

EXMINSY

PRAHA

ATARI 800 XE/XL a vyšší,

i pro ROZŠÍŘENOU PAMĚŤ

BASIC K A N T O R 1

Vl. Kvasníčka

Program k výuce Basicu - vhodně pro začátečníky

V Praze, 1994

K A N T O R - 1

Výukové programy k BASICu

Veškerá autorská práva vyhrazena pro Vladimíra Kvasničku, Praha

Výhradní práva šíření programů, návodu i výukových listů má fa EXMINSY, Praha

Obsah

1. Úvodní část
2. Temata výukového programu
3. Obsah kazety (diskety)
4. Obsluha programu
5. Proměnné, pole a matice
 - 5.1. Proměnné
 - 5.2. Pole
 - 5.3. Matice
6. Výukové listy

Připravila firma EXMINSY jako manuál ke kazetě (disketě) "KANTOR 1"

Atari klub PRAHA organizuje levné večerní i dálkové kurzy TURBO-BASICu. Na podzim zahájí i kurzy programování PC (POWER BASIC, QUICK BASIC).

Informace poskytnete EXMINSY, Kubíkova 1180, 18200 PRAHA 8, tel. 02/859 2980

1. ÚVODNÍ ČÁST

Soubor Kantor 1 je první díl série výukových programů zaměřených na BASIC, s podtitulem "z BASICu do strojáku." Jazyk Basic je stále ještě základním jazykem k uživatelskému programování. Dokládá to skutečnost, že patří ke standardní výbavě všech verzí MS DOSu (ať už pod názvem GW Basic nebo Quick Basic), nebo i to, že v příručkách doprovázejících tiskárny (a nejen typy EPSON) je programování tiskáren interpretováno v jazyku Basic. Basic byl totiž vytvořen jako aparát blízký lidskému myšlení a jeho příkazy jsou - byť v angličtině - srozumitelné a snadno zapamatovatelné. Jeho mutace (na př. POWER BASIC pro PC nebo TURBO BASIC pro Atari i PC) byly účelně doplněny nejen směrem ke zpracování dat, ale i směrem ke strukturovanému programování.

Předládáný program a sada příkladů žáka připraví jak k přechodu "dolů" na assembler, tak "nahoru" k TURBO Basicu a dalším. Jako rozšíření poznávacích možností je k dispozici výuková literatura, jejíž seznam je v poslední kapitole. Je zaměřena na popis různých programových triků a zajímavých rutin v BASICU i v ASSEMBLERU z programátorské praxe autora. Programy jsou určeny těm, kteří s jazykem BASIC začínají a chtějí se s počítačem ATARI důkladněji seznámit. Ulehčí jim i přechod na TURBO BASIC ATARI.

2. Temata výukového programu

1. Základní informace o počítači - příkazy BYE, DOS.
Jak zacházet s klávesnicí.
2. Příkazy PRINT, LIST, RUN - číslování řádků.
Využití zvláštních znaků - ERROR, FRE - logický řádek.
Čárka a středník v programech.
3. Počítač jako kalkulačka, nebo-li matematické, trigonometrické funkce počítače - proměnné.
4. Příkaz POSITION, GOTO, GOSUB, STOP, END, CONT,
ON X GOTO, ON X GOSUB, CLOAD, CSAVE, INPUT.

5. Úvod do řetězových proměnných.
6. Příkazy DIM, ADR, LEN, ASC, CHR\$, STR\$, CLR,
pole DIM A(5), DIM A(5,5).
7. Logické operace - IF...THEN, NOT, AND, OR,
smýčky FOR...NEXT, STEP, POP.

3. OBSAH KAZETY (DISKETY)

Hlavní program KANTOR 1 a KANTOR.GR

- 1/ Mr.1/1 PRINT
- 2/ Mr.1/2 POSITION
- 3/ Mr.1/3 GOSUB
- 4/ Mr.1/4 INPUT
- 5/ Mr.1/5 THEN
- 6/ Mr.1/6 DIM
- 7/ Mr.1/7 LEN
- 8/ Mr.1/8 STRING
- 9/ Mr.1/9 ASC
- 10/ Mr.1/10 FOR/NEXT
- 11/ Mr.1/11 TISK
- 12/ Mr.1/12 CTV.
- 13/ Mr.1/13 STEP
- 14/ Mr.1/14 SČÍTÁNÍ
- 15/ Mr.1/15 MINCIŘ

4. OBSLUHA PROGRAMU

Po nahrání programu se na horní části obrazovky objeví stručný obsah jednotlivých částí programu. Na spodní

částí obrazovky je MENU. Pomocí šipek z klávesnice si můžete zvolit část, která vás bude zajímat, a po stisku RETURN začne výuka. Klávesou SPACE (mezerník) můžete zastavit běh programu, nestačí-li sledovat výuku. Chcete-li přerušit výuku, stisknete RESET, vrátíte se do MENU. Stiskem klávesy B v MENU se dostanete do BASICU, kde si budete moci vyzkoušet své znalosti, nebo nahrát do počítače, vyzkoušet, případně upravit ukázkové programy, které následují za výukovým programem. Obsahují jen ty příkazy, které jste se naučili v první části.

V programu nenajdete jen strohé výukové texty, ale i řadu výukových prográmků, a ukázek jejich funkčnosti pomocí trasování kurzorem. Obrazovka je rozdělena na tři části. Horní část obrazovky je určena pro BASIC. Tato část obrazovky se chová stejně jako v BASICU, a je určena pro ukázky programů. Ve středu obrazovky se nachází klávesnice, která spolupracuje s obrazovkou pro BASIC. Tato klávesnice imituje klávesnici počítače a slouží hlavně při výuce prvního tematu - "jak zacházet s klávesnicí." Spodní část obrazovky je určena pro výukové texty.

Příkaz DOS zavolá MENU (BASIC TURBO 2000), v kterém můžete používat tyto funkce:

BASIC TURBO 2000

Load Save Quit Verify

- Load - Nahrání např. ukázkových programů v T2000 z kazety do počítače.
- Save - Ukládání vlastních programu na kazetu.
- Quit - Návrat do BASICU.
- Verify - Ověření záznamu na kazetě.

Vlastníte-li počítač s ramdiskem, např. typ 130XE, můžete programy v BASICU ukládat do ramdisku. Ramdisk je rezidentní a ovládá se těmito povely :

Stisk kláves START+RESET vyčistí obsah ramdisku. Vyčištění ramdisku je nutno provést vždy před jeho prvním použitím.

Napsání příkazu DOS + stisk klávesy OPTION a RETURN vypíše obsah ramdisku, tedy názvy programů které jsou v

něm uloženy.

Do ramdisku ukládáte programy pomocí příkazů:

SAVE "R:název" zpět LOAD "R:název"

nebo

LIST "R:název" zpět ENTER "R:název" a pod.

Program vyhledá první soubor, který má začátek názvu shodný s celým požadovaným názvem. Pokud takový soubor neexistuje, je ohlášena chyba. Pozor na záměny příkazů SAVE-LOAD, LIST-ENTER. Též by se ohlásila chyba. Přejeme všem začínajícím programátorům mnoho úspěchu...

5. Proměnné, pole a matice.

5.1. Proměnné:

BASIC provádí řadu číselných operací. Výsledky těchto operací můžeme vytisknout na obrazovku:

PRINT 8 * 7

Nebo uložit do proměnných:

A = 8 * 7

To znamená, že nějakému zvolenému znaku (proměnné) přiřazujeme číselnou hodnotu. Proměnné mohou nabývat různých hodnot, záleží na tom, jakou hodnotu jí přiřadíme. Jejich hodnotu můžeme zvyšovat nebo snižovat:

A = A + 2 : B = B - 1

Můžeme je také používat v různých matematických operacích (vzorcích):

C = 7 * X - (B / 27)

Dále v příkazech, kde svým číselným obsahem nahrazují číselnou hodnotu:

GOTO Z:POSITION X,Y

V některých případech se bez proměnných neobejdete:

FOR A=1 TO 10

ON A GOTO ... a pod.

Proměnnými mohou být různé znaky abecedy nebo kombinace znaků a čísel a také krátké názvy, které nám připomínají, k jakému účelu proměnná slouží (např. HR-hráč, DL-délka a pod.).

POZOR!

Proměnná nemůže být nikdy číslo. Toto číslo bychom již nikdy nemohli v programu použít, počítač by ho vždy považoval za proměnnou. Vyvarujte se také příliš dlouhým názvům proměnných. Dlouhý název může zavinit chyby programu. Takové chyby se velice těžko hledají. Pro název proměnné použijte max. 4 znaky (to je doporučení ze zkušenosti).

Majde-li počítač v odeslané řádce do paměti (stisk RETURN) proměnnou, která ještě nebyla použita, zařadí jí do tabulky názvů proměnných a přiřadí jí pořadové číslo, v jakém byla do tabulky uložena. První proměnná má pořadové číslo 128, poslední nula. Celkem je tedy možno použít 128 proměnných. I přesto, že je to poměrně vysoké číslo, používejte proměnných co nejméně. Stejnou proměnnou lze přece použít k více účelům v různých částech programu. Proměnným takto použitým se říká lokální proměnné.

5.2. Pole.

Dalším typem proměnných jsou pole. Stejně jako proměnné mohou i pole obsahovat pouze číselnou hodnotu. Čím se tedy liší? Každá proměnná si pamatuje pouze jednu číselnou hodnotu. Pole může však mít řadu paměťových míst, takže do něj lze uložit více údajů. Prvky, z kterých je datové pole sestavené, jsou očíslované. Na jednotlivé prvky pole se můžeme zeptat udáním příslušného čísla místa. Pole je vynikajícím pomocníkem k vytváření číselných tabulek v paměti počítače. Pro pole platí stejné zásady označování pomocí znaků abecedy, stejně jako u proměnných. Příkazem DIM nastavíme, kolik bude mít pole paměťových míst.

Napr. DIM A(5)

Tento příkaz vytvoří v paměti počítače prostor pro 6 čísel $\langle A(0), A(1), A(2), A(3), A(4), A(5) \rangle$. Pro každé toto číslo bude vynecháno 6 bytes paměti.

Jednotlivým buňkám pole můžeme přiřazovat různé číselné hodnoty:

$A(2)=27:A(0)=255$

Přepínáním jednotlivých buňek pole, můžeme řídit i pomocí jiné proměnné:

Nulování buňek pole: 5 DIM A(5)
10 FOR X=0 TO 5
20 A(X)=0
30 N.X

Jak tato pole využít v programech? Důležitou vlastností pole je právě možnost jeho řízení pomocí jiné proměnné. Pro snadnější pochopení si využítí pole vysvětlíme na následujících ukázkách. První ukázka je řešena pouze pomocí proměnných, druhá s pomocí pole.

Řešení s proměnnými:

Budeme řešit část hry, v které hraje 6 hráčů. V každém kole získá každý hráč určitý počet bodů, které je potřeba připočítat k jeho celkovým bodům. Máme k dispozici tyto proměnné:

B - Počet bodů získaných po každém kole.

H - Hráč číslo.

B1 až B6 - Celkový počet hráčů.

```

500 ON H GOTO 501,502,503,504,505,506
501 B1=B1+8:GOTO 507
502 B2=B2+8:GOTO 507
503 B3=B3+8:GOTO 507
504 B4=B4+8:GOTO 507
505 B5=B5+8:GOTO 507
506 B6=B6+8:GOTO 507
507 .....

```

Řešení s polem:

Zadání je naprosto stejné. Místo proměnných B1 až B6, které slouží k celkovému součtu bodů každého hráče, je použito pole <BX(6)>.

```

5 DIM BX(6)

```

```

..

```

```

..

```

```

500 BX(H) = BX(H) + B

```

S poli se dají dělat i jiná kouzla. Jestliže máme např. dvě různá pole, jedno může řídit druhé. Pozor však na dimenzování pole, definované na začátku programu. Jestliže DIM A(5), potom A(6)=169 ohlásí ERROR.

Příklad spojení dvou polí:

Máme dvě pole: A(7), B(7) a proměnné "VY, X a Y". Potom můžeme napsat vzorec:

$$VY = A(B(X)) + A(B(Y))$$

5.3. Matice:

Matice je vlastně dvojrozměrné pole. To znamená, že i maticím se přiřazují pouze číselné hodnoty. I zde jsou jednotlivé prvky matice očíslované. Matice však očekává dvě čísla, dva rozměry. První číselný údaj udává vlastně počet sloupců, druhý údaj počet řádek. Matice je rovněž nutno dimenzovat na začátku programu příkazem DIM, stejně jako pole a platí pro ní i stejné zásady označování, které se rovněž provádějí pomocí velkých znaků abecedy nebo krátkými názvy.

Příklad: DIM A(5,3),HR(12,36)

Příkaz DIM A(5,3) vyhradí v paměťovém prostoru 6 x 4 míst po 6 bytes. Zobraze si tuto matici názorně:

	0	1	2	3	4	5
0	A(0,0)	A(1,0)	A(2,0)	A(3,0)	A(4,0)	A(5,0)
1	A(0,1)	A(1,1)	A(2,1)	A(3,1)	A(4,1)	A(5,1)
2	A(0,2)	A(1,2)	A(2,2)	A(3,2)	A(4,2)	A(5,2)
3	A(0,3)	A(1,3)	A(2,3)	A(3,3)	A(4,3)	A(5,3)

Jak matice využít? Matice je vlastně vynikající číselnou tabulkou. Tato tabulka může obsahovat různé informace. Představme si, že matice dimenzovaná na S(7,7) si pamatuje rozestavení hracích kamenů na šachovnici. Pěšák má přiřazenu číselnou hodnotu 1, věž 2, kůň 3, prázdné pole má nulovou hodnotu atd. Myní záleží na programátorovi, jak dokáže tuto situaci logicky vyřešit a měnit číselnou sestavu pole tak, aby tato změna odpovídala pravidlům šachů a jak tyto číselné údaje zobrazit v grafické formě na obrazovce. S jednotlivými maticemi lze provádět různé matematické operace, které jsou podobné jako u polí. Můžeme tedy jednotlivým prvkům matice přiřazovat číselnou hodnotu <A(2,2)=12>, prvky matice nahrazovat proměnnými <A(X,Y)=1>, s jednotlivými prvky matice provádět matematické operace <A(2,2)=A(7,7)*5:A(2,2)=A(2,2)+8(4,7)>, a pod.

Je nutno podotknout, že Atari BASIC sice umožňuje práci s maticemi, ale je velmi ochuzen o maticové operace.

Některé BASICy mají instrukce, umožňující provést matematické operace se všemi prvky matic.

Příklad: Matice A je součtem matic B a C.

```
100 DIM A(10,12),B(10,12),C(10,12)
```

```
200 MAT A = B + C
```

Podobně i součin: Matice Q je součinem matic X a Y.

```
100 DIM Q(10,30),X(8,30),Y(10,20)
```

```
200 MAT Q = M * N
```

To znamená, že se vynásobí všechny shodné prvky matic X a Y. Shodný rozměr matic X a Y je (8,20).

Dalšími pomocníky mohou být i dosazovací operátory, které přiřazují matici konstanty.

Např.: MAT A=ZERO Vynulování všech prvků matice A.

MAT A=CON Přiřazení hodnoty 1 všem prvkům matice A.

A pod.

Zajímavým příkazem je i MAT READ, který umožňuje přiřadit prvkům matice (pole) hodnoty uvedené v seznamu konstant příkazu DATA. Podobnou zajímavou instrukcí je i MAT INPUT a pod..

Co naplat! Máš počítač ATARI nic takového neumí a všechny podobné situace se musí řešit programově. Podívejme se, jak se řeší součet všech prvků dvou matic.

```
10 DIM A(2,3),B(2,3)
```

```
20 FOR X=0 TO 2:FOR Y=0 TO 3
```

```
30 A(X,Y)=0:B(X,Y)=0
```

```

40 NEXT Y:NEXT X
50 ? "Zadej prvky matice A a B"
60 FOR X=0 TO 2:FOR Y=0 TO 3
70   ? "A(";X;";";Y;")=";:INPUT M
80   ? "B(";X;";";Y;")=";:INPUT N
90   A(X,Y)=M:B(X,Y)=N
100 NEXT Y:NEXT X
110 ? "Soucet dvou matic:"
120 FOR X=0 TO 2:FOR Y=0 TO 3
130   ? "A(";X;";";Y;")+B(";X;";";Y;")=";A(X,Y)+B(X,Y)
140 NEXT Y:NEXT X

```

Řádek 10 - Dimenzování matic.

Řádek 20 až 40 - Nulování všech prvků matic.

Řádek 50 až 100 - Naplnění matic číselnými hodnotami pomocí příkazu INPUT.

Řádek 110 až 140 - Sečtení všech prvků dvou matic a výtisk součtu na obrazovku.

6. Výukové listy

Listy vytvářejí zázemí pro ty, kteří se chtějí o tajích programování dozvědět něco navíc. Od čísla 4 jsou listy doplňovány také informacemi o tom, jak diskutovaný problém zvládnout v ASSEMBLERU.

V současné době jsou na skladě tyto listy:

1. Zápis textu do GR. 8, 11 stran
2. "Player missile" grafika, 11 stran
4. Ramdisk počítačů, 10 stran

- 8. Zvukové efekty, 10 stran
- 9. Čtení z obrazovky, 10 stran
- 11. Pořizujeme si tiskárnu, 10 stran
- 12. O grafice, Movie Maker Animátor, 18 stran
- 13. Myš pro Atari, pokr. Movie Maker Animátor, 18 stran.

Autor i firma EXMINSY Vám přeji

úspěch při programování

V Praze, leden 1994

EXMINSY

Kubíkova 1180, 182 00 Praha 8

tel. 02 / 859 2980

nové i zánovní
POČÍTAČE pro MLÁDEŽ (ATARI I PC)

programy na zakázku
(podle zadání připravíme analýzu a realizaci)

INDIVIDUELNÍ ŠKOLENÍ U ZÁKAZNÍKA s PC
(pro začínající i pokročilé)

ekonomické i lékařské systémy
faxy a kopírky pro náročné, záznamníky i telefony

NOVINKA! Instalace sítě:

1x PC + více ATARI 800/130 .