodnota bude odpovídat pozici 25 na řádce 5 (březen).

Měřítko každé číarové hodnoty v intervalu 25 až 75 je určeno následujícím výpočtem:

$$Pozice = (data - \min_data) * \frac{rozsah_pozic}{rozsah_dat} + \min_pozice \quad (4-1)$$

V našem příkladu je rozsah dat 724 - 99, čili 625. Rozsah tiskových pozic je 75 - 25, čili 50, a vybrali jsme si 25 jako minimální tiskovou pozici. Horizontální pozice pro každou datovou hodnotu se pak určí ze vztahu:

$$\text{Tisková pozice} = (\text{datová hodnota} - 99) * \frac{50}{625} + 25$$

neboli

$$P = (data - 99) * .8 + 25$$

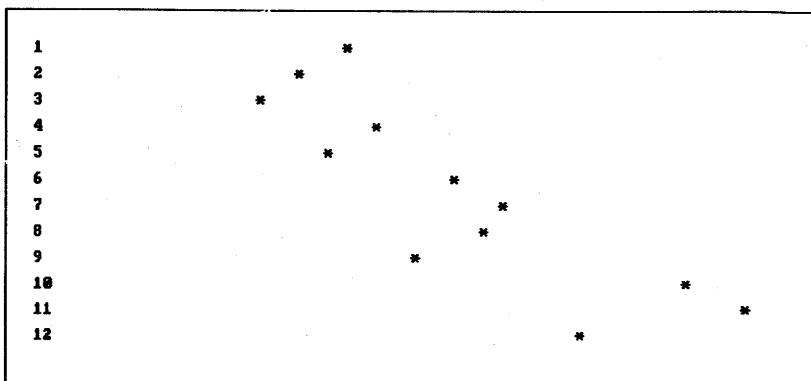
Program 4-1 užívá tento vztah pro vytvoření grafu vývoje dat (viz Obr. 4-3). Každá tisková pozice je pak zaokrouhlena na nejbližší celé číslo pomocí funkce INT. Tento program může být zobecněn pro libovolnou množinu dat a velikost displeje užitím vzorce /4-1/ při načtení minimální a maximální hodnoty dat a tiskových pozic jako vstupních údajů z klávesnice.

Pro vytvoření grafu, který znázorňuje hodnoty dat vertikálně (Obr. 4-2b), provedeme v Prog. 4-1 dvě změny. Prvou, pro každou datovou položku musíme určit příslušnou řádku, která bude odpovídat velikosti datové hodnoty.

```
10 'PROGRAM 4-1. HORIZONTALNI VYVOJ DAT POMOCI PRIKAZU
20   'PRINT. HODNOTY PRODEJE JSOU TRANSFORMOVANY DO
30   'SLOUPCU 25-75. MESICE JSOU POCITANY SHORA DOLU
35   'NA KAZDE DRUHE RADCE.
40 CLS
50 R = (75-25)/(724-99)      'R JE POMER UZIVANY V MERITKU
60 FOR K = 1 TO 12
70   READ S
80   P = INT ((S - 99) * R + 25 + .5)
90   PRINT K;TAB(P); "*"
100  PRINT
110 NEXT K
```

```
120 DATA 210, 150, 99, 250, 183, 352, 410, 390, 300, 651,  
    724, 516  
130 END
```

Prog. 4-1 Horizontální graf vývoje dat pomocí příkazu PRINT.



OBR. 4-3 Graf vývoje dat, vytvořený programem Prog. 4-1, s horizontální reprezentací hodnot prodeje z Obr. 4-1.

Druhou pozici na řádce určíme podle měsíce, kterému odpovídá příslušná datová položka. Každá datová hodnota je znázorněna tiskem znaku na pozici, určené číslem řádku a sloupce.

K určení pozice na obrazovce lze použít speciální příkazy jazyku BASIC. Tyto příkazy jsou různé na různých počítačových systémech. Definujme následující příkaz, který určí pozici na řádku a sloupci:

LOCATE R,C – Umístí kurzor na stínítku obrazovky do sloupce C a řádky R. R a C mohou být konstanty nebo výrazy. Pokud to nejsou celá čísla, budou zaokrouhlena.

Příkaz LOCATE bude používán k nastavení znakové pozice na obrazovce pro následující příkaz PRINT. Řádky

jsou číslovány od 1 pro horní řádku obrazovky do maximálního počtu pro dolní řádku obrazovky. Vztah tohoto příkazu k příkazům užívaným na různých mikropočítáčích je uveden v Dodatku A.

Nechť je pro data z Obr. 4-1 a obrazovku se 24 řádky po 40 sloupcích každý měsíc reprezentován každým třetím sloupcem, počínaje od 3. až do 36. sloupce. Hodnoty prodeje budeme vynášet na horních 20 řádkách obrazovky. Největší hodnota (724) je pak vynesena ve sloupci 33 (listopad) na řádce 1 (horní řádka).

Pozice na každé řádce je určena výrazem $3 * M$, kde M je číslo požadovaného měsíce (leden = 1, únor = 2, atd.). Pro měřítko hodnot prodeje na 20 řádkách užijeme obecný vztah:

$$\text{Řádka} = (\max_data - data) * \frac{\text{rozsah}_\text{rádek}}{\text{rozsah}_\text{dat}} + \min_\text{řád} \quad (4-2)$$

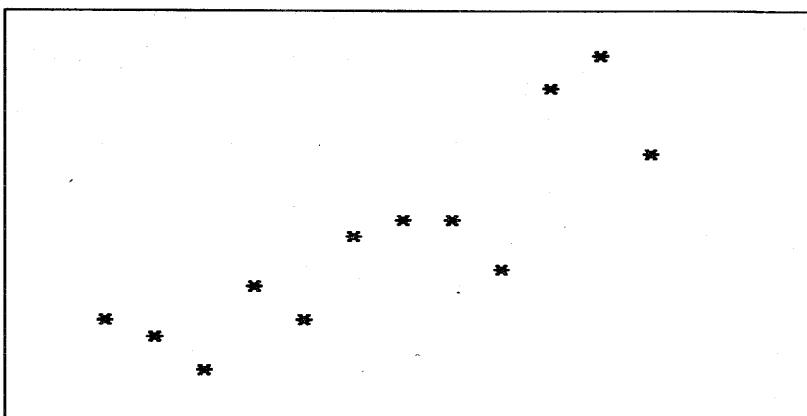
V tomto příkladu jsme zvolili rozsah řádek 19 (čili 20 - 1) a nejmenší číslo řádky 1. Obr. 4-4 ukazuje vertikální graf vytvořený programem Prog. 4-2.

```
10 'PROGRAM 4-2. VERTIKALNI VYVOJ DAT POMOCI PRIKAZU
20   'PRINT. HODNOTY PRODEJE JSOU TRANSFORMOVANY DO
30   'RADKU 1 - 20. MESICE PREDSTAVUJE KAZDY 3. RADEK
40   'POCINAJE 3. SLOUPCEM DO SLOUPCE 36.
50 CLS
60 T = ( 20 - 1 ) / ( 724 - 99 )
70 FOR M = 1 TO 12
80   READ S
90   RO = INT (( 724 - S ) * T + 1 + .5 )
100  CO = M * 3
110  LOCATE RO,CO
120  PRINT "*";
130 NEXT M
```

```
140 DATA 210, 150, 99, 250, 183, 352, 410, 390, 300, 651,  
    724, 516  
150 END
```

Prog. 4-2 Vertikální graf vývoje dat s příkazem PRINT.

Program 4-2 může být zobecněn tak, aby umožňoval proměnnou pozici grafu na obrazovce. Potom zadáváme požadovaný počet řádek a pozice sloupců na každé řádce jako vstupní data. Rozsah dat a maximální hodnota dat by byla určena programem ze vstupních dat.



Obr. 4-4 Graf vývoje dat, vytvořený programem Prog. 4-2, s vertikální reprezentací datových hodnot z Obr. 4-1.

2.1.2 Metoda bodové grafiky

Užití speciálních grafických příkazů pro vytváření grafů dat znamená, že chápeme stínítko obrazovky v pojmech souřadnic místo řádek a sloupců (znakových pozic). Vytvořme tentýž graf z Prog. 4-2 vynášením bodů příkazem PSET. Předpokládejme, že máme rozlišovací schopnost 128 bodů v horizontálním směru a 48 bodů vertikálně. Graf můžeme umístit tak, že užijeme vodorovně 0 až 40 a svisle 10 až 120.

Měsíce budou vynášeny vodorovně na každém 10. bodu, počínaje souřadnicí 10. Měřítka hodnot budou určena mezi vertikálními souřadnicemi 0 až 40 pomocí následujícího vztahu:

$$Y = (max_data - data) * \frac{rozsah_souřadnic_Y}{rozsah_dat} + min_Y \quad (4-3)$$

Program 4-3 vytváří výsledný graf vývoje dat pomocí bodů. Vynášení dat při horizontální reprezentaci datových hodnot je záležitostí výměny úloh souřadnic X a Y, při uvážení rozměrů stínítka obrazovky.

```
10  'PROGRAM 4-3. VERTIKALNI VYVOJ DAT VYNASENIM BODU.
20      'HODNOTY PRODEJE JSOU TRANSFORMOVANY DO BODU
30      '0 - 40. MESICE PREDSTAVUJE KAZDY 10. BOD,
40      'POCINAJE 10. DO 120. BODU.
50  CLS
60  SCREEN 1
70  R = (40-0)/(724-99)    'R JE POMER UZITY PRO MERITKO
80  X = 10
90  FOR K = 1 TO 12
100     READ S
110     Y = INT((724-S)*R+.5)
120     PSET(X,Y)
130     X = X + 10
140 NEXT K
150 DATA 210, 150, 99, 250, 183, 352, 410, 390, 300, 651,
    724, 516
160 END
```

Prog. 4-3 Vertikální graf vývoje dat s vynášením bodů.

Při vynášení grafů z bodů mohou být vynášené body propojeny přímkovými úseků. Takový graf (viz Obr. 4-5) vytváří program Prog. 4-4.

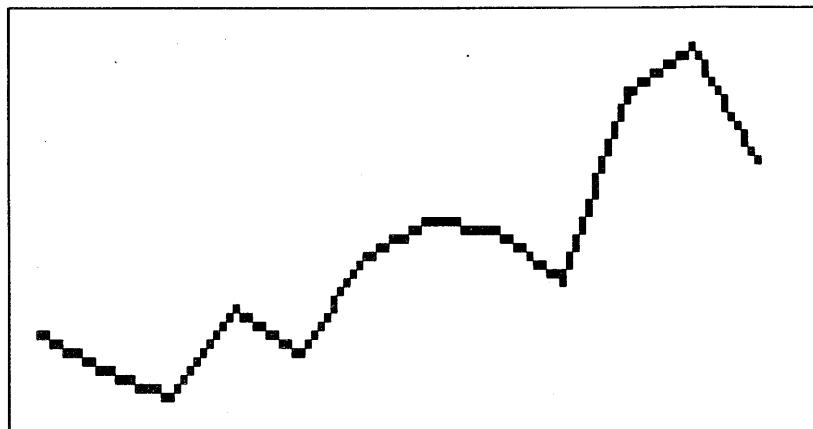
```
10  'PROGRAM 4-4. VERTIKALNI VYVOJ DAT S PROPOJOVACIMI
```

```

20      'CARAMI. HODNOTY PRODEJE JSOU TRANSFORMOVANY DO
30      'BODU 0 - 40. MESICE PREDSTAVUJE KAZDY 10.
40      'BOD, POCINAJE 10. DO 120. BODU. HODNOTY PRO
45      'SOUSEDNI MESICE JSOU PROPOJENY USECKAMI.
50  CLS
60  DIM X(12), Y(12)
70  R = (40-0)/(724-99)  'R JE POMER UZITY PRO MERITKO
80  X1 = 1
90  FOR M = 1 TO 12
100    READ S
110    Y(M) = INT (( 724 - S ) * R + .5 )
120    X(M) = X1
130    X1 = X1 + 10
140 NEXT M
150 SCREEN 1
160 FOR K = 1 TO 11
170   LINE(X(K),Y(K))-(X(K+1),Y(K+1))
180 NEXT K
190 DATA 210, 150, 99, 250, 183, 352, 410, 390, 300, 651,
    724, 516
200 END

```

Prog. 4-4 Vertikální graf vývoje dat s přímkovými úsekky.



Obr. 4-5 Graf vývoje dat, vytvořený programem Prog. 4-4, s vertikálním vynesením hodnot prodeje (Obr. 4-1) a zletězením přímkových úseků.

2.2. Grafy s popisem

Základní techniky v předcházející části jsou použitelné pro rychlé vynášení jednoduchých grafů a zobrazování vývoje dat. Grafy vývoje dat však pokryvají jen velmi málo kvalitativních informací. Obvykle potřebujeme z grafů určit přesnější informace. Označení souřadnic vynášených bodů na souřadních osách nám umožní určit přesněji vzájemné vztahy a interpolovat mezi vynášenými body.

Grafy s popisem vyžadují určitou modifikaci vztahů pro měřítka /4-1/, /4-2/ a /4-3/. Rozsah dat v těchto vztazích nyní odpovídá rozsahu grafu s popisem. Když vynášíme například množinu dat s rozsahem od -96 do 86 do grafu, popsaného od -100 do 100, je užity datový rozsah 200 (rozsah grafu). Podobně by minimální datová hodnota byla -100 a maximální 100.

Souřadnicové osy X a Y mohou být vytvářeny tiskem (PRINT) řady znaků plus (" + ") a pomlček (" - "). Takový graf s popisem je vytvářen programem Prog. 4-5 a zobrazen na Obr. 4-6.

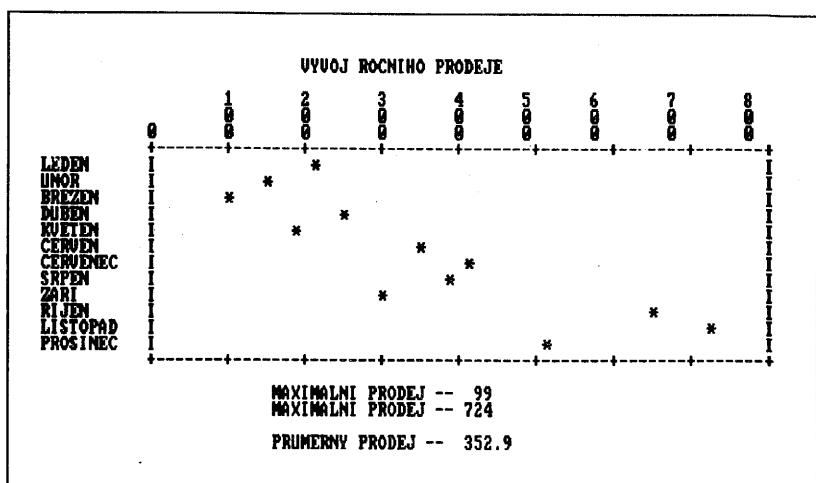
```
10 ' PROGRAM 4-5. GRAF S POPISEM VYTVORENY PRIKAZEM PRINT.
20      'HODNOTY PRODEJE JSOU TRANSFORMOVANY DO SLOUPCU
30      '12 - 76. MESICE JSOU VYNASENY DO KAZDE RADKY
35      'SHORA DOLU.
40 CLS
50 PRINT TAB(28); "VYVOJ ROCNIHO PRODEJE"
60 PRINT
70 PRINT TAB(12); "      1      2      3      4
80      5      6      7      8"
80 PRINT TAB(12); "      0      0      0      0
90 PRINT TAB(12); "0      0      0      0
          0      0      0      0"
```

```

100 PRINT TAB(12); "+-----+-----+-----+-----+
+-----+-----+-----+"
110 R = ( 76 - 12 ) / ( 800 - 0 )
120 LO = 10000
130 HI = 0
140 T = 0   'T JE SOUCET VSECH DATOVYCH HODNOT
150 FOR K = 1 TO 12
160   READ M$, S
170   IF S < LO THEN LO = S
180   IF S > HI THEN HI = S
190   T = T + S
200   P = INT (( S - 0 ) * R + 12 + .5 )
210   PRINT M$; TAB(12); "I"; TAB(P); "*"; TAB(76); "I"
220 NEXT K
230 PRINT TAB(12); "+-----+-----+-----+-----+
+-----+-----+-----+"
240 PRINT
250 PRINT TAB(25); "MAXIMALNI PRODEJ -- "; USING "###"; LO
260 PRINT TAB(25); "MAXIMALNI PRODEJ -- "; USING "###"; HI
270 PRINT
280 PRINT TAB(25); "PRUMERNY PRODEJ -- "; USING "##.##"; T/12
290 DATA "LEDEN", 210, "UNOR", 150, "BREZEN", 99, "DUBEN", 250
300 DATA "KVETEN", 183, "CERVEN", 352, "CERVENEC", 410
310 DATA "SRPEN", 390, "ZARI", 300, "RIJEN", 651, "LISTOPAD", 724
320 DATA "PROSINEC", 516
330 END

```

Prog. 4-5 Datový graf s popisem, vytvořený příkazem PRINT.



Obr. 4-6 Graf s popisem se souřadnými osami. Vytvořen programem Prog. 4-5 pomocí příkazu PRINT.

Bodová grafika nabízí pružnější metodu kreslení souřadných os kreslením přímek. Touto metodou je vytvářen graf s popisem v Prog. 4-6. Výsledný výstup je uveden na Obr. 4-7.

```

10  'PROGRAM 4-6. GRAF S POPISEM A PROPOJOVACIMI CARAMI
20      'VYTVORENY VYNASENIM BODU. PREDPOKLADANA VELI-
30      'KOST STINITKA JE 640X200 BODU S 25 RADKAMI PO
40      '80 ZNACICH. HODNOTY PRODEJE (V ROZSAHU 0-800)
50      'JSOU TRANSFORMOVANY DO BODU 27-155. MESICE
55      'UZIVAJI KAZDYCH 40 PLOSEK OD 148.
60  DIM X(12), Y(12)
70  CLS
80  SCREEN 2
90  PRINT
100 PRINT TAB(31); "HODNOTY ROCNIHO PRODEJE"
110 LINE(128,27)-(128,155)
120 LINE(608,155)-(608,27)
130     'KRESLENI VYNASECICH CAR PRO HODNOTY PRODEJE
140 FOR Y = 27 TO 155 STEP 16
150     LINE(125,Y)-(608,Y)

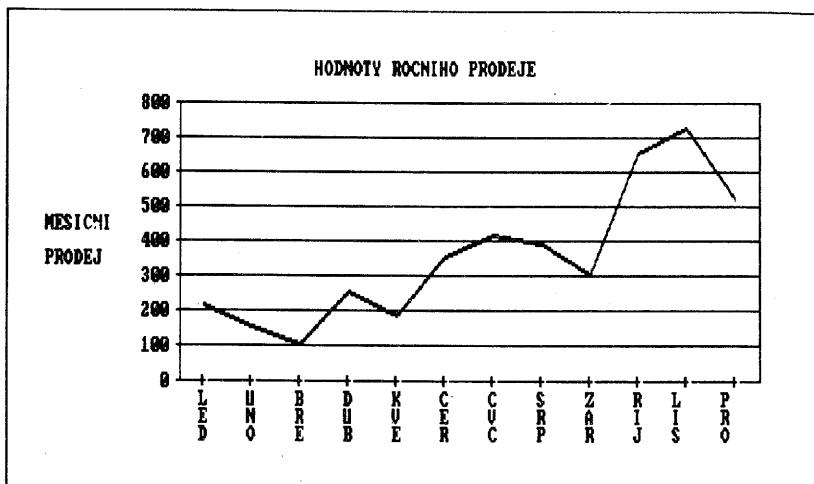
```

```

160 NEXT Y
170      'POPIS VYNASECICH CAR
180 RO = 20   'ZACATEK NA RADCE 20
190 FOR S = 0 TO 800 STEP 100
200      LOCATE RO,13
210      PRINT USING "###"; S
220      RO = RO - 2
230 NEXT S
240      'NAKRESLENI ZNACEK PRO MESICE
250 FOR CO = 19 TO 74 STEP 5
260      LOCATE 20,CO
270      PRINT "+";
280 NEXT CO
290 PRINT TAB(19); "L      U      B      D      K      C      C      S
      Z      R      L      P"
300 PRINT TAB(19); "E      N      R      U      V      E      V      R
      A      I      I      R"
310 PRINT TAB(19); "D      O      E      B      E      R      C      P
      R      J      S      O"
320 LOCATE 11,3
330 PRINT "MESICNI"
340 LOCATE 13,3
350 PRINT "PRODEJ"
360      'KRESLENI CAR GRAFU
370 T = ( 155 - 27 ) / ( 800 - 0 )
380 X1 = 148
390 FOR K = 1 TO 12
400      X(K) = X1
410      X1 = X1 + 40
420      READ S
430      Y(K) = INT (( 800 - S ) * T + 27 + .5 )
440 NEXT K
450 FOR K = 1 TO 11
460      LINE(X(K),Y(K))-(X(K+1),Y(K+1))
470      'KRESLENI CARY S DVOJNASOBNOU TLoustkou
480      LINE(X(K),Y(K)-1)-(X(K+1),Y(K+1)-1)
490 NEXT K
500 DATA 210, 150, 99, 250, 183, 352, 410, 390, 300, 651, 724, 516
510 END

```

Prog. 4-6 Graf s popisem vytvářený kreslením přímek.



Obr. 4-7 Graf s popisem a souřadnými osami. Vytvořen programem Prog. 4-6 pomocí metod bodové grafiky.

Při konstruování grafů s popisem doporučujeme používat následující pravidla. Popis by měl být jednoduchý a přesný. Příliš mnoho popisů může zmást grafy a snížit efektivitu poskytování informací. Pro jasnější popis jsou lepší větší písmena a čísla. Je-li to možné, mají být názvy umístěny v rádcích nebo v oblastech, které označují, místo uvedení ve zvláštních tabulkách nebo legendách. Dělení souřadných os, označující hodnoty, by mělo být voleno po jednoduchých krocích, jako jsou například násobky deseti. Pro usnadnění interpolace mezi vynesenými body uvádíme i souřadnice nulového bodu (počátku souřadnic). Dělení by mělo být vyznačeno značkami . Vlastní datové čáry (čárové grafy) by měly být kresleny silněji, než souřadné osy nebo pomocné přímky (vynášecí čáry apod.). Uvedená pravidla byla aplikována při konstrukci grafu na Obr. 4-7.

2.3. Sloupcové diagramy - barvy a stínování

Další technikou pro zvýšení názornosti grafů je znázornění velikostí údajů v podobě "sloupců" (úseček) místo bodů. Tato technika je ilustrována v Prog. 4-7, který užívá metody příkazu PRINT. Obr. 4-8 ukazuje výsledný sloupcový graf.

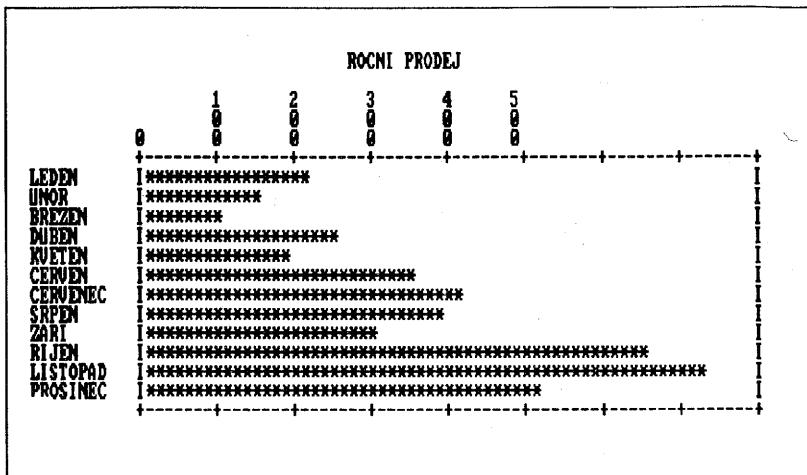
```
10  'PROGRAM 4-7. SLOUPCOVY DIAGRAM (HORIZONTALNI) S POPI-
20      'SEM VYTVARENY PRIKAZEM PRINT. HODNOTY PRODEJE
30      'JSOU VYNASENY DO SLOUPCU 12-76. MESICE ZNAZORNUJE
35      'KAZDY RADEK POCINAJE PRVYM VOLNYM RADKEM NAHORE
40      'SMEREM DOLU.
45  CLS
50  PRINT TAB(34); "ROCNÍ PRODEJ"
60  PRINT
70  PRINT TAB(12); "      1      2      3      4
     5
80  PRINT TAB(12); "      0      0      0      0
     0
90  PRINT TAB(12); "0      0      0      0      0
     0

100 PRINT TAB(12); "+-----+-----+-----+-----+-----+
-----+-----+-----+-----+"
110 R = ( 76 - 12 ) / ( 800 - 0 )
120 FOR K = 1 TO 12
130     READ M$, S
140     P = INT (( S - 0 ) * R + 12 + .5 )
150     PRINT M$; TAB(12); "I";
160     FOR C = 13 TO P
170         PRINT "*";
180     NEXT C
190     PRINT TAB(76); "I"
200 NEXT K

210 PRINT TAB(12); "+-----+-----+-----+-----+-----+
-----+-----+-----+-----+"
220 DATA "LEDEN",210,"UNOR",150,"BREZEN",99,"DUBEN",250
230 DATA "KVETEN",183,"CERVEN",352,"CERVENEC",410
240 DATA "SRPEN",390,"ZARI",300,"RIJEN",651,"LISTOPAD",724
```

```
250 DATA "PROSINEC",516  
260 END
```

Prog. 4-7 Sloupcový graf s popisem, vytvořený příkazem PRINT.



Obr. 4-8 Sloupcový diagram s popisem vytvořený Prog. 4-7 pomocí příkazu PRINT.

Příkladem na užití bodové grafiky při vytváření sloupcových diagramů je Prog. 4-8. Výsledný sloupcový graf je uveden na Obr. 4-9. V tomto diagramu je šířka vnesených sloupců větší, než mezery mezi sloupci. Tento postup se v praxi osvědčil, protože užší sloupce jsou obvykle méně účinné.

```
10  'PROGRAM 4-8. HORIZONTALNI SLOUPCOVY DIAGRAM S POPISEM  
20      'VYTVORENY POMOCI BODOVE GRAFIKY. PREDPOKLADANA  
30      'VELIKOST STINITKA 640x200 BODU, S 25 RADKAMI  
40      'A 80 ZNAKY NA RADEK. HODNOTY PRODEJE (V INTERVALU  
50      '0-800) JSOU VYNASENY DO SOURADNIC 27-155. MESICE  
60      'ZNAZORNUJE KAZDÝCH 40 BODU POCINAJE NA 136.  
65  DIM X(12), Y(12)  
70  CLS  
80  SCREEN 2
```

```

90 PRINT
100 PRINT TAB(31); "HODNOTY ROCNIHO PRODEJE"
110 LINE(128,24)-(128,163)
120 LINE(128,163)-(608,163)
130 LINE(608,163)-(608,24)
140 LINE(608,24)-(128,24)
150 'VYTVORENI ZNACEK PRO VELIKOSTI PRODEJE
160 FOR Y = 27 TO 155 STEP 8
170 LINE(125,Y)-(131,Y)
180 NEXT Y
)190 'POPISOVANI ZNACEK
200 RO = 20 'START NA RADKU 20
210 FOR S = 0 TO 800 STEP 100
220 LOCATE RO,13
230 PRINT USING "###"; S
240 RO = RO - 2
250 NEXT S
260 'VYTVORENI ZNACEK PRO MESICE
270 FOR CO = 19 TO 74 STEP 5 'START NA SLOUPCI 19
280 LOCATE 21,CO
290 PRINT "+";
300 NEXT CO
310 PRINT TAB(19); "L      U      B      D      K      C      C      S
                  Z      R      L      P"
320 PRINT TAB(19); "E      N      R      U      V      E      E      R
                  A      I      I      R"
330 PRINT TAB(19); "D      O      E      B      E      R      V      P
                  R      J      S      O";
340 LOCATE 11,3
350 PRINT "MESICNI"
360 LOCATE 13,3
370 PRINT " PRODEJ"
380 'KRESLENI SLOUPCU DIAGRAMU
390 T = ( 155 - 27 ) / ( 800 - 0 )
400 X1 = 136 'PRVY SLOUPEC ZACINA NA 136
410 FOR K = 1 TO 12
420 READ S
430 Y = INT (( 800 - S ) * T + 27 + .5 )
440 FOR X = X1 TO X1 + 24
450 LINE(X,Y)-(X,155)
460 NEXT X

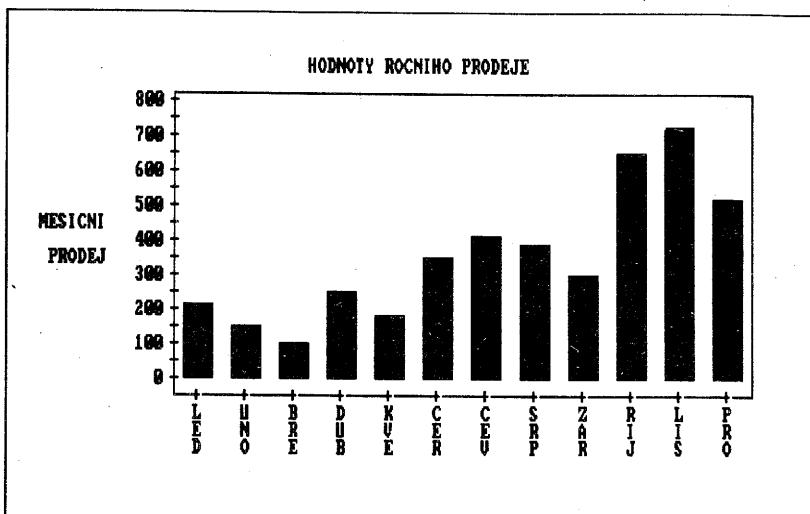
```

```

470      X1 := X1 + 40
480 NEXT K
90 DATA 210,150,99,250,183,352,410,390,300,651,724,516
500 END

```

Prog. 4-8 Sloupcový graf s popisem, vytvořený kreslením přímek.



Obr. 4-9 Sloupcový diagram s popisem, vytvořený programem Prog. 4-8 pomocí bodové grafiky.

Barvy mohou být při vytváření sloupcových grafů použity pro zvýšení názornosti nebo ke zlepšení vzhledu grafu. Výběr kombinací barev by přitom měl být pečlivě uvážen, jak bylo uvedeno v části 3-5. Použití nadměrného počtu nebo nevhodných kombinací barev může ve skutečnosti snížit působivost grafu.

Různé vzory stínování mohou být užity jak v barevných, tak v černobílých grafech. Stejně jako u barev, měli bychom vybrat vzory pro stínování tak, aby nesnižovaly působivost grafu. Neměly být používány bizarní nebo rušivé vzory by . Pokud mají být stínovány sousedící plochy, je velmi účinné

plynulé stínování od nejtmavšího po nejsvětlejší. Program 4-9 vytváří graf stínovaný na základě tohoto pravidla, jak je ukázáno na Obr. 4-10.

Grafy mohou být vytvářeny interaktivními metodami, které byly diskutovány v kapitole 3. Graf můžeme v takovém případě vytvářet na základě vstupu z klávesnice nebo pomocí světelného pera, pohybujícího se po obrazovce, nebo digitalizační tabulky. Jakmile je takový graf zobrazen na obrazovce, můžeme doplnit souřadnicové osy s popisem a zaznamenat vzniklý graf pro účely budoucího užití ve zprávách nebo hlášeních i na vnější paměť počítače. Takový graf můžeme také zvětšovat nebo zmenšovat nebo transformovat libovolným ze způsobů diskutovaných v části III.

```
10 'PROGRAM 4-9. STINOVANY SLOUPCOVY DIAGRAM POMOCI BOD.
20      'GRAFIKY. PREDPOKLADANA VELIKOST STINITKA 640x800
30      'BODU, S 25 RADKY A 80 ZNAKY NA RADEK. PRODEJNI
40      'HODNOTY (V INTERVALU 0-800) JSOU ZOBRAZENY DO
50      'SOURADNIC 36-156. KAZDE CTVRTLETI UZIVA 64 BODU
60      '(SIRKA), POCINAJE NA 160.
70      '
80 CLS
90 SCREEN 2
100 PRINT
110 PRINT TAB(28); "CTVRTLETNI PRODEJ PODLE OBLASTI"
120      'NAKRESLENI RAMECKU PRO DIAGRAM
130 LINE(128,24)-(128,162)
140 LINE(128,162)-(608,162)
150 LINE(608,162)-(608,24)
160 LINE(608,24)-(128,24)
170      'VYTVORENI ZNACEK PRO VELIKOST PRODEJE
180 FOR Y = 36 TO 156 STEP 8
190      LINE(125,Y)-(131,Y)
200 NEXT Y
210      'POPIS ZNACEK
220 RO = 20
```

```

230 FOR S = 0 TO 30 STEP 10
240     LOCATE RO,13
250     PRINT USING "##"; S
260     RO = RO - 5
270 NEXT S
280     'POPIS CTVRTLETI
290 LOCATE 23,1
300 PRINT TAB(22); " PRVNI      DRUHE      TRETI
          CTVRTE"
310 PRINT TAB(22); "CTVRTLET.   CTVRTLET.   CTVVRTLET.
          CTVRTLET.";
320 LOCATE 12,3
330 PRINT "PRODEJ"
340 LOCATE 14,2
350 PRINT "(miliony)"
360     'POPIS KRAJU (OBLASTI)
370 LOCATE 8,67: PRINT "ZAPAD"
380 LOCATE 13,67: PRINT "JIH"
390 LOCATE 18,67: PRINT "STRED"
400     /* VYTVORENI SLOUPCU, JEDNOHO PRO KAZDE CTVRTLETI */
410 T = ( 156 - 36 ) / ( 30 - 0 )
420 XL = 160   'SLOUPEC PRVEHO CTVRTLETI ZACINA NA 160
430 XR = XL + 64
440 FOR Q = 1 TO 4   'CTVRTLETI JSOU CTYRI
450     YB = 156
460     'A JE UZITO PRO KOREKCE HODNOTY YT (HODNOTA
470     'VELIKOSTI PRODEJE V MERITKU) PRO KAZDOUN OBLAST.
480     'KOREKCE JE NEZBYTNNA, PROTOZE STINOVANE PLOCHY
490     'PREDSTAVUJICI CTVRTLETNI PRODEJ PRO KAZDOUN OBLAST
495     'JSOU KLADENY NAD SEBE.
500     A = 0
510     FOR D=1 TO 3   'PRO KAZDE CTVRTLETI EXISTUJI DATA
      3 OBLASTI
520     READ S
530     'PREVOD NA MILLIONY
540     S = S / 10 ^ 6
550     YT = INT (( 30 - S ) * T + 36 + .5 )
560     'KOREKCE YT O HODNOTU A. A JE CAST, KTERA JIZ
570     'BYLA VYTVOrena PREDCHASEJICIMI HODNOTAMI
575     'PRODEJE NA OBLAST.
580     YT = YT - A

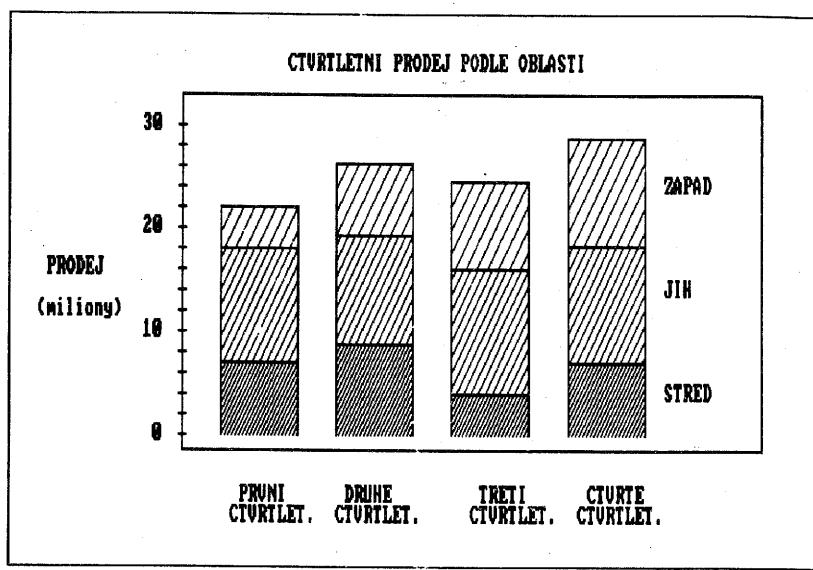
```

```

590      LINE(XL,YT)-(XL,YB)    'NAKRESLENI HRANICE
600      LINE(XR,YT)-(XR,YB)    'TETO CASTI
610      LINE(XL,YT)-(XR,YT)
620      'VYPLNENI PRAVEHO TROJUHELNIKU TETO CASTI
630      FOR X1 = XR TO XL STEP -D * 3
640          X = X1
650          Y = YT
660          PSET(X,Y)
670          Y = Y + 1
680          X = X - 1
690          IF Y <= YB AND X > XL THEN 660
700      NEXT X1
710      'VYPLNENI LEVEHO TROJUHELNIKU TETO CASTI
720      FOR Y1 = YT TO YB STEP D * 3
730          Y = Y1
740          X = XR
750          PSET(X,Y)
760          Y = Y + 1
770          X = X - 1
780          IF Y <= YB AND X > XL THEN 750
790      NEXT Y1
800      'URCENI KOREKCE POTREBNE PRO DALSI OBLAST
810      A = 156 - YT
820      YB = YT - 1
830      NEXT D
840      'PRICNY POSUV NA SLOUPEC DALSIHO CTVRTLETI
850      XL = XL + 96
860      XR = XL + 64
870      NEXT Q
880 DATA 7000000,11000000,4000000
890 DATA 8800000,10500000,7000000
900 DATA 4000000,12000000,8500000
910 DATA 7000000,11333000,10500000
920 END

```

Prog. 4-9 Stínovaný sloupcový graf, vytvořený bodovou grafikou.



Obr. 4-10 Sloupcový graf se stínováním, výstup programu Prog. 4-9.

PROJEKTY PROGRAMOVÁNÍ

- 4-1. Napište program pro zobrazení grafu vývoje dat s horizontálním vynášením datových hodnot. Užijte příkaz PRINT a rovnice /4-1/ pro umístění grafu na stínítku obrazovky. Vstupní hodnoty budou obsahovat minimální a maximální tiskové pozice a hodnoty dat. Pak bude počítán interval hodnot dat, interval tiskových pozic a tisková pozice každého vynášeného datového bodu. Uvažujte libovolný počet datových bodů.
- 4-2. Napište program pro zobrazení grafu vývoje dat s vertikálním vynášením datových hodnot pomocí příkazu PRINT. Užijte podprogram pro určení minimální a maximální hodnoty ze vstupní množiny dat, obsahující libovolný počet datových bodů. Vstupují hodnoty minimální a maximální tiskové řádky a vypočítá se interval tiskových řádek a datových hodnot. Umístění každého vynášeného datového bodu podle vztahu /4-2/.
- 4-3. Modifikujte Prog. 4-7 tak, aby vytvářel vertikální sloupcový diagram pomocí příkazu PRINT.
- 4-4. Napište program pro horizontální vnesení grafu vývoje dat pomocí bodové grafiky. Minimální a maximální hodnoty dat se určí z množiny vstupních dat. Výpočty budou obsahovat interval dat, interval bodů v horizontálním směru a souřadnice X každé datové hodnoty.
- 4-5. Modifikujte program Prog. 4-4 tak, aby se vnesené datové hodnoty propojily přímkovými úseky, tj. aby vznikla lomená křivka.

4-6. Modifikujte program Prog. 4-4 tak, aby se datové body propojily lomenou čarou z přímkových úseků a zobrazte souřadnicové osy s popisem.

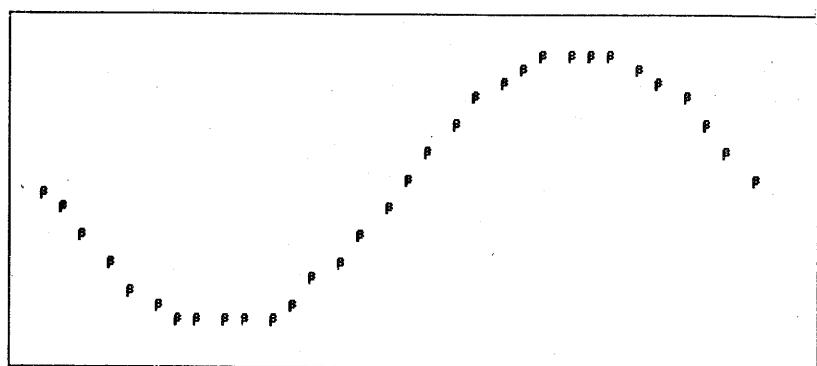
4-7. Modifikujte Prog. 4-9 tak, aby vytvářel sloupcový graf (horizontální). Zajistěte volbu barev a stínovacích vzorů, jako vstupní údaje.

4-8. Napište program pro interaktivní vytváření grafů podle vstupních dat. To znamená, že body nebo sloupce budou zobrazovány bezprostředně po uvedení příslušné hodnoty.

3. Křivky

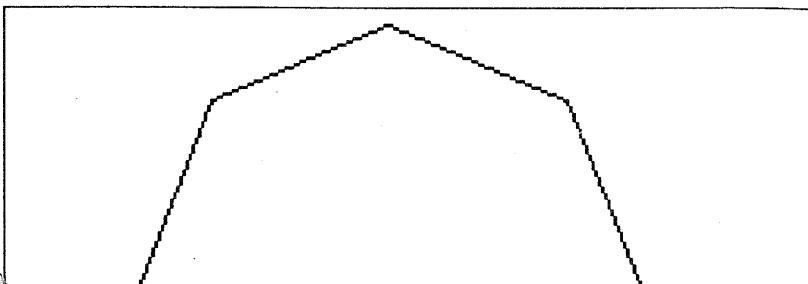


Křivky je možno zařadit do našich obrázků a grafů pomocí základních metod uvedených v předcházejících kapitolách. Aproximace tvaru křivky pomocí příkazu PRINT je uvedena na Obr. 5-1. Znakovou pozici určujeme podle rovnice křivky nebo podle nákresu křivky na grafickém papíru. Protože metoda příkazu PRINT pro zobrazování křivek má velmi omezené možnosti zaměříme se v naší diskusi na metody bodové grafiky.

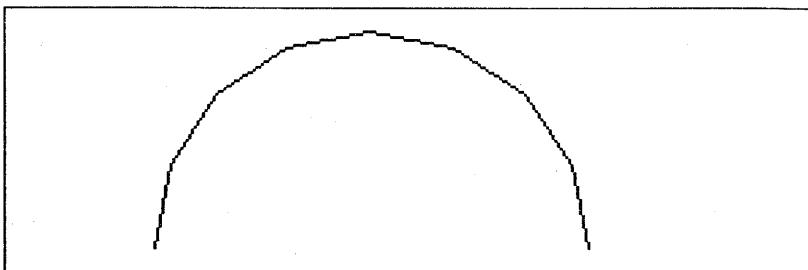


Obr. 5-1 Tištěný grafy mohou být užity pro approximaci obecného tvaru křivky.

Bodová grafika je prostředkem pro přesnější zobrazování křivek. Tvar křivky lze approximovat přímkovými úsekůmi nebo body. Čím vyšší rozlišovací schopnost má použitý systém, tím lepší approximace lze dosáhnout. Obr. 5-2 ukazuje vliv změny délky přímkových úseků na vzhled křivky. S růstem počtu přímkových úseků mezi koncovými body (X_1, Y_1) a (X_2, Y_2) se zvětšuje hladkost výsledné křivky. Nicméně, čím více přímkových úseků užijeme, tím více času systém potřebuje pro vytvoření obrazu. V některých aplikacích, např. jako je animace, potřebujeme aby byl obraz vytvořen rychle. V takovém případě se pro approximaci tvaru křivky spokojíme s menším počtem přímkových úseků a bodů.



(a)



(b)

Obr. 5-2 Aproximace křivky: (a) 4 přímkové úseky, (b) 8 přímkových úseků.

3.1. Kružnice

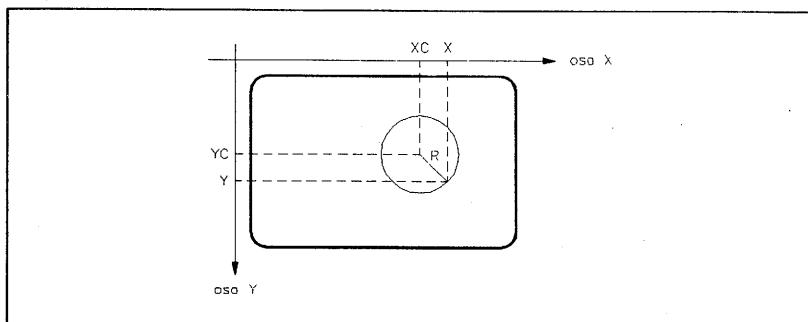
Nejběžněji užívanými křivkami jsou kružnice. Tato křivka je užívána jako základní složka pro vytváření obrazů a při zobrazování kruhových diagramů (pie charts - "koláčkové diagramy") - viz část 5-4.

Pro nakreslení kružnice na stínítku obrazovky musíme specifikovat její umístění a velikost (Obr. 5-3). Umístění je určeno souřadnicemi středu (XC, YC) kružnice. Velikost je určena poloměrem R (radius). Dále potřebujeme ještě pravidlo nebo rovnici pro stanovení souřadnic bodů na požadované kruhové dráze. Tato rovnice kružnice může být

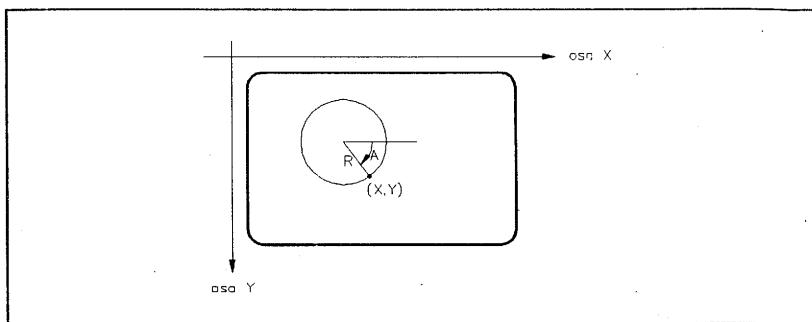
vyjádřena v několika formách. Vhodnou formou je určování souřadnic sousedních bodů (X,Y) na obvodu kruhu na základě velikosti úhlu měřené ve směru otáčení hodinových ručiček od horizontální osy (Obr. 5-4). Při užití této metody můžeme vypočítat hodnoty souřadnic X a Y z hodnot XC, YC, R a A ze vztahů

$$\begin{aligned} X &= XC + R * \cos(A) \\ Y &= YC + R * \sin(A) \end{aligned} \quad (5-1)$$

Úhel A je měřen v radiánech. Hodnota úhlu v radiánech se mění od 0 na začátku do 6,28315 ($2 * \pi$) pro jednu úplnou otáčku (tj. opět ve stejném místě, ale na konci prvej otáčky). Tyto hodnoty velikosti úhlu v radiánech odpovídají ve stupních intervalu od 0 do 360 stupňů.



Obr. 5-3 Kružnice s poloměrem R a souřadnicemi středu (XC, YC).



Obr. 5-4 Souřadnice pozic (X, Y) na kruhové dráze jsou určeny na základě poloměru R a hodnot úhlu A měřených ve směru otáčení hodinových ručiček od horizontálny.

3.1.1 Algoritmus pro kreslení kružnice z přímkových úseků

Program pro kreslení kružnic můžeme navrhnout na základě rovnic / Obr.5-1/ s approximací kružnic přímkovými úsekami. Program 5-1 požaduje vstup parametrů XC , YC , R a počet bodů, které mají být vyneseny, a vytváří výstup, jak je uvedeno na Obr. 5-5.

Kreslení urychlí zmenšení počtu přímkových úseků approximujících kružnici. Tím se však přímkové úseky prodlouží a výsledný obraz vypadá spíše jako mnohoúhelník než jako kružnice (Obr. 5-5 a).

Často můžeme také zjistit, že obrázek vypadá spíše jako elipsa. K tomu může dojít v případě, kdy jsou rozlišovací schopnosti ve směrech X a Y rozdílné. Jak bylo uvedeno v části 3-5, můžeme tento rozdíl kompenzovat korekcí hodnoty souřadnice Y . V takovém případě násobíme v programu termín $R * SIN(A)$ poměrem rozlišovacích schopností Y k X .

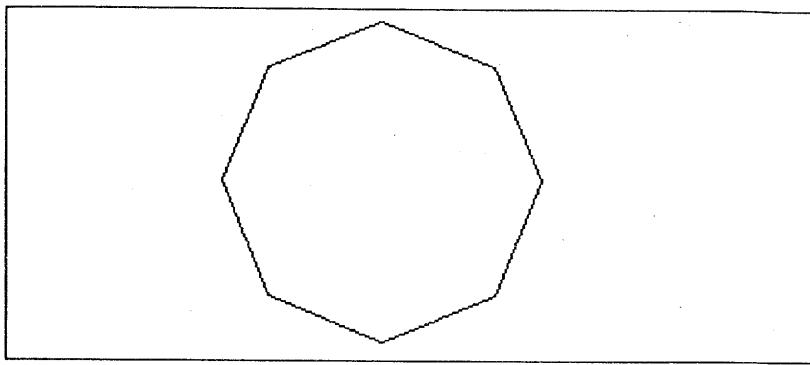
```
10 'PROGRAM 5-1. GENERATOR KRUZNIC S KRESLENIM PRIMEK.
20 CLS
```

```

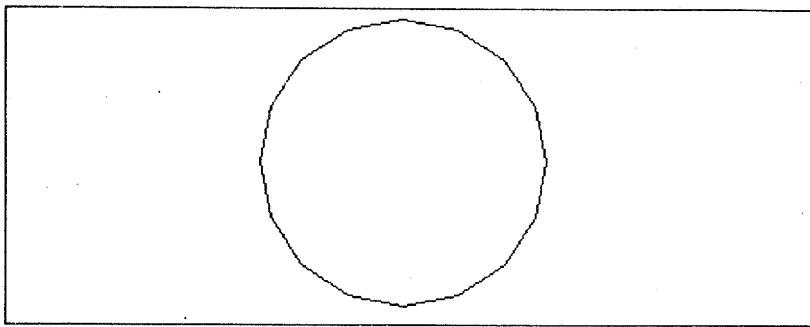
30 PRINT "MAXIMALNI HODNOTY HORIZONTALNI A VERTIKALNI"
40 PRINT "PRO TENTO ROZLISOVACI REZIM"
50 INPUT XM, YM
60 PRINT
70 PRINT "UVED SOURADNICE STREDU KRUZNICE"
80 INPUT XC, YC
90 IF XC < 0 OR XC > XM OR YC < 0 OR YC > YM THEN 350
100 PRINT
110 PRINT "UVED POLOMER KRUZNICE"
120 INPUT R
130 IF R < 0 THEN 350
140 IF XC+R>XM OR XC-R<0 OR YC+R>YM OR YC-R<0 THEN 350
150 PRINT
160 PRINT "UVED POSET BODU"
170 INPUT N
180 'RE = EKVIVALENT 360 STUPNU V RADIANECH
190 RE = 360 * 3.141593 / 180
200 DA = RE / N      'ROZDELENI KRUZNICE NA N CASTI
210 CLS
220 SCREEN 1
230 X1 = XC + R
240 Y1 = YC
250 FOR A = DA TO RE STEP DA
260     X2 = XC + R * COS(A)
270     Y2 = YC + R * SIN(A)
280     LINE(X1,Y1)-(X2,Y2)
290     'UCHOVANI NOVEHO KONCOVEHO BODU
300     X1 = X2
310     Y1 = Y2
320 NEXT A
330 LINE(X1,Y1)-(XC+R, YC)
340 GOTO 390
350 PRINT "SOURADNICE MIMO INTERVAL. UVED S PRO UKONCIENI
         NEBO R PRO OPAKOVANI"
360 INPUT C$
370 IF C$ = "S" THEN 390
380 IF C$ = "R" THEN 70
390 END

```

Prog. 5-1 Generátor kružnic, užívající metody kreslení přímek a přírůstků úhlu.



(a)



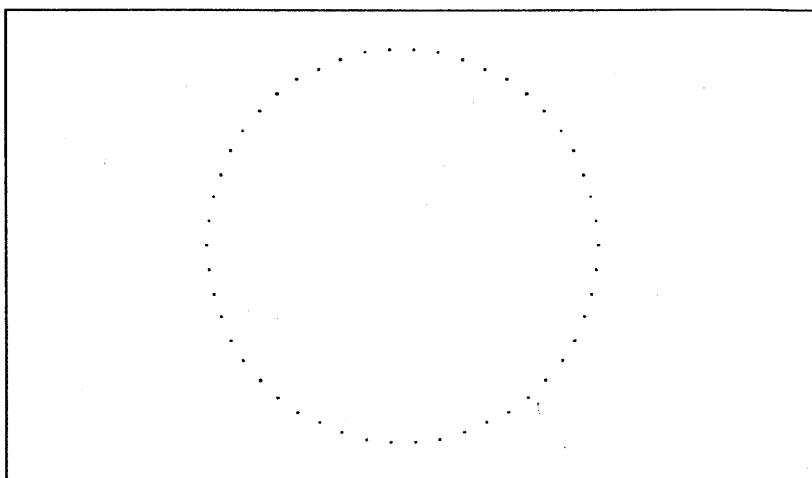
(b)

Obr. 5-5 Kružnice aproximovaná (a) 8 přímovými úseků a (b) 16 úseků.

3.1.2 Algoritmus kreslení kružnice vynášením bodů

Vynášení bodů na kružnici s vynecháním spojovacích přímkových úseků také urychlí kreslení kružnic. Ušetří se časy potřebné pro kreslení přímkových úseků a oko člověka si pokud nejsou tyto body od sebe příliš vzdálené "doplní" křivkové úseky mezi jednotlivými body (Obr. 5-6). Čím více bodů zobrazíme, tím lepší bude naše aproximace. Nejlepší možnou aproximaci získáme při vynášení bodů těsně vedle

sebe. V tomto případě jsou vynášeny body souřadnicové sítě co nejbližší k požadované kruhové dráze. Vzdálenost mezi sousedními body souřadnicové sítě bodů je jednotkou v horizontálním a vertikálním směru. Úhlová vzdálenost (v radiánech) mezi dvěma body sítě na kružnici může být tudíž approximována jako převrácená hodnota poloměru kružnice, jak je uvedeno na Obr. 5-7.

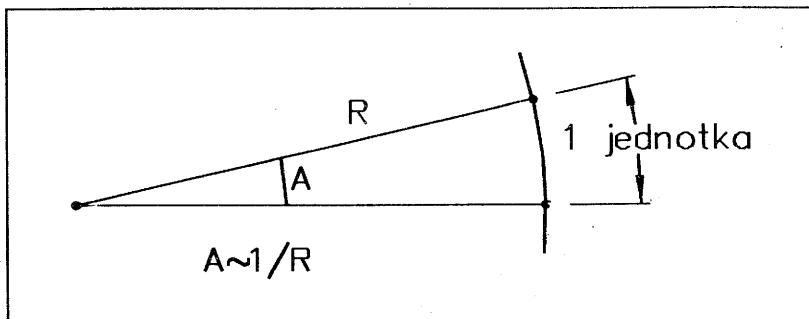


Obr. 5-6 Kružnice vynesená pomocí bodů.

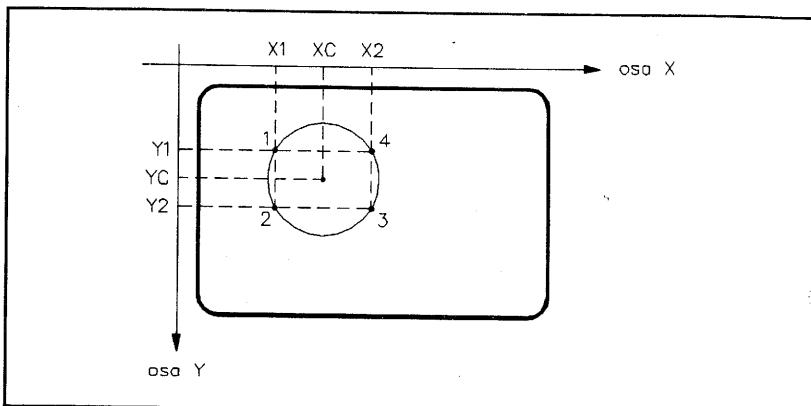
Tato approximace je pro většinu případů dostatečná. Menší přírůstky úhlu na kružnici zajistí, aby na kružnici nebyly mezery, ale některé výpočty se přitom budou provádět vícekrát, a stejně jako při vynášení odpovídajících bodů.

Další úspory času můžeme dosáhnout využitím symetrie kružnice. Nemusíme počítat každý bod kružnice. Horní polovina má stejný tvar jako dolní polovina; levá polovina má stejný tvar jako pravá. To znamená, že každá hodnota X na kružnici odpovídá dvěma hodnotám Y a každá hodnota Y

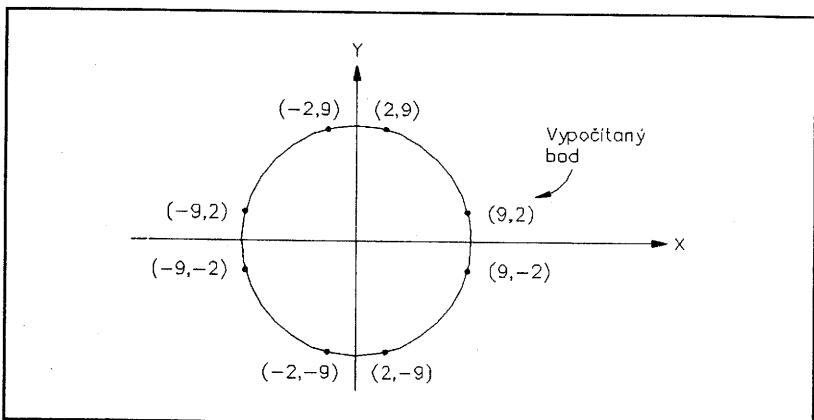
odpovídá dvěma hodnotám X, jak je uvedeno na Obr. 5-8. Dovedeme-li tuto myšlenku poněkud dále, můžeme získat čtyři body na kružnici záměnou všech souřadnic X a Y. Takže, když je bod (X,Y) na kružnici, pak je také bod (Y,X) na kružnici. Znamená to, že musíme počítat pouze body jedné osminy kruhové dráhy (oblouk 45 stupňů). Všechny zbývající body celé kružnice lze získat z těchto bodů. Obr. 5-9 ukazuje všech osm bodů kružnice, které mohou být vyneseny na základě výpočtu jediného bodu o souřadnicích (9,2). Kružnice na tomto obrázku má střed v počátku souřadnicového systému. Pro kružnici se středem v bodě (XC,YC) připočteme XC ke všem souřadnicím X a YC připočteme ke všem souřadnicím Y. Toto přičtení přesune celou kružnici na požadované místo.



Obr. 5-7 Vztah mezi úhlovou vzdáleností A (v radiánech) mezi dvěma sousedními body o vzdálenosti 1 jednotky a poloměru R kružnice procházející těmito dvěma body.



Obr. 5-8 Symetrie kružnice. Body 1 a 2 mají shodné souřadnice X; body 1 a 4 mají shodné souřadnice Y



Obr. 5-9 Je-li bod na pozici (9,2) vypočítán jako bod kružnice, pak všech osm bodů, uvedených na obrázku, může být vynesen v současně.

Program 5-2 ilustruje metodu vytváření kružnice vynášením bodů, která počítá jenom body v intervalu 0-45 stupňů. Zbyvající body jsou získány na základě symetrie. Čtyři příkazy pro vynášení bodů na začátku Prog. 5-2 můžeme vynechat spuštěním smyčky s hodnotou $A = 0$. To by vedlo k

programu s menším počtem příkazů, ale každý z těchto čtyř počátečních bodů by pak byl vynesen dvakrát. Některé body mohou být vyneseny dvakrát. K tomu dochází při větších hodnotách R a příčinou je zaokrouhlování při počítání. Tento program bude také ponechávat malé mezery na kružnicích s malými hodnotami R. To je způsobeno opět zaokrouhlováním hodnot souřadnic vynášených bodů.

```

10  'PROGRAM 5-2. GENERATOR KRUZNIC UZIVAJICI VYNASENI BODU
20      'POCITA BODY NA KRUZNICI Z 0-45 STUPNU
30          'A VYNASI VSECKY SYMETRICKE BODY.
40  CLS
50  PRINT "UVED MAXIMALNI HORIZONTALNI A VERTIKALNI"
60  PRINT "HODNOTY PRO TENTO ROZLISOVACI REZIM"
70  INPUT XM, YM
80  PRINT
90  PRINT "UVED SOURADNICE STREDU KRUZNICE"
100 INPUT XC, YC
110 IF XC<0 OR XC>XM OR YC<0 OR YC>YM THEN 440
120 PRINT
130 PRINT "UVED POLOMER KRUZNICE"
140 INPUT R
150 IF R < 0 THEN 440
160 IF XC+R>XM OR XC-R<0 OR YC+R>YM OR YC-R<0 THEN 440
170 CLS
180 SCREEN 1
190     'RA = KOREKCE ROZLISOVACI SCHOPNOSTI
200 RA = 5 / 6
210 PSET(XC+R, YC)
220 PSET(XC-R, YC)
230 PSET(XC, YC+R*RA)
240 PSET(XC, YC-R*RA)
250 DA = 1 / R
260     'RE = EKVIVALENT UHLU 45 STUPNU V RADIANECH
270 RE = 45 * 3.141593 / 180
280 FOR A = DA TO RE STEP DA
290     DX = R * COS(A)
300     DY = R * SIN(A)
310     GOSUB 340
320 NEXT A

```

```

330 GOTO 480
340      'VYNESENÍ VSECH SYMETRICKÝCH BODU
350 PSET(XC+DX, YC+DY*RA)
360 PSET(XC-DX, YC+DY*RA)
370 PSET(XC+DX, YC-DY*RA)
380 PSET(XC-DX, YC-DY*RA)
390 PSET(XC+DY, YC+DX*RA)
400 PSET(XC-DY, YC+DX*RA)
410 PSET(XC+DY, YC-DX*RA)
420 PSET(XC-DY, YC-DX*RA)
430 RETURN
440 PRINT "SOURADNICE MIMO INTERVAL. UVED S PRO STOP"
445 PRINT "NEBO R PRO OPAKUJ !"
450 INPUT C$
460 IF C$ = "S" THEN 480
470 IF C$ = "R" THEN 90
480 END

```

Prog. 5-2 Generátor kružnic, užívající vynášení bodů a přírůstků úhlu.

Dalšího zvýšení rychlosti programů pro kreslení kružnic můžeme dosáhnout použitím vztahů pro výpočet souřadnic vynášených bodů, které neobsahují funkce sinus (SIN) ani kosinus (COS). Metoda realizující tento způsob počítá pozice bodů na kružnici ze souřadnic dříve získaných bodů. Pokud souřadnice (X_1, Y_1) jsou souřadnicemi nějakého bodu na kružnici, získáme souřadnice následujícího bodu na této kružnici (X_2, Y_2) ze vztahů

$$X_2 = XC + (X_1 - XC) * CA + (Y_1 - YC) * SA \quad (5-2)$$

$$Y_2 = YC + (Y_1 - YC) * CA - (X_1 - XC) * SA$$

kde CA a SA jsou konstanty vypočítané pro zvolený přírůstek úhlu DA podle vztahů

$$CA = \cos(DA) \quad (5-3)$$

$$SA = \sin(DA)$$

Vztahy /5-2/ mohou být použity v Prog. 5-1 nebo Prog. 5-2 k určování souřadnic bodů na kružnici, aniž by se musely počítat hodnoty funkcí SIN a COS v každém kroku. Počáteční bod pro vytváření kružnice pomocí vztahů /5-2/ má souřadnice $X_1 = XC$, $Y_1 = YC + R$. Výpočty jsou dokončeny, když platí $X_2 - XC = Y_2 - YC$. Touto metodou můžeme pro každý bod vypočítaný podle vztahů /5-2/ vynášet osm bodů na kružnici. Kompenzace rozdílů rozlišovacích schopností se provádí násobením členu ($Y_1 - YC$) ve vztahu pro výpočet X_2 poměrem rozlišovacích schopností X ku Y a násobením členu ($X_1 - XC$) ve vztahu pro výpočet Y_2 poměrem rozlišovacích schopností Y ku X.

Nakonec můžeme vyloučit zaokrouhlování prováděných operací. Toho bude dosaženo užitím jednotkových přírůstků v jednom směru souřadnic místo užití úhlových výpočtů. V takovém případě použijeme výpočty, které neobsahují úhly a nevyžadují žádné zaokrouhlování. Odvození takových metod je obtížnější. Jsou založeny na myšlence, že se při hledání bodu sítě nejbližšího k požadované kružnici pohybuje ve směru osy Y (nebo X) o jednu jednotku za určitou dobu. Program 5-3 je příkladem takového postupu. Začínáme s bodem o souřadnicích (XC , $YC + R$). V každém následujícím kroku pak přidáme jedničku k souřadnici X. Tím bude určena odpovídající hodnota souřadnice Y bodu sítě nejbližšího kružnici. Tato hodnota Y bude buď shodná se současnou hodnotou, nebo bude o jednotku menší než současná hodnota. To, kterou z těchto dvou hodnot máme vynést, posoudíme podle hodnoty parametru P. Tento parametr poskytuje odhad, který z bodů je nejbližše kružnici. Je-li $P = 0$, odečteme jedničku od hodnoty souřadnice Y, jinak velikost této souřadnice ponecháme. Rychlosť zpracování programu Prog. 5-3 může být zvětšena ještě

prováděním všech výpočtů v aritmetice s celými čísly. Korekci rozdílů rozlišovacích schopností můžeme provést vynásobením vypočítané hodnoty souřadnice Y v Prog. 5-3 poměrem rozlišovací schopnosti Y ku rozlišovací schopnosti X před vyvoláním podprogramu pro vynášení bodů.

Na některých mikropočítáčových systémech je k dispozici speciální grafický příkaz pro kreslení kružnic. Zavedeme takový příkaz také do naší sady grafických příkazů:

CIRCLE(XC,YC),R – Vynese kružnici z bodů se středem (XC,YC) a poloměrem R. Hodnoty XC, YC a R mohou být konstanty nebo výrazy. Pokud jejich hodnota není celé číslo, provede se zaokrouhlení.

Tento příkaz bude použit ve všech následujících programech užívajících kreslení kružnic a budeme předpokládat, že příkaz CIRCLE vytváří kružnici podle algoritmu v Prog. 5-3. Pro ty systémy, které nemají příkaz pro kreslení kružnice, může být implementován Prog. 5-3 jako podprogram.

```
10  'PROGRAM 5-3. VYNASENI KRUZNICE PRI POUZITI JEDNOTKOVE
20      'INKREMENTACE PODEL OSY X A VYPOCET ODPOVIDAJICI
30          'HODNOTY Y.
40  CLS
50  PRINT "UVED MAXIMALNI HORIZONTALNI A VERTIKALNI"
60  PRINT "HODNOTY PRO TENTO ROZLISOVACI REZIM"
70  INPUT XM, YM
80  PRINT
90  PRINT "UVED SOURADNICE STREDU KRUZNICE"
100 INPUT XC, YC
110 IF XC<0 OR XC>XM OR YC<0 OR YC>YM THEN 530
120 PRINT
130 PRINT "UVED POLOMER KRUZNICE"
140 INPUT R
150 IF R < 0 THEN 530
```

```

160 IF XC+R>XM OR XC-R<0 OR YC+R>YM OR YC-R<0 THEN 530
170 CLS
180 SCREEN 1
190      'RA = KOREKCE ROZLISOVACI SCHOPNOSTI
200 RA = 5 / 6
210 X = 0
220 Y = R
230      'HODNOTA P INDIKUJE KTERA HODNOTA Y
240      'JE BLIZE KE KRUZNICI
250 P = 3 - 2 * R
260 GOSUB 370
270 IF P >= 0 THEN 300
280 P = P + 4 * X + 6
290 GOTO 320
300 P = P + 4 * ( X - Y ) + 10
310 Y = Y - 1
320 X = X + 1
330 GOSUB 430
340 IF X < Y THEN 260
350 IF X = Y THEN GOSUB 370
360 GOTO 570
370      'VYNESENI 4 BODU
380 PSET(XC+X, YC+Y*RA)
390 PSET(XC-X, YC-Y*RA)
400 PSET(XC+Y, YC-X*RA)
410 PSET(XC-Y, YC+X*RA)
420 RETURN
430      'VYNESENI VSECH SYMETRICKYCH BODU
440 PSET(XC+X, YC+Y*RA)
450 PSET(XC-X, YC+Y*RA)
460 PSET(XC+X, YC-Y*RA)
470 PSET(XC-X, YC-Y*RA)
480 PSET(XC+Y, YC+X*RA)
490 PSET(XC+Y, YC-X*RA)
500 PSET(XC-Y, YC+X*RA)
510 PSET(XC-Y, YC-X*RA)
520 RETURN
530 PRINT "SOURADNICE MIMO INTERVAL. STISKNI S PRO STOP,"
535 PRINT "R PRO OPAKOVANI !"
540 INPUT C$
550 IF C$ = "S" THEN 570

```

```
560 IF C$ = "R" THEN 90  
570 END
```

Prog. 5-3 Generátor kružnic s vynášením bodů a jednotkovými přírůstky X.

3.2. Ostatní křivky

Přestože je kružnice křívkou s níž se setkáváme nejčastěji, jsou v grafice používány také různé jiné křivky. Tyto křivky můžeme zobrazovat metodami podobnými metodám užitym pro zobrazování kružnic. Body na křivce (nebo tiskové pozice znaků) mohou být vypočítány z rovnic uvažované křivky a v případě potřeby korigovány na rozdíl rozlišovacích schopností. Pak můžeme vynášet body na vhodných místech stínítka obrazovky a propojit tyto body přímkovými úseky. Pro některé křivky můžeme využít symetrie nebo jiných pravidel, která redukují výpočty.

Křivky, diskutované v této části, mohou být užity pro grafické modelování citovaných aplikačních oblastí, nebo v některých případech pro doplnění tabulek dat (aproximace). Metody aproximace (jako je metoda nejmenších čtverců) umožňují reprezentaci tabulky dat křívkou.

3.2.1 Eliptické křivky

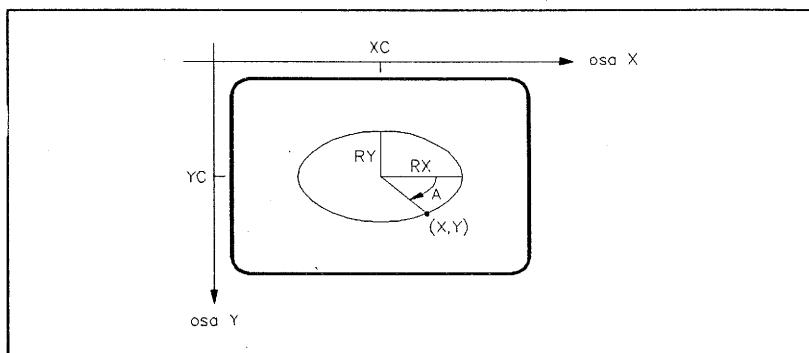
Eliptická křivka může být chápána jako variace kružnice, protože v přesném smyslu je kružnice speciálním případem elipsy. Pokud protáhneme kružnici v jednom směru (řekněme ve směru X), vznikne elipsa. Rovnice elipsy lze napsat ve tvaru:

$$X=XC+RX*\cos(A) \quad (5-4)$$

$$Y=YC+RY*\sin(A)$$

V těchto rovnicích je A úhel měřený v radiánech od horizontální osy ve směru otáčení hodinových ručiček (Obr. 5-10). Pokud je $RX > RY$, je tato elipsa delší ve směru X. Pokud $RY > RX$, je elipsa delší ve směru Y. Pro případ $RX = RY$ dostáváme kružnici.

Eliptické křivky jsou užitečné v mnoha oblastech grafického modelování. Např. některé části strojů a zařízení mají eliptické tvary, třírozměrné zobrazení válce ukáže jeho konce jako elipsy a pod. Možnost snadného zobrazení elips zvyšuje naši pružnost při vytváření zobrazení z široké oblasti grafických aplikací.



Obr. 5-10 Elipsa vynesená podle vztahů /5-4/ při $RX > RY$ a středem o souřadnicích (XC, YC) .

Programy pro kreslení kružnic, uvedené v části 5-1, mohou být modifikovány pro kreslení kružnic i elips. To je možno provést náhradou rovnic /5-1/ obecnějšími rovnicemi /5-4/. Některé mikropočítače mají možnost u příkazů pro kreslení kružnic volby elipsy.

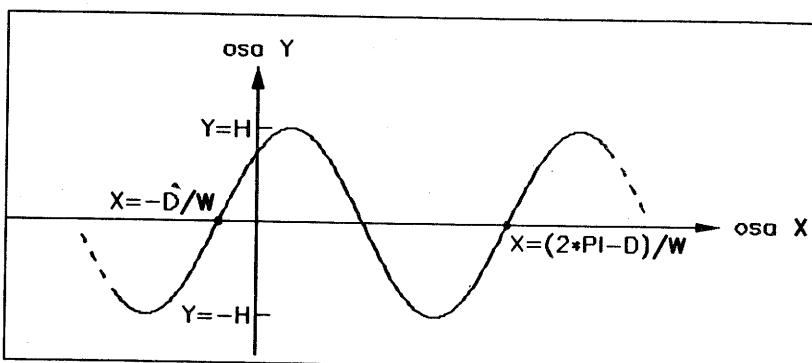
3.2.2 Sinusové křivky

Obecnou rovnici sinusové křivky lze napsat ve tvaru:

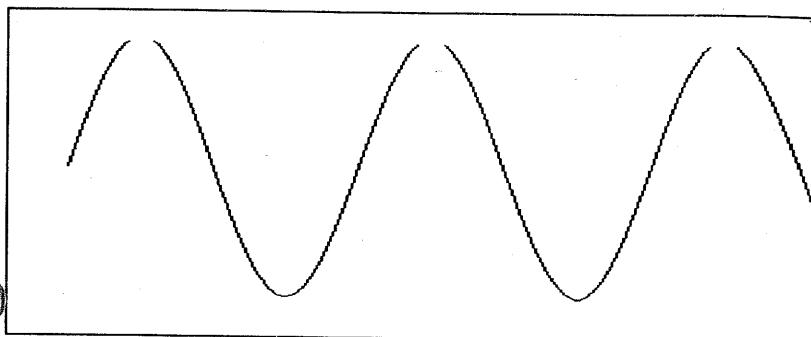
$$Y = H \cdot \sin(W \cdot X + D) \quad (5-5)$$

Obr. 5-11 ukazuje sinusovou křivku nakreslenou v obvyklém souřadnicovém systému. Frekvence W určuje počet kmitů (nebo cyklů) této křivky v daném intervalu X . Parametr D určuje přírůstek (posun) křivky vpravo nebo vlevo. Má-li D nulovou hodnotu, dostaneme standardní sinusovou křivku, a $D = \pi / 2$ vytvoří standardní kosinusovou křivku. Obr. 5-12 znázorňuje tři cykly sinusové křivky pro hodnoty parametrů $H = 50$, $W = 2 * \pi / 50$, $D = 0$ v intervalu $X = 0$ do $X = 150$. Tato křivka je vytvořena programem Prog. 5-4.

Výstup programu Prog. 5-4 je vytvořen užitím jednotkových kroků ve směru osy X a výpočtem hodnot souřadnice Y , výsledné body jsou pak propojeny přímkovými úseky. Tento program neuvažuje symetrii funkce.



Obr. 5-11 Standardní sinusová křivka podle rovnice /5-5/. Jeden kmit křivky je znázorněn v intervalu hodnot $-D/W$ a $(2\pi - D)/W$ souřadnice X , souřadnice Y osciluje mezi hodnotami maxima H a minima $-H$.



Obr. 5-12 Tři kmity sinusové křivky nakreslené programem Prog. 5-4.

```

10  'PROGRAM 5-4. SINUSOVE KRIVKY.
20  CLS
30  PRINT "PROGRAM KRESLICI SINUSOIDU POMOCI ROVNICE"
40  PRINT
50  PRINT "      Y = H * SIN ( W * X + D )"
60  PRINT
70  PRINT "UVED MAXIMALNI HODNOTU VERTIKALNI SOURADNICE"
80  PRINT "PRO TENTO ROZLISOVACI REZIM"
90  INPUT YM
100 M = YM / 2
110 PRINT "UVED VYSKU KRIVKY (H), FREKVENCI (W),"
120 PRINT "POSUV (D), H NESMI BYT VETSI NEZ POLOVINA"
130 PRINT "VYSKY STINITKA (HODNOTY YM)."
140 INPUT H, W, D
150 IF H > M THEN 110
160 PRINT "UVED MAXIMALNI A MINIMALNI HODNOTY X"
170 INPUT XL, XR
180 CLS
190 SCREEN 1
200 Y1 = H * SIN(W * XL + D)
210 IF Y1 >= 0 THEN Y1 = M - Y1
220 IF Y1 < 0 THEN Y1 = M + ABS(Y1)
230 X1 = XL
240 FOR X2 = XL TO XR
250     Y2 = H * SIN(W * X2 + D)
260     IF Y2 >= 0 THEN Y2 = M - Y2
270     IF Y2 < 0 THEN Y2 = M + ABS(Y2)

```

```

280      LINE(X1,Y1)-(X2,Y2)
290      'ZAPAMATOVANI NOVEHO BODU
300      X1 = X2
310      Y1 = Y2
320 NEXT X2
330 END

```

Prog. 5-4 Sinusová křivka vynášením bodů.

Nicméně, každá čtvrtina cyklu sinusové křivky může být opakována podle bodů mezi $X = -D/W$ a $X = (\pi/2 - D)/W$, t.j. podle prvého kvadrantu. To znamená, že pokud víme, že (X, Y) je nějaký bod na sinusové křivce v prvním kvadrantu, jsou následující body také v prvním cyklu sinusové křivky: $(\pi - X, Y)$, $(3\pi/2 - X, -Y)$, $(2\pi - X, -Y)$. Tyto symetrické body mohou být vynášeny opakováně pro všechny znázorňované cykly, aniž bychom museli přepočítávat jejich hodnoty podle rovnice /5-5/.

Sinusové křivky jsou užitečné v grafických aplikacích obsahujících kmitavý pohyb. Mezi tyto aplikace patří simulace hlasových křivek, hudba, kmity pružiny, skákání míče nebo kívání kyvadla. V případech pružiny nebo míče musíme vzít v úvahu také tření. Amplituda kmitů se pak zmenšuje s každým kmitem. Tento pokles amplitudy můžeme modelovat násobením funkce sinus ve vztahu /5-5/ exponenciální funkcí $\text{EXP}(-K \cdot X)$. Konstanta K určuje rychlosť poklesu amplitudy. Při hodnotě $K = 0.1$ bude amplituda klesat přibližně o 1/2 za jeden cyklus. Obě funkce SIN a EXP jsou k dispozici ve všech verzích jazyka BASIC.

3.2.3 Polynomiální křivky

Tato třída křivek obsahuje prakticky nekonečný seznam rovnic. Tyto rovnice mají všechny shodnou základní strukturu a patří mezi ně i rovnice přímky a paraboly. Rovnice přímky může být napsána ve tvaru

$$Y=C1*X+C2 \quad (5-6)$$

kde konstanty C1 a C2 jsou čísla, která určují směrnici (sklon) a průsečík přímky s osou Y. Tato čísla se nazývají koeficienty rovnice. Přímka je klasifikována jako polynom stupně 1. Doplněním členů, které obsahují vyšší mocniny X do této rovnice, vzniknou polynomy vyšších stupňů. Polynom stupně 2 (parabola) má tvar

$$Y=C1*Y*Y+C2*X+C3 \quad (5-7)$$

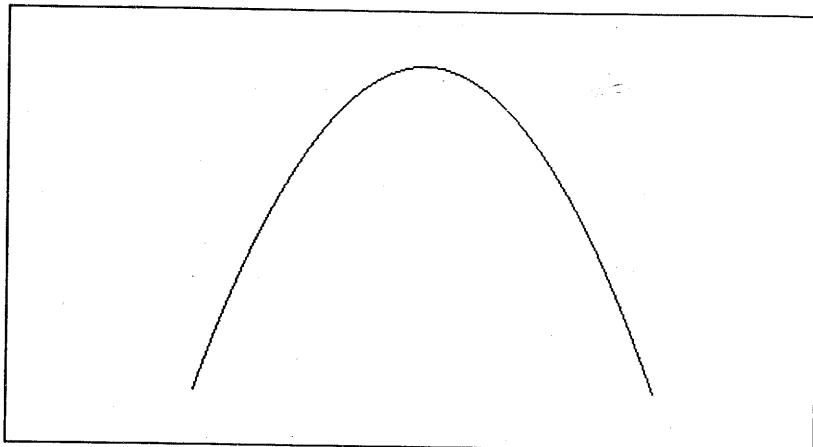
Parabola může být použita k approximaci tvaru křivek vynášených podle tabulek dat (pro vývoj dat nebo interpolační informace) nebo pro modelaci pohybu objektů. Dráha pohybu míče, hozeného na určitou vzdáenosť, je právě popsána parabolou. Míč dosáhne určité maximální výše, pak padá zpět na zem. Výška je měřena souřadnicí Y, horizontální vzdáenosť je měřena souřadnicí X. Maximální výšku dráhy míče dostaneme pro hodnotu souřadnice X ze vztahu

$$X=-\frac{C2}{2*C1} \quad (5-8)$$

Obr. 5-13 ukazuje parabolickou dráhu, pro určité hodnoty koeficientů, nakreslenou programem Prog. 5-5. Tato křivka je symetrická podle hodnoty X, uvedené v /5-8/, takže program Prog. 5-5 počítá pouze poloviční počet bodů v intervalu X.

Program 5-5 vytváří libovolnou danou parabolu, určenou koeficienty C1, C2 a C3. V závislosti na hodnotě koeficientu C1 bude parabola buď stoupat do maximální hodnoty Y, nebo klesat do minimální hodnoty Y středního bodu hodnoty X

/5-8/. Hodnota Y středního bodu bude maximum, pokud C1 > 0, nebo minimum pro C1 < 0. Program vynáší parabolu tak, že střední bod této křivky je vynesen uprostřed horní poloviny (při C1 > 0), nebo uprostřed dolní poloviny (při C1 < 0) stínítka obrazovky.



Obr. 5-13 Parabolická křivka vynesená programem Prog. 5-5.

```
10 'PROGRAM 5-5. PARABOLICKÉ KRIVKY Z POLYNOMICKÝCH ROVNIC
20   'VYNESE STŘEDNÍ ČÁST PARABOLY (TJ. KDE SE KRIVKA
30   'OBRACÍ). VYPOCITA SE VRCHOL, KORIGUJE SE TAK, ABY
40   'LEZEL V Y = 0 NEBO NA MAXIMALMI HODNOTE NA STŘE-
50   'NITKU (ZAVISI NA KRIVCE), A PAK SE VYNESE STŘED
60   'PODEL OSY X. HODNOTY Y PRO ZBYVAJICI POLOVINU
70   'KRIVKU SE POCITAJI (PRI KLESAJICIM X) A VYNASEJI
80   'SPOLU SE SYMETRICKÝMI BODY PRAVE POLOVINY KRIVKY.
90   'MEZI SOUSEDNÍMI BODY JSOU KRESLENY PRIMKOVÉ USEKY
100 '
110 CLS
120 PRINT "TENTO PROGRAM KRESLI PARABOLU PODLE ROVNICE"
130 PRINT
140 PRINT "      Y = C1 * X ^ 2 + C2 * X + C3"
150 PRINT
160 PRINT "POKUD JE C1 < 0, BUDE KRIVKA STOUPAT DO MAXIMA"
```

```

170 PRINT "A PAK OPET KLESAT. POKUD JE C1>0, BUDE KRIVKA"
180 PRINT "KLESAT DO URCITEHO MINIMA A PAK OPET STOUPAT."
190 PRINT
200 PRINT "UVED HODNOTY C1, C2, C3 (C1 NESMI BYT NULLOVA)."
210 INPUT C(1), C(2), C(3)
220 IF C(1)=0 THEN PRINT "C1 MA NULOVOU HODNOTU!": GOTO 200
230 PRINT
240 PRINT "UVED MAXIMALNI VERTIKALNI A HORIZONTALNI"
250 PRINT "HODNOTY SOURADNIC PRO TENTO ROZLISOVACI REZIM"
260 INPUT XM, YM
270 XC =INT(XM/2)    'XC JE STRED STINITKA
280     'URCENI SOURADNICE X VRCHOLU
290 X =-C(2)/(2*C(1))
300     'URCENI SOURADNICE Y VRCHOLU
310 YV = C(1) * X * X + C(2) * X + C(3)
320     'JE VRCHOL MINIMEM, NEBO MAXIMEM KRIVKY ?
330     'PRI MINIMU JE Y1 = YM, PRI MAXIMU Y1 = 0.
340 IF C(1) < 0 THEN Y1 = 0
350 IF C(1) > 0 THEN Y1 = YM
360 XL1 = XC
370 XR1 = XC
380 XL2 = XC
390 XR2 = XC
400 CLS
410 SCREEN 1
420     'VYPOCET BODU VLEVO OD VRCHOLU
430 X = X - 1
440     'PRO TUTO NOVOU HODNOTU X URCIT HODNOTU Y Z POLY-
450     'NOMICKE ROVNICE PRO Y. VYHODNOCENI POLYNOMU
460     'POMOCI NASLEDUJICI SMYCKY. TUTO METODU LZE UZIT
465     'PRO VYPOCET POLYNOMU LIBOVOLNEHO STUPNE.
470 N = 3
480 Y = C(1)
490 FOR K = 2 TO N
500     Y = Y * X + C(K)
510 NEXT K
520     'KOREKCE HODNOTY SOURADNICE Y PRED VYNESENIM
530 IF C(1) < 0 THEN Y2 = YV - Y
540 IF C(1) > 0 THEN Y2 = YM - (Y - YV)
550 XL2 = XL2 - 1
560 XR2 = XR2 + 1

```

```

570      'JSOU VSECHNY BODY JESTE NA STINITKU ?
580 IF Y2<0 OR Y2>YM OR XL2<0 OR XL2>XM THEN 650
590 LINE(XL1,Y1)-(XL2,Y2)
600 LINE(XR1,Y1)-(XR2,Y2)
610 Y1 = Y2
620 XL1 = XL2
630 XR1 = XR2
640 GOTO 430
650 END

```

Prog. 5-5 Parabola vynášením bodů.

Polynomiální křivky vyššího řádu mohou být vytvářeny úpravou programu Prog. 5-5 tak, aby pracoval s hodnotami koeficientů a rovnicemi

$$Y=C(1)*X^N+C(2)*X^{(N-1)}+\dots+C(N-1)*X+C(N) \quad (5-9)$$

Polynomiální rovnice proměnného stupně N mohou být užitečné pro approximaci dat. Pro tabulku dat můžeme kreslit hladkou datovou křivku pomocí, řekněme, tří- nebo čtyřstupňového polynomu. Vynášení polynomu může vyžadovat určité experimentování s intervalem dat. Nejprve bychom mohli vypsat seznam bodů křivky podle intervalu X, pak určit vhodné měřítka pro zobrazení křivky v určité oblasti stínítka obrazovky.

3.2.4 Normální křivky

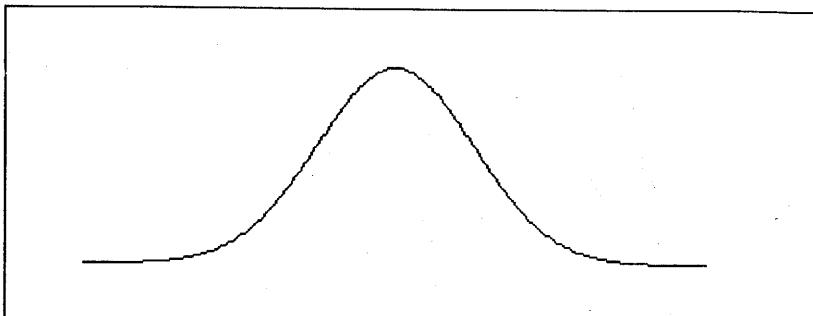
Normální, neboli Gaussova křivka (někdy nazývaná zvonová křivka), má rovnici

$$Y=\frac{EXP^{-0.5*(X-M)*(X-M)}}{S*S} \quad S*SQR(2*PI) \quad (5-10)$$

kde Y je pravděpodobnost, že X má hodnotu z určité množiny dat. M je průměrná (neboli střední) hodnota všech hodnot X a S je směrodatná odchylka.

Rovnice /5-10/ je příkladem pravděpodobnostního rozdělení. Existuje velmi mnoho jiných křivek pravděpodobnostního rozdělení, ale normální křivka má prvořadou důležitost, protože mnoho nejčastěji se vyskytujících jevů přibližně sleduje toto rozdělení pravděpodobnosti. Pravděpodobnost, že nějaký zaměstnanec velkého podniku bude mít určitý plat X , může být odhadnuta jako Y podle uvedeného vztahu. Hodnota $X = M$ znamená plat všech zaměstnanců a hodnota S určuje měřítko rozptylu platů nad a pod tímto středem. Přibližně 68 procent zaměstnanců bude mít platy v intervalu jedné směrodatné odchylky ($M-S$ do $M+S$) a přibližně 99 procent bude mít platy v intervalu tří směrodatných odchylek ($M-3*S$ do $M+3*S$). Jiné aplikace normální křivky zahrnují zobrazování pravděpodobnostního rozdělení doby životnosti elektrických a mechanických součástí, variace velikostí vyráběných položek, váhu žen v daném intervalu věku, rychlost dodávek do skladů nebo variace denních teplot v nějakém městě.

Obr. 5-14 ilustruje tvar normální pravděpodobnostní křivky. Tento obrázek byl získán z Prog. 5-6, který na základě vstupní množiny dat počítá M a S a vynáší normální křivku. Hodnoty M a S jsou počítány podle standardních rovnic. Normální křivky jsou symetrické podle středu ($X = M$), takže Prog. 5-6 může počítat pouze body pro polovinu této křivky. Křivka je umístěna uprostřed stínítka obrazovky s hodnotou Y zobrazenou mezi 50 až 150, což odpovídá čtyřem směrodatným odchylkám.



Obr. 5-14 Normální pravděpodobnostní křivka, vynesená s hodnotami $M = 51.8$ a $S = 29.2$ programem Prog. 5-6

```

10 'PROGRAM 5-6. VYNASI NORMALNI KRIVKU ODVOZENOU ZE VZOR-
20'   'KU DAT (DO 500 HODNOT). KRIVKA JE CENTROVANA NA
30'   'OBRAZOVCE A MERITKO JE URCENO TAK, ABY Y BYLO 50
40'   'AZ 150. BODY V LEVE CASTI KRIVKY JSOU VYPOCITANY
50'   'A VYNESENY. SYMETRICKE BODY JSOU VYNESENY NA
60'   'PRAVE STRANE KRIVKY.
65'   '-----'
70 CLS
80 DIM D(500)
90 PI = 3.14159
100 PRINT "UVED MAXIMALNI HODNOTU HORIZONTALNI SOURADNICE"
110 INPUT XM
120 XC = XM / 2    'XC JE STRED OSY X
130 PRINT "UVED POSET HODNOT VZORKU"
140 INPUT N
150      'URCENI PRUMERU
160 PRINT "UVED POSTUPNE VSECHNY HODNOTY"
170 T = 0
180 FOR K = 1 TO N
190     INPUT D(K)
200     T = T + D(K)
210 NEXT K
220 M = T / N      'M JE STRED DATOVYCH HODNOT
230      'VYPOCET SMERODATNE ODCHYLKY
240 T = 0
250 FOR K = 1 TO N

```

```

260      T = T + (D(K) - M) ^ 2
270 NEXT K
280 V = T / N      'V JE ODCHYLKA
290 SD = V ^ .5    'SD JE SMERODATNA ODCHYLKA
300      'URCENI STREDNIHO BODU (MAXIMA) KRIVKY
310      'Y JE MAXIMALNI, KDYZ X = M. ARGUMENT
320      'PRO EXP JE 0 A EXP(0) = 1.
330 X = M
340 YV = 1 / (SD * SQR(2 * PI))
350      'INTERVAL BODU KRIVKY (0 - YV) JE ZOBRAZEN DO
360      'INTERVALU SOURADNIC Y 50 - 150.
370 YS = (150 - 50) / (YV - 0)
380      'A DO INTERVALU 0 - XM SOURADNIC X
390 XS = (XM - 0) / (8 * SD)
400 XL1 = XC
410 XR1 = XC
420 Y1 = 50
430 CP = SD*SQR(2*PI) 'VYPOCET KONSTANTNI CASTI ROVNICE
440      'VYPOCET BODU LEVE STRANY KRIVKY. VYNESENI OBOU
445      'STRAN.
450 CLS
460 SCREEN 1
470 FOR X = M-1 TO M-4*SD STEP -1
480      Y = EXP(-.5 * (X - M) ^ 2 / V) / CP
490      'ZOBRAZENI Y DO INTERVALU 50 - 150
500      Y2 = (YV - Y) * YS + 50
510      'ZOBRAZENI X DO INTERVALU 0 - XM
520      XL2 = XC - ((M - X) * XS)
530      XR2 = XC + ((M - X) * XS)
540      LINE(XL1,Y1)-(XL2,Y2)
550      LINE(XR1,Y1)-(XR2,Y2)
560      'ULOZENI TECHTO BODU
570      XL1 = XL2
580      XR1 = XR2
590      Y1 = Y2
600 NEXT X
610 END

```

Prog. 5-6 Normální křivka vynášením bodů.

V této části jsme uvedli přehled nejčastěji užívaných křivek. Existuje mnoho jiných křivek, které mohou být

užitečné v určitých grafických aplikacích. Křivky Bezierovy a B-spline jsou užitečné pro zobrazování třírozměrných ploch, stejně tak jako pro návrh tvaru automobilů a letadel. Legendreovy a Besselovy funkce mohou být použity při modelování fyzikálních systémů, jako jsou atomové a molekulární struktury, rozdělení teplotních nebo gravitačních polí. Poissonovo a hypergeometrické pravděpodobnostní rozdělení jsou užitečná pro modelování statistických aplikací. Tyto modely zahrnují simulaci zákaznických telefonních linek s různým počtem informátorů v bance nebo znázorňování různých alternativních výběrů projekčních týmů z nějaké skupiny zaměstnanců.

3.3. Obrázky s křivkami

Programy v této části jsou příklady pro kreslení obrázků z křivek. Programem 5-7 vytvoříme obrázek Obr. 5-15, který je nakreslen z krátkých přímkových úseků podle návrhu na grafickém papíru. Obr. 5-16 ukazuje požární auto, obsahující kružnice, spirálu a normální křivku, nakreslené programem Prog. 5-8. Program 5-9 vytváří grafické "umění" pomocí křivek. Tento program demonstruje některé z mnoha možností užití trigonometrických funkcí. Výsledné vzory jsou uvedeny na Obr. 5-17.

```
10 'PROGRAM 5-7. DINOSAURUS POMOCI KRIVEK VYTVORENYCH Z
20     'PRIMKOVYCH USEKU. BODY BYLY ZISKANY Z NAVRHU
30     'NA GRAFICKEM PAPIRE. PREDPOKLADANA VELIKOST
35     'STINITKA JE 640 x 200 BODU.
40 CLS
50 SCREEN 2
60 R = 5/6
70     'NAKRESLENI JEDNE CASTI OBRAZKU
80 READ N     'NACTENI POCTU BODU CASTI
90 IF N = 0 THEN 420
100 READ X1, Y1
110 Y1 = Y1 * R     'KOREKCE Y PODLE ROZLISOVACI SCHOPNOSTI
```

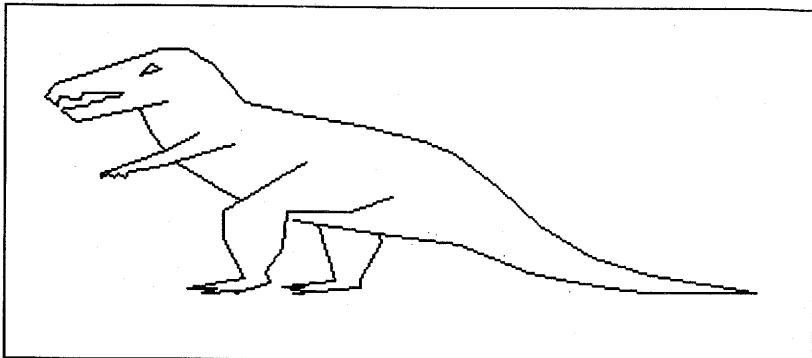
```

120 FOR P = 2 TO N
130     READ X2,Y2
140     Y2 = Y2 * R
150     LINE(X1,Y1)-(X2,Y2)
160     X1 = X2
170     Y1 = Y2
180 NEXT P
190 GOTO 80
200     'DATA PRO HLAVNI CAST TELA
210 DATA 38,223,60,180,71,173,65,174,63,179,64,180,62,185
220 DATA 62,187,60,192,60,194,57,200,57,202,55,183,55,182
230 DATA 59,172,58,171,62,165,57,170,50,220,31,233,31,245
240 DATA 40,260,60,290,67,315,72,345,80,360,87,380,105,400
250 DATA 127,425,145,452,155,502,164,450,164,434,162,415
260 DATA 158,395,153,362,138,323,132,283,125
270     'DATA PRO VELKOU NOHU
280 DATA 16,289,92,250,120,250,135,260,157,233,163,247,163
290 DATA 240,165,257,164,256,166,272,159,270,153,278,140
300 DATA 280,120,285,120,310,120,330,112
310     'DATA PRO MALOU NOHU

320 DATA 11,295,127,303,158,278,162,288,162,282,163,291
,164
330 DATA 283,165,315,162,315,157,325,137,323,132
340     'DATA PRO RUKU
350 DATA 11,238,77,220,87,192,99,192,102,195,100,198,102
360 DATA 200,100,203,102,205,98,218,96,255,83
370     'DATA PRO VYPLNENI TELA
380 DATA 3,258,114,245,105,228,93
390 DATA 3,222,86,215,77,210,65
400     'DATA PRO OKO
410 DATA 4,220,42,210,45,215,40,220,42
420 DATA 0
430 END

```

Prog. 5-7 Kreslení dinosaura křívkami, approximovanými úseky přímek.



Obr. 5-15 Obrázek nakreslený podle návrhu na grafickém papíru programem 5-7 approximací křivek přímkovými úsekky.

```
10  'PROGRAM 5-8. POZARNI AUTO Z PRIMEK, KRUZNIC, OBLOUKU,
20      'ZVONOVE KRIVKY A SPIRALY.
30      '----- KRESLENI VYSTUPU -----
40  CLS
50  SCREEN 1
60  READ X1, Y1
70  FOR K = 1 TO 8
80      READ X2, Y2
90      LINE(X1,Y1)-(X2,Y2)
100     X1 = X2
110     Y1 = Y2
120 NEXT K
130     'DVERE A OKNO
140 FOR M = 1 TO 2
150     READ N
160     READ X1, Y1
170     FOR K = 1 TO N
180         READ X2, Y2
190         LINE(X1,Y1)-(X2,Y2)
200         X1 = X2
210         Y1 = Y2
220     NEXT K
230 NEXT M
240     '---- DOKONCENI TELA, ZEBRIKU, STOJANKU ZVONU ----
250 FOR L = 1 TO 13
260     'NACTENI SOURADNIC, TLOUSTKY, SMERU TLOUSTKY
```

```

270      READ X1, Y1, X2, Y2, T, D$
280      IF D$ = "X" THEN 350
290      'VYTVORENI TLOUSTKY VE SMERU Y
300      FOR K = 0 TO T - 1
310          LINE(X1,Y1+K)-(X2,Y2+K)
320      NEXT K
330      GOTO 390
340      'VYTVORENI TLOUSTKY VE SMERU X
350      FOR K = 0 TO T - 1
360          LINE(X1+K,Y1)-(X2+K,Y2)
370      NEXT K
380      Y1 = Y2
390      NEXT L
400      '----- NAKRESLENI BLATNIKU -----
410      FOR K = 1 TO 2
420          READ XC, YC, R
430          FOR A = 3.14159 TO 6.28318 STEP 1/R
440              XW = XC + R * COS(A)
450              YW = YC + R * SIN(A)
460              PSET(XW,YW),1
470      NEXT A
480      COLOR 0
490      LINE(XC-R+1,YC)-(XC+R-1,YC),0  'VYMAZANI OBRYSU
      TELA
500      COLOR 0
510      NEXT K
520      '----- NAKRESLENI PNEUMATIK A NABOJE KOL -----
530      FOR K = 1 TO 4
540          READ X,Y,R
550          CIRCLE(X,Y),R
560      NEXT K
570      '----- NAKRESLENI HADICE -----
580      R = 1
590      A = .01
600      READ XC, YC
610      X1 = XC + R * COS(-A) * 1.4    'ZAVINUTI SPIRALY VE
620      Y1 = YC + R * SIN(-A)           'SMERU X
630      A = A + .1
640      R = R + .13
650      X2 = XC + R * COS(-A) * 1.4
660      Y2 = YC + R * SIN(-A)

```

```

670 LINE(X1,Y1)-(X2,Y2)
680 X1 = X2      'UCHOVANI STAVAJICICH BODU
690 Y1 = Y2
700 IF R < 22 THEN 630
710      ----- POPIS -----
720 LOCATE 15,33
730 PRINT "NO. 9"
740 LOCATE 19,27
750 PRINT "BMFD"
760      ----- NAKRESLENI ZVONU -----
770 READ M,SD
780 CP = SD * SQR(6.28318)
790 YV = 1 / CP
800 RY = (90 - 60) / YV
810 RX = (186 - 159) / (3.4 * SD)
820 XL1 = 166
830 XR1 = 166
840 Y1 = 60
850 FOR X = M-1 TO M-1.7*SD STEP -1
860     Y = EXP(-.5* (X - M) ^ 2 / (SD * SD)) / CP
870     Y2 = (YV - Y) * RY + 60
880     XL2 = 166 - ((M - X) * RX)
890     XR2 = 166 + (M - X) * RX
900     LINE(XL1, Y1)-(XL2,Y2)
910     LINE(XR1, Y1)-(XR2, Y2)
920     XL1 = XL2
930     XR1 = XR2
940     Y1 = Y2
950 NEXT X
960 LINE(XL1,Y1)-(XR1,Y1)
970 CIRCLE(166,Y1+2),3      'SRDCE ZVONU
980      -----
990 DATA 12,164,313,164,295, 100,251,100,237,60,194,60,194
1000 DATA 92,27,92,27,164,5,201,68,201,156,245,156,245,100
1010 DATA 233,68,201,68,4,205,73,205,100, 240, 100, 230,
    73, 204
1020 DATA 73,194,60,194,164,1,X,34, 100, 187, 100, 3, Y,
    34, 124
1030 DATA 187,124,3,Y,55,103,55,124,3,X,76, 103, 76, 124,
    3, X

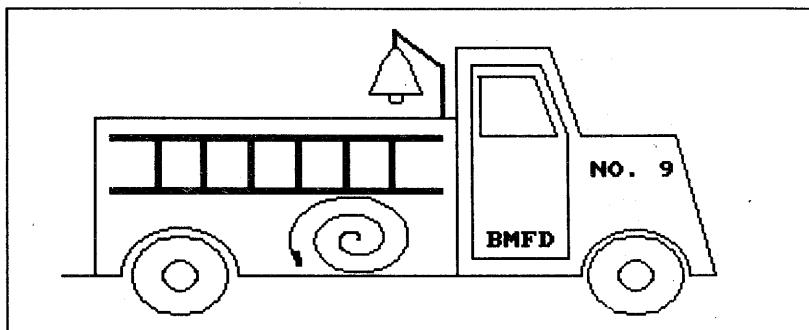
```

```

1040 DATA 98,103,98,124,3,X,120,103,120,124, 3, X, 142,
    103, 142
1050 DATA 124, 3, X, 163, 103, 163, 124, 3, X, 187, 92,
    187, 68, 2, X, 187
1060 DATA 68, 165, 52, 2, Y, 165, 52, 165, 60, 2, X, 119,
    153, 120, 159, 3
1070 DATA X, 66, 164, 27, 276, 164, 25, 66, 164, 22, 276,
    164, 22, 66, 164
1080 DATA 8,276,164,8,147,147,1,7
1090 END

```

)Prog. 5-8 Kreslení požárního auta rovnicemi křivek.



Obr. 5-16 Obrázek nakreslený programem Prog. 5-8 pomocí rovnic křivek.

```

10 'PROGRAM 5-9. VZORY Z KRIVEK.
20 CLS
30 XM = 319
40 YM = 199
50 SCREEN 1
60 '----- JETEL -----
70 XC = 160   'STRED OBRAZOVKY
80 YC = 100
90 FOR R = 20 TO 50 STEP 15
100     X1 = XC
110     Y1 = YC
120     FOR A = 0 TO 6.28318 STEP 1/R
130         R1 = R * SIN(2 * A)
140         X2 = XC + R1 * COS(A)
150         Y2 = YC + R1 * SIN(A)

```

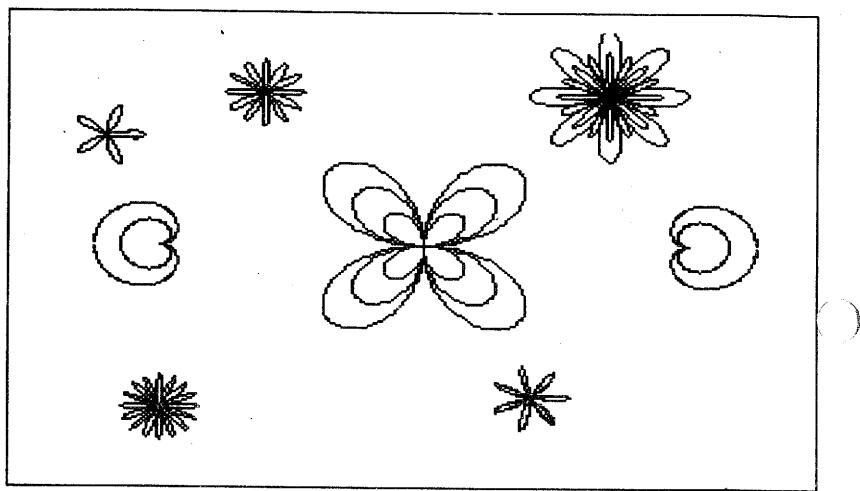
```

160      LINE(X1,Y1)-(X2,Y2)
170      X1 = X2      'ULOZENI POSLEDNICH BODU
180      Y1 = Y2
190      NEXT A
200 NEXT R
210
220      '----- STRANOVE OBRAZCE (KARDIOIDY) -----
230 XL = 60   'STRED LEVEHO OBRAZCE JE XL,YC
240 YC = 100
250 XR = 260 'STRED PRAVEHO OBRAZCE JE XR,Y
260 FOR R = 15 TO 25 STEP 10
270     XL1 = XL
280     XR1 = XR
290     YL1 = YC
300     YL3 = YC
310     YR1 = YC
320     YR3 = YC
330     FOR A = 0 TO 3.14159 STEP 1/R
340         R1 = R * SIN(A / 2)
350         DX = R1 * COS(A)
360         DY = R1 * SIN(A)
370         XL2 = XL + DX
380         YL2 = YC + DY
390         YL4 = YC - DY
400         LINE(XL1,YL1)-(XL2,YL2)
410         LINE(XL1,YL3)-(XL2,YL4)
420         XR2 = XR - DX
430         YR2 = YC + DY
440         YR4 = YC - DY
450         LINE(XR1,YR1)-(XR2,YR2)
460         LINE(XR1,YR3)-(XR2,YR4)
470         XL1 = XL2
480         YL1 = YL2
490         YL3 = YL4
500         XR1 = XR2
510         YR1 = YR2
520         YR3 = YR4
530     NEXT A
540 NEXT R
550
560      '----- KVETINOVE VZORY -----

```

```
570 READ XC,YC,R,P
580 IF XC = 0 THEN 820
590 GOSUB 620
600 GOTO 570
610 GOTO 820
620 ----- KRESLENI KVETINOVYCH VZORU -----
630     X1 = XC + R
640     Y1 =YC
650     FOR A = 0 TO 6.28318 STEP 1/R
660         R1 = R * COS(P * A)
670         X2 = XC + R1 * COS(A)
680         Y2 = YC + R1 * SIN(A)
690         LINE(X1,Y1)-(X2,Y2)
700         X1 = X2
710         Y1 = Y2
720     NEXT A
730 RETURN
740 -----
750 DATA 200,30,15,7
760 DATA 100,170,15,6
770 DATA 60,25,15,8
780 DATA 40,150,14,5
790 DATA 230,169,20,8
800 DATA 230,169,30,4
810 DATA 0,0,0,0,0,0,0
820 END
```

Prog. 5-9 Umělecké vzory z křivek.



Obr. 5-17 Umělecké vzory z křivek, vytvořené programem Prog. 5-9.

3.4. Grafy a kruhové diagramy

Křivky jsou použitelné pro mnoho typů grafů a diagramů. Programy, uvedené v této části, ilustrují techniky kreslení grafů pomocí křivek.

3.4.1 Grafy

Grafy můžeme zobrazovat podle bodů, jejichž hodnoty jsou dány tabulkou dat, nebo vypočítány z rovnic. Při znázorňování tabulky dat můžeme vynášet udané body a propojit je přímkovými úseky nebo můžeme použít některou z metod approximace křivek. Křivky jsou pak kresleny tak, aby procházely určitou množinou dat, pomocí nějaké analytické techniky (jako je metoda nejmenších čtverců). Přitom je možno použít interaktivní metodu, která kreslí křivku podle příkazů zadaných z klávesnice, nebo na základě vstupních informací ze světelného pera. Při grafickém znázorňování

rovníc počítáme souřadnice podle rovnice a bud' vynášíme jednotlivé body těsně vedle sebe, nebo ve větších vzdálenostech propojené přímkovými úseky.

Příkladem vynášení křivky podle libovolné specifikované rovnice je program Prog. 5-10. Rozměry grafu pro zvolenou křivku jsou vstupními hodnotami programu. Obr. 5-18 ukazuje výstup Prog. 5-10 pro případ třístupňového polynomu.

```
10 'PROGRAM 5-10. VYNESENI LIBOVOLNE ROVNICE.
20   'UMOZNUJE UZIVATELI NAPSAT ROVNICI NA RADEK 450 A
30   'ZADAT MINIMALNI A MAXIMALNI HODNOTY X A Y PRO
40   'KTERE MA BYT ROVNICE VYNESENA. PROGRAM KRESLI
50   'SOURADNOU SIT PRO PLOSKY 74-574 NA OSE X A 12-188
60   'NA OSE Y.
65   -----
70 CLS
80 PRINT "CHCETE INSTRUKCE ? (A/N)"
90 INPUT I$
100 IF I$ = "N" THEN 220
110 PRINT "TENTO PROGRAM ZOBRAZUJE GRAF LIBOVOLNE ROVNICE"
120 PRINT "V SOURADNE SITI NA STINITKU OBRAZOVKY. ROVNICE"
130 PRINT "MUSI BYT ZAPSANA NA RADEK 450 TOHOTO PROGRAMU."
140 PRINT "VASE ROVNICE MUSI UZIVAT PROMENNOU Y JAKO ZA-"
150 PRINT "VISLOU A X JAKO NEZAVISLOU (NAPR. Y = 6*X + 20)"
160 PRINT "BYLA-LI VASE ROVNICE ZAPSANA A VY JSTE NAPSAL"
170 PRINT "RUN, BUDETE VYZVAN K UVEDENI MINIMALNI A MAXI-"
180 PRINT "MALNI HODNOTY SOURADNICE X, PRO KTERE CHCETE"
190 PRINT "ROVNICI ZNAZORNIT, DALE MINIMALNI A MAXIMALNI"
200 PRINT "HODNOTY SOURADNICE Y, KTERE MAJI BYT ZOBRAZENY"
210 PRINT "NA SOURADNE SITI."
215 PRINT
220 PRINT "OBSAHUJE RADEK 450 VASI ROVNICI ? (A/N)"
230 INPUT E$
240 IF E$ = "A" THEN 300
250 IF E$ = "N" THEN 780
```

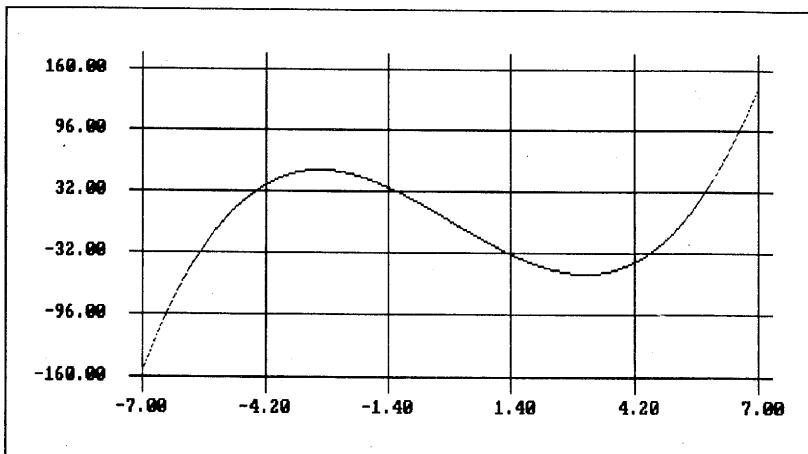
```
260 PRINT
270 PRINT "NAPIS CISLO RADKY 450 A VASI ROVNICI."
280 PRINT "STISKNETE KLAVESU RETURN A NAPISTE RUN."
290 GOTO 780
300   'GRAFICKA ROVNICE NA MRIZCE
310 PRINT
320 PRINT "VLOZTE MINIMALNI A MAXIMALNI HODNOTY X PRO
      KTERE"
330 PRINT "SE MA ROVNICE KRESLIT."
340 INPUT XL,XR
350 PRINT
360 PRINT "VLOZTE MINIMALNI A MAXIMALNI HODNOTY Y KTERE"
370 PRINT "SE MAJI OBJEVIT NA MRIZCE."
380 INPUT YB,YT
390 GOSUB 520      'NAKRESLI MRIZKU
400           'KRESLI ROVNICI
410 XA = (564 - 84) / XD
420 YA = (180 - 20) / YD
430 FOR XG = 84 TO 564
440   X = XL + (XG - 84) / 484 * XD
450 Y=X^3-27*X
460   YG = 180 - (Y - YB) * YA
470   IF YG < 20 OR YG > 180 THEN 490
480   PSET(XG,YG)
490 NEXT XG
500 LOCATE 1,1
510 GOTO 780
520   'KRESLI MRIZKU S POPISEM
530 CLS
540 SCREEN 2
550 FOR Y = 20 TO 180 STEP 32
560   LINE (74,Y)-(574,Y)
570 NEXT Y
580 FOR X = 84 TO 564 STEP 96
590   LINE(X,12)-(X,188)
600 NEXT X
610 XD = XR - XL
620 YD = YT - YB
630 R = 23
640 FOR K = 0 TO 5
650   LOCATE R,1
```

```

660      S = YB + YD * K / 5
670      PRINT USING "####.##";S
680      R = R - 4
690 NEXT K
700 C = 7
710 FOR K = 0 TO 5
720      LOCATE 25,C
730      S = XL + XD * K / 5
740      PRINT USING "####.##";S;
750      C = C + 12
760 NEXT K
770 RETURN
780 END
)

```

Prog. 5-10 Kreslení obecného grafu vynášením bodů pro libovolnou vstupní rovnici.



Obr. 5-18 Graf funkce $X^{3-27}X$, výstup programu Prog.5-10.

3.4.2 Kruhové diagramy

Algoritmus kreslení kružnic je hlavní složkou programu pro tvorbu kruhových diagramů (pie charts - koláčové diagramy). Program 5-11 právě ilustruje tuto aplikaci. Vstupní data tohoto programu zahrnují jméno a relativní velikost

(datovou hodnotu) pro každou část diagramu. Umístění jména části na diagramu je určeno úhlem odpovídajícím ose příslušné části (sektoru). Pro umístění označení platí následující pravidlo. Pokud je sektor v pravé části kruhového diagramu, začíná jeho jméno na půlící ose sektoru 4 jednotky za obvodem kruhu. Pokud je sektor nalevo, končí jeho jméno na ose sektoru. Pozice začátků popisu zajistíme přepočtem souřadnic z pixelů na znakové pozice. To snadno provedeme dělením souřadnice X počtem horizontálních pixelů v jednom znaku a dělením souřadnice Y počtem pixelů v jednom znaku ve vertikálním směru. Obrázek 5-19 ukazuje kruhový diagram vytvořený programem Prog. 5-11.

Mají-li být kruhové diagramy srozumitelné, neměli by být kresleny s větším počtem částí (sektorů) než 5 až 6. Stínování a vybarvení částí kruhových diagramů je podobné jako u sloupkových grafů. Stínovací vzory mají být jednoduché a odstupňované od tmavých po světlé kolem kruhu. Popisy umísťujeme co nejbliže plochám, které označují.

Druh grafu nebo diagramu, který zvolíme pro znázornění dat, má mít značný vliv na srozumitelnost zobrazení. Kruhové diagramy jsou zpravidla výhodnější pro vyjádření informací o procentech. Čárové grafy a sloupkové diagramy jsou vhodnou volbou pro vyjádření informací o datových množstvích, jako jsou např. nákupy, prodeje, stav vásob a podobně. Můžeme také kreslit grafy a diagramy v závislosti na čase nebo v poměru k jinému parametru. Obecně lze říci, že čas je nevhodnější zobrazovat na horizontální ose, v případě vztahu jiných parametrů je třeba respektovat obecných zvyklostí a pamatovat na přehlednost grafu či diagramu. Složitější datové vztahy mohou být znázorňovány uvedením více křivek v grafu, překrýváním sloupců u sloupkového diagramu nebo užitím třírozměrného grafu.

```

10 'PROGRAM 5-11. POPSANY KOLACOVY DIAGRAM.
20 CLS
30 DIM N$(8), V(8)
40 PRINT "VLOZTE MAXIMALNI HORIZONTALNI A VERTIKALNI"
50 PRINT "HODNOTY PRO TENTO ROZLISOVACI REZIM."
60 INPUT XM,YM
70 PRINT "VLOZTE POSET ZNAKU, KTERE MOHOU BYT"
80 PRINT "TISKNUTY NA RADKU OBRAZOVKY V TOMTO"
90 PRINT "ROZLISOVACIM REZIMU"
100 INPUT C1
)110 PC = XM / C1           'PC JE POSET BODU NA ZNAK
    HORIZONTALNE
120 PRINT "VLOZTE POSET DOSTUPNYCH RADEK"
130 PRINT "V TOMTO ROZLISOVACIM REZIMU"
140 INPUT R1
150 PR = YM / R1           'PR JE POSET BODU NA ZNAK
    VERTIKALNE
160 PRINT
170 PRINT "VLOZTE SOURADNICE STREDU KOLACE"
180 INPUT XC,YC
190 PRINT "VLOZTE POLOMER"
200 INPUT R
210 IF XC+R>XM OR XC-R<0 OR YC+R>YM OR YC-R<0 THEN 830
220 PRINT
230 PRINT "VLOZTE POSET ODDILU (MAX. 8)"
240 INPUT N
250 PRINT "VLOZTE NAZEV DIAGRAMU"
260 INPUT T$
270 PRINT "VLOZTE NAZEV A HODNOTU PRO KAZDOU CAST
280 T = 0
290           'VSTUP DAT
300 FOR K = 1 TO N
310     INPUT N$(K), V(K)
320     T = T + V(K)
330 NEXT K
340 CLS
350 SCREEN 1
360 LOCATE 1, INT(C1/2-.5*LEN(T$))
    'VYCENTROVAT TITULEK
370 PRINT T$
380 CIRCLE (XC,YC),R

```

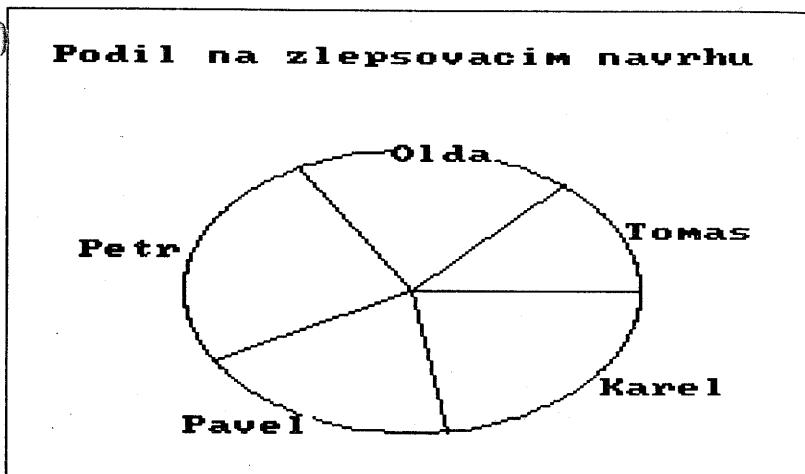
```

390 YA = 5 / 6          'YA JE OPRAVA PRO ROZLISENI
400 B = 0                'B JE UHEL KTERY DETERMINUJE
    PREDCHOZI RADKA
410 S = 0
420 RE = 360*3.14159/180   'RE JE EKVIVALENT 360 STUPNU V
    RADIANECH,
430 FOR K = 1 TO N
440      'PRIMKA, KTERA SE MA VYNASET, JE URCENA HODNO-
450      'TOU TETO A PREDCHAZEJICI CASTI,
460      S = S + V(K)
470      'UHEL JE URCEN PROCENTEM KRUHU,
480      'KTERE ODPOVIDA POMERU S / T
490      A = RE * S / T
500      XP = XC + R * COS(A)
510      YP = YC + R * SIN(A) * YA
520      LINE(XC,YC)-(XP,YP)
530      'ZAPIS NAZVU CASTI
540      'URCENI BODU O 4 JEDNOTKY DALE OD STREDU
550      'OBLOUKU PATRICIHO TETO CASTI
560      AC=B+(A-B)/2   'AC JE UHEL OD HORIZONTALY
570      XL =XC+(R+4)*COS(AC)  'KTERY BY ROZPULIL TUTO CAST
580      YL = YC +(R + 4) * SIN(AC) * YA
590      '(XL,YL) JE BOD UZITY PRO ZAPIS NAZVU CASTI
600      'TENTO BOD SE UZIJE JAKO POCATECNI BOD NAZVU,
610      'JE-LI VPROVO, NEBO JAKO KONCOVY BOD, JE-LI
620      'VLEVO. JAKO STREDNI BOD NAZVU, JE-LI NAHORE
625      'NEBO DOLE.
630      'BOD JE ZACATKEM NAZVU
640      IF XL > XC + 10 THEN 740
650      'BOD JE KONCEM NAZVU
660      IF XL < XC - 10 THEN 710
670      'JINAK JE BOD STREDEM NAZVU. KOREKCE XL PODLE
680      'POLOVICKY POCTU BODU NA NAZEV
690      XL = XL - LEN(N$(K)) / 2 * PC
700      GOTO 740
710      'BOD JE KONCEM NAZVU. POSUV ZPET O POSET BODU
720      'NA NAZEV.
730      XL = XL - LEN(N$(K)) * PC
740      'PREVOD SOURADNICE (XL,YL) V BODECH NA NEJBLIZSI
750      'ODPOVIDAJICI TISKOVOU POZICI
760      RO = INT(YL / PR + .5) + 1

```

```
770      CO = INT(XL / PC + .5) + 1
780      LOCATE RO,CO
790      PRINT N$(K);
800      B = A
810 NEXT K
820 GOTO 840
830 PRINT "SOURADNICE MIMO INTERVAL"
840 END
```

Prog. 5-11 Kruhový diagram.



Obr. 5-19 Kruhový diagram, výstup programu Prog. 5-11.

PROJEKTY PROGRAMOVÁNÍ

- 5-1. Upravte Prog. 5-1 pro vstup velikosti přírůstku (kroku) úhlu vytváření kružnice, místo počtu bodů. Tento program pak bude zobrazovat trojúhelník pro daný přírůstek úhlu 120 stupňů a výšeče pro hodnotu 90 stupňů.
- 5-2. Odvod'te program pro zobrazení kruhové výseče (oblouku), při poloměru a souřadnicích středu. Výseč může být určena jedním z hodnot koncových bodů, nebo počátečním a koncovým úhlem.
- 5-3. Odvod'te program pro zobrazení půlměsíce (nebo měsíce) pomocí dvou kruhových oblouků o různých poloměrech.
- 5-4. Napište program pro vytvoření vybarveného kruhu. Vymalujte vnitřek libovolnou zvolenou barvou nakreslením průměrových čar (prochází středem a mají délku R na obě strany od středu) této barvy. Podle rovnice /5-1/ jsou koncové body libovolné průměrové přímky 180 stupňů od sebe. Speciálního efektu lze dosáhnout změnou barvy při kreslení každé průměrové čáry.
- 5-5. Vnitřek kružnice může být vyplněn body v různých vzdálenostech od středu až do obvodu kruhu. Napište program pro vybarvení kružnice touto metodou pomocí rovnic /5-1/ a měnícího se poloměru od 0 do R .
- 5-6. Napište program pro vybarvení vnitřku kružnice pomocí horizontálních čar. Pro kružnici se středem (XC, YC) jsou souřadnice koncových bodů horizontální úsečky napříč kružnicí $XC - SQR(R^2 - (Y - YC)^2)$ a $XC + SQR(R^2 - (Y - YC)^2)$.

Souřadnice Y se mění od $YC - R$ do $YC + R$. Změny mezer mezi čarami a vynášení bodů místo plných čar vytvářejí různé vzory stínování. (Vybarvování vertikálními čarami lze provádět záměnou hodnot X a Y a záměnou XC a YC v uvedených vztazích).

- 5-7. Napište program pro zobrazení eliptické křivky vybarvené libovolnou z metod z Prog. 5-4, 5-5 nebo 5-6.
- 5-8. Upravte Prog. 5-4 pro vytváření grafu sinusové křivky s popsanými osami. Redukujte množství výpočtů využitím symetrie funkce sinus. Tento program by měl používat rovnici /5-5/ s libovolnými vstupními hodnotami H, W a D. Zobrazte tři cykly této křivky od $X = -D/W$ do $X = (6\pi - D)/W$.
- 5-9. Program vytvořený v Prog. 5-8 může být upraven pro zobrazení tlumené sinusové funkce násobením funkce sinus funkcí $\text{EXP}(-K \cdot X)$ pro každou hodnotu X. Zobrazte výslednou křivku pro $X = 0$ do $X = (10\pi - D)/W$ pro libovolnou kladnou hodnotu K.
- 5-10. Napište program pro zobrazení množiny dat jako malé kružnice, jejichž středy mají souřadnice, odpovídající daným hodnotám. Součástí zobrazení budou i souřadné osy s popisem.
- 5-11. Napište program pro interaktivní kreslení křivky, procházející množinou vnesených datových bodů. Tento program by měl umožňovat, aby byla křivka kreslena podle vstupu z klávesnice, nebo z jiných interaktivních zařízení.
- 5-12. Napište program pro zobrazení množiny datových bodů a parabolické křivky, která co nejtěsněji prochází touto množinou. Pro množinu N vstupních datových bodů $(X(1), Y(1)), (X(2), Y(2)), \dots, (X(N), Y(N))$, lze určit

koeficienty paraboly (C_1 , C_2 a C_3 viz rovnice 5-7) pomocí metody nejmenších čtverců. Následující soustava rovnic umožňuje určit hodnoty koeficientů C_1 , C_2 a C_3 paraboly:

$$\# Y(I) = C_1 * \# X(I)^2 + C_2 * \# X(I) + C_3 * N$$

$$\# X(I) * Y(I) = C_1 * \# X(I)^3 + C_2 * \# X(I)^2 + C_3 * \# X(I)$$

$$\# (X(I)^2) * Y(I) = C_1 * \# X(I)^4 + C_2 * \# X(I)^3 + C_3 * \# X(I)^2$$

kde symbol $\#$ (sigma) znamená součet přes všechny hodnoty I od 1 do N .

- 5-13. Navrhněte na grafickém papíru obrazec nebo scénu a napište pro jeho zobrazení program, užívající příslušné rovnice křivek pro approximaci kresby. Vyplňte obraz stínovými a barevnými vzory.