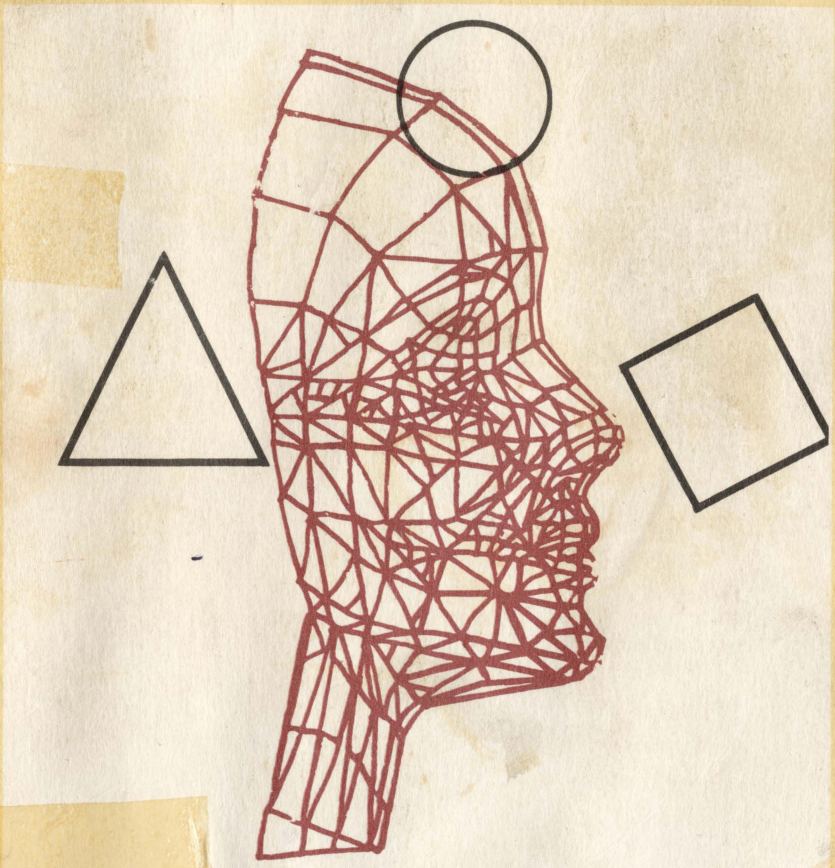
František Turnovec

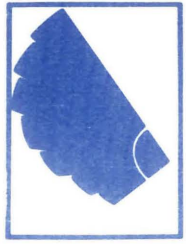
SKÚSTE TO S BASICOM



SMENA

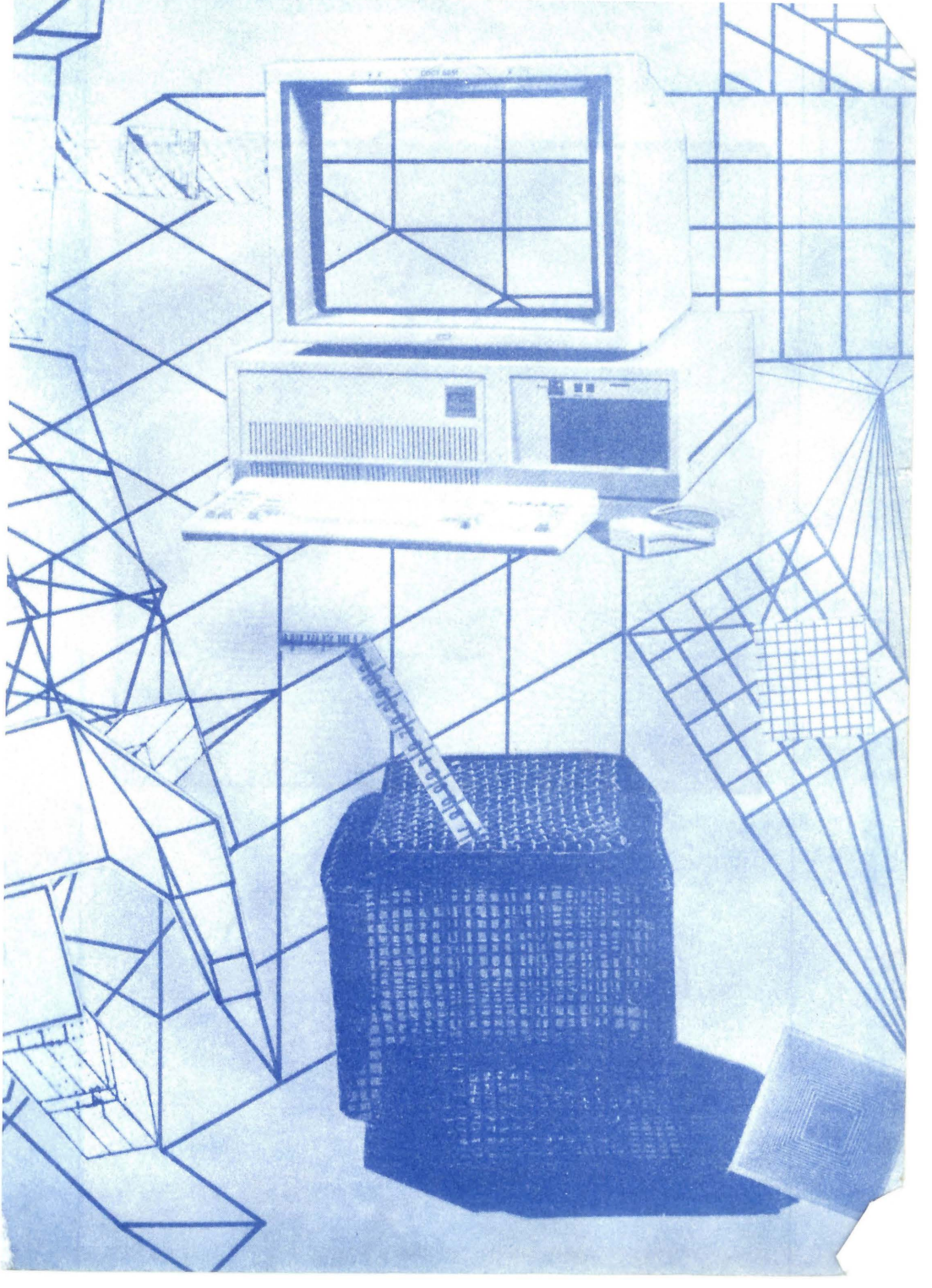
SMENA

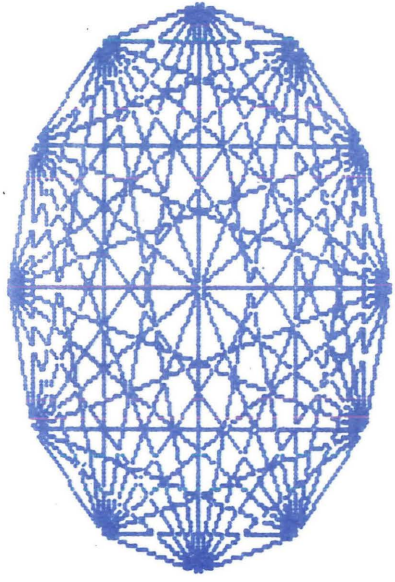
Vydavatelstvo Smena

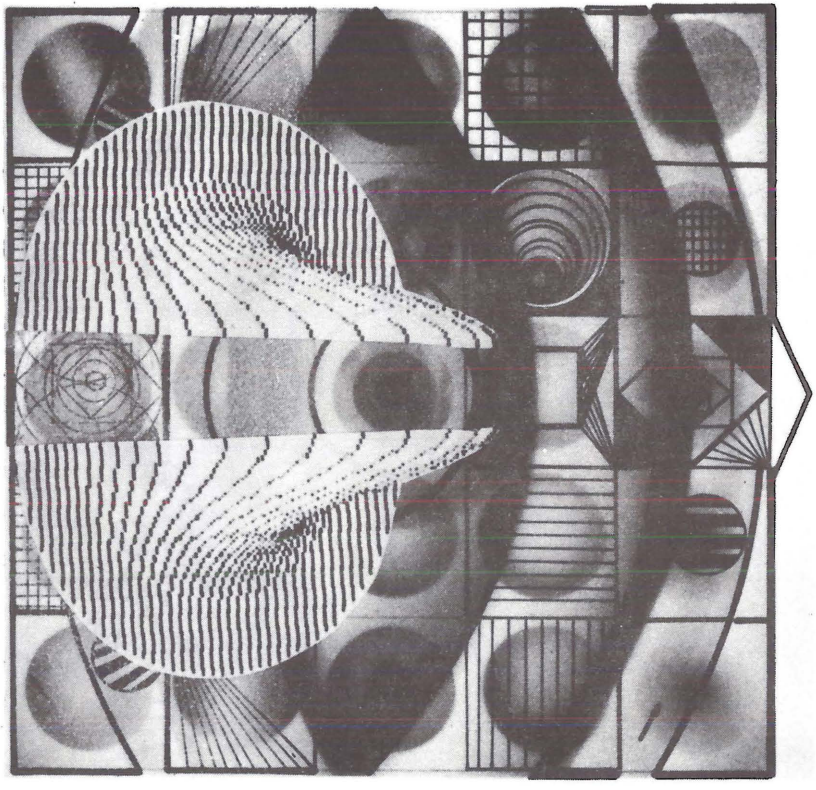


Edícia Sputnik

Compu
e relativ
ata-proc







František Turnovec

SKÚSTE TO S BASICOM

Úvod do programovania
a 50 programov pre PMD 85,
Didaktik Alfa,
Atari XL a Atari XE

SMENA
BRATISLAVA
1989

Vydavateľstvo Smena

Edícia Spútnik

František Turnovec

©
RNDr. Ing. František Turnovec
1989

Rukopis lektorovali

doc. Ing. Pavol Návrat, CSc.,
Ing. Ján Sepp, CSc.

Illustrations
©
Peter Rónai
1989

ISBN 80-221-0006-4

OBSAH

Úvod ... / 11

1. Programujeme v basicu ... / 17
1.1 Práca s osobnými počítačmi PMD 85 a Didaktik Alfa ... / 18
1.1.1 Klávesnica ... / 18
1.1.2 Obrazovka ... / 21
1.1.3 Začiatok práce – privolanie interpreta basicu ... / 22
1.2 Povely interpreta ... / 23
1.2.1 Vkladanie programov ... / 23
1.2.2 Prezeranie programov ... / 25
1.2.3 Editovanie programu ... / 25
1.2.4 Spustenie a pozastavenie vykonávania programu ... / 28
1.2.5 Povely na prácu s magnetofónom ... / 28
1.2.6 Využitie funkčných klávesov ... / 31
1.3 Základné prvky jazyka basic ... / 32
1.3.1 Priamy a programový režim ... / 33
1.3.2 Formát programových riadkov ... / 33
1.3.3 Konštanty, premenné a polia ... / 34
1.3.4 Štandardné funkcie ... / 38
1.3.5 Výrazy a relácie ... / 42
1.3.6 Prehľad príkazov jazyka basic G ... / 45
1.4 Programovanie výpočtov a práce s textami ... / 48
1.4.1 Začiatok a koniec programu, komentáre ... / 48
1.4.2 Vstupy a výstupy ... / 49
1.4.3 Priraďovací príkaz LET ... / 55
1.4.4 Príkazy skokov ... / 58
1.4.5 Programovanie cyklov ... / 63
1.4.6 Deklarácia polí a práca s poľami ... / 68
1.4.7 Deklarácia a čítanie zoznamu konštánt ... / 73
1.4.8 Príkazy funkcií a podprogramov ... / 79
1.4.9 Príkazy na prácu so súbormi na kazetách ... / 88
1.4.10 Niekoľko užitočných poznatkov ... / 92
1.5 Grafika ... / 99
1.5.1 Formáty obrazovky ... / 99
1.5.2 Jednoduché kreslenie príkazom PRINT ... / 101
1.5.3 Príkazy na kreslenie v bodovom režime ... / 103
1.5.4 Príkazy na kreslenie v bajtovom režime ... / 117
1.5.5 Niekoľko užitočných poznatkov ... / 123
1.6 Programovanie výstupov na tlačiareň ... / 129
1.6.1 Príkazy výstupov na tlačiareň ... / 129
1.6.2 Riadenie formátu tlače ... / 133

Vydavateľstvo Smena

1.7 Práca s osobnými počítačmi Atari 800/XL a 130/XE ... / 138
1.7.1 Klávesnica a obrazovka ... / 140
1.7.2 Povely interpreta jazyka Atari basic ... / 143
1.7.3 Spoločný základ ... / 146
1.7.4 Práca s reťazcami ... / 147
1.7.5 Práca so súbormi ... / 151
1.7.6 Farebná grafika ... / 156
1.7.7 Spracovanie zvukov ... / 163
1.7.8 Štartovanie programov v strojovom kóde ... / 168
2. Tridsať programov nielen pre zábavu ... / 173
2.1 Kreslíme na obrazovke ... / 175
2.1.1 Kružnica a elipsa ... / 175
2.1.2 Pravidelné n-uholníky ... / 178
2.1.3 Špirály a hviezdy ... / 179
2.1.4 Kocka a valec ... / 182
2.1.5 Kruhový pohyb ... / 184
2.1.6 Grafy funkcií a diagramy ... / 185
2.1.7 Portréty a karikatúry ... / 190
2.2 Jedenásť hier s počítačom ... / 198
2.2.1 Hra POKLAD ... / 198
2.2.2 Hra HANOJSKÉ VEŽE ... / 201
2.2.3 Hra ZÁPALKY ... / 205
2.2.4 Hra ŽIVOT ... / 208
2.2.5 Hra ČÍSLA ... / 213
2.2.6 Hra SLOVÁ ... / 216
2.2.7 Hra TROJUHOLNÍKY ... / 219
2.2.8 Hra '15 ... / 223
2.2.9 Hra ŤAH JAZDCOM ... / 227
2.2.10 Hra DÁMA ... / 232
2.2.11 Hra HÚSENICA ... / 235
2.3 Počítač – pomocník v práci ... / 239
2.3.1 Kvadratické rovnice ... / 239
2.3.2 Malý počtár ... / 240
2.3.3 Mikrotest ... / 244
2.3.4 Poradca ... / 248
2.3.5 Adresár ... / 253

Príloha 1 – Chybové hlásenia v jazyku basic G ... / 260
Príloha 2 – Chybové hlásenia v jazyku Atari basic ... / 263
Príloha 3 – Tabuľka ASCII kódu ... / 266
Literatúra ... / 269

ÚVOD

Jazyk basic (z anglického Beginners All-purpose Symbolic Instruction Code) vznikol v roku 1964 ako jednoduchý nástroj na prácu s počítačom pre začiatočníkov. Jeho autori pravdepodobne nepredpokladali, akú popularitu si získa. Za veľké rozšírenie basic vďačí (okrem svojej jednoduchosti) predovšetkým rýchlemu rozvoju osobných počítačov v poslednom desaťročí. V roku 1977 sa začali vyrábať prvé 8-bitové osobné počítače (Apple, Commodore). Za nimi nasledovalo veľa ďalších, viac alebo menej úspešných typov osobných počítačov rôznych firiem. Všetky 8-bitové osobné počítače používajú basic ako základný (a v mnohých prípadoch aj jediný) vyšší programovací jazyk.

V druhej polovici osemdesiatych rokov sa 8-bitové osobné počítače stali súčasťou vybavenia škôl, domov pionierov a krúžkov mladých programátorov v ČSSR. Boli to predovšetkým PMD 85 a 'školské' počítače Didaktik Alfa, v ČSR aj počítače IQ 150 a IQ 151. V domácnostiach sa zjavili zahraničné osobné počítače, najprv obľúbený Sinclair ZX Spectrum, v posledných rokoch počítače firmy Atari (Atari XL a XE).

V podmienkach „pretekov" rôznych firiem vo výrobe osobných počítačov sa jazyk basic nevyvíjal práve najšťastnejšie. Vzniklo veľa jeho dialektov pre rôzne typy osobných počítačov, pričom niektoré z týchto dialektov sú si veľmi vzdialené. Každý dialekt obsahuje spoločný základ – niekoľko štandardných príkazov, ktoré na všetkých počítačoch fungujú rovnako. Iné príkazy, hoci sú si podobné, majú v rôznych dialektoch odlišné použitie. Každý dialekt basicu má aj svoje vlastné, iba jemu „zrozumiteľné" príkazy. Preto nemá veľký význam učiť sa basic všeobecne, bez ohľadu na typ počítača, s ktorým chceme pracovať.

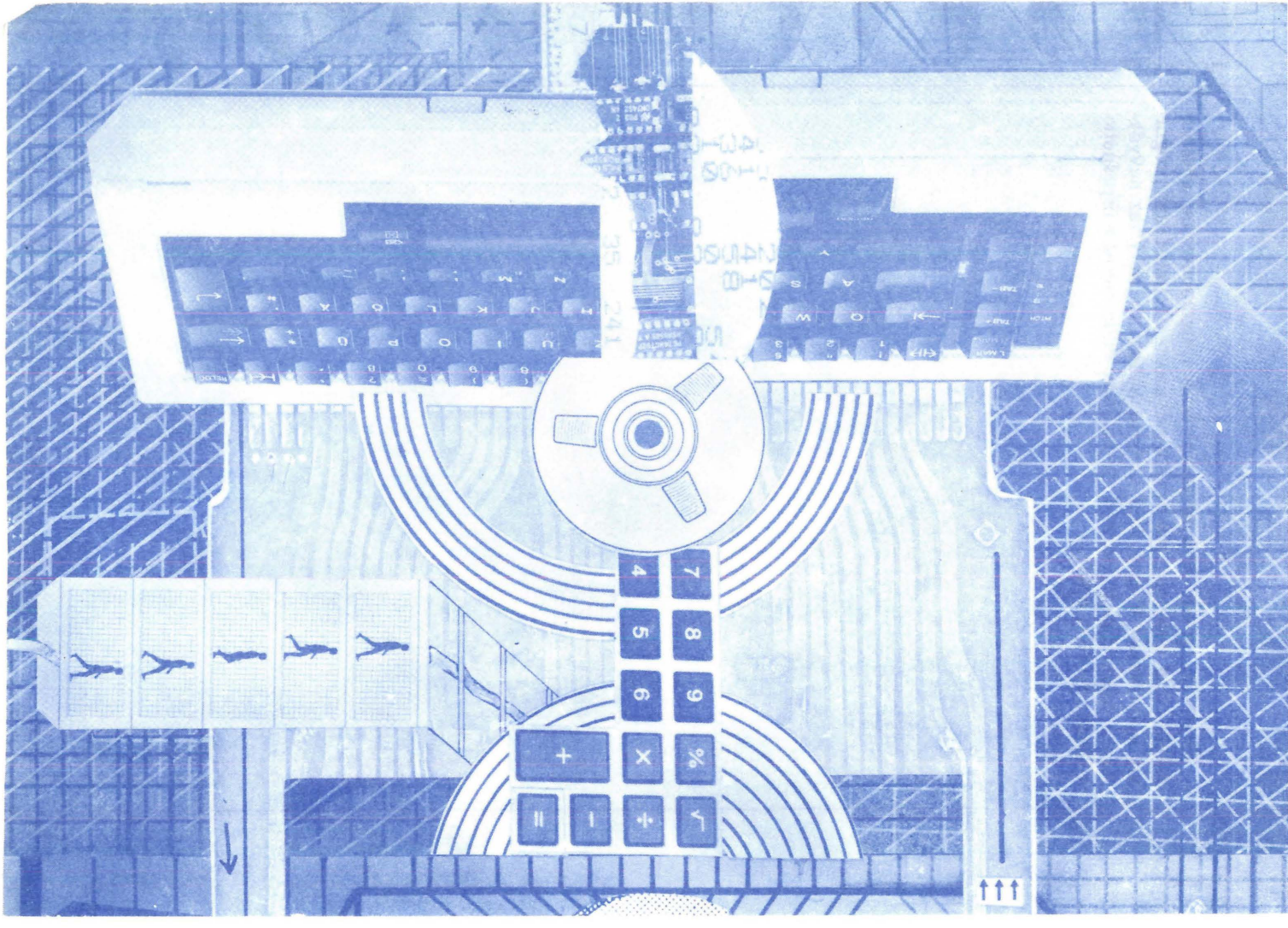
Táto kniha vás bližšie oboznámi s dialektmi basicu pre osobné počítače PMD 85 a Didaktik Alfa (tzv. basic G) a s odlišnosťami dialektu používaného na počítačoch Atari XL a XE (Atari basic). Naučíte sa zostavovať programy na jednoduché výpočty, spracovanie textov a „kreslenie" (grafiku). Výklad sa opiera o veľký počet názorných príkladov. Okrem nich v knihe uvádzame 50 ucelených programov z rôznych oblastí použitia počítačov – od jednoduchých výpočtových programov po programy počítačových hier. Znalosť jedného dialektu basicu uľahčuje rýchle zvládnutie jeho ďalších dialektov. Preto hoci je táto kniha určená predovšetkým používateľom počíta-

čov PMD 85, Didaktik Alfa a Atari XL, resp. XE, môžu v nej nájsť užitočné informácie a inšpiráciu na tvorbu programov aj záujemcovia o prácu s inými typmi osobných počítačov.

8-bitové osobné počítače sa u nás už stali užitočným pomocníkom v menej rozsiahlych výpočtových a administratívnych prácach, pri výučbe a v neposlednom rade aj vďačným partnerom detí (a často i dospelých) pri najrôznejších hrách. Vývoj sa však nezastavil a viedol k novým, dokonalejším typom – 16-bitovým a 32-bitovým osobným počítačom typu IBM PC, ktoré sú už plnohodnotným nástrojom profesionálnej činnosti. (Niekedy sa termín osobné počítače chápe v zúženom zmysle a označujú sa ním len tieto počítače, kým ostatné, najmä 8-bitové, sa radia medzi tzv. domáce počítače.) Na písacom stole, bez veľkých nárokov na obsluhu a špeciálne zabezpečenie, poskytujú technikovi, ekonómovi, projektantovi, lekárovi, vedcovi, riadiacemu pracovníkovi, a dokonca aj výtvarníkovi a hudobníkovi pracovné možnosti, ktoré ešte pred niekoľkými rokmi boli vstave zvládnuť len veľké sálové počítače s početnou obsluhou. Mnohí z vás sa s nimi stretnú vo svojom ďalšom živote a práci. Jazyk basic pre tieto počítače poskytuje oveľa viac možností ako dialekty basicu pre 8-bitové osobné počítače, možno na nich používať aj ďalšie, dokonalejšie jazyky, predovšetkým pascal, a bohaté aplikačné programové vybavenie. Skúsenosti a poznatky, ktoré získate v práci s 8-bitovými osobnými počítačmi, sa však nestratia. Pomôžu vám pri prechode k dokonalejšej a výkonnejšej výpočtovej technike.

Autor pokladá za svoju milú povinnosť poďakovať šéfredaktorovi časopisu Elektrón RNDr. Ľ. Györffymu za podnet k napísaniu tejto knihy. Jej časti, predovšetkým niektoré z programov, boli publikované v prvých dvoch ročníkoch Zenitu pionierov v rokoch 1986–1988 a čitateľský ohlas seriálov Program pre počítač, Programujeme v basicu a Kreslíme na obrazovke povzbudil autora k ich prepracovaniu a ucelenému usporiadaniu. Kniha je určená pre rovnaký okruh čitateľov, ktorí sledovali Zenit pionierov – žiakov základných škôl a ich učiteľov, ktorí majú záujem o prácu s osobnými počítačmi, ich využívanie v škole, v bežnom živote a pri odpočinku. Autor chce tiež vyjadriť svoju vďaku recenzentom knihy doc. Ing. P. Návratovi, CSc., z Katedry počítačov EF SVŠT v Bratislave a Ing. J. Seppovi, CSc., z VÚVT v Žiline za cenné pripomienky a návrhy na zdokonalenie rukopisu, ako aj Ing. S. Vejmolovi, CSc., z VŠE v Prahe za poskytnuté materiály a priateľské rady pri spracovaní knihy.

Čitateľom, predovšetkým začínajúcim programátorom, chceme zaželať veľa úspechov pri prvých pokusoch o samostatné programy a vytrvalosť v čase, keď sa tieto pokusy na prvý raz nevydaria – odmenou za trpezlivosť im nakoniec bude radosť z úspešného výsledku tvorivej práce.



1
2
3
2
3

1. PROGRAMUJEME V BASICU

V tejto časti uvádzame prehľad poznatkov o práci s 8-bitovými osobnými počítačmi v jazyku basic. Vychádzame pritom z dialektov jazyka basic pre počítače PMD 85 a Didaktik Alfa, ktoré sú prakticky totožné. Pri výklade jednotlivých príkazov a funkcií upozorňujeme, či a ako sú použiteľné na počítačoch Atari XL a XE. Poznatky o basicu pre počítače firmy Atari potom dopĺňame (najmä v súvislosti s grafikou a spracovaním textov, kde sa jednotlivé dialekty basicu najviac odlišujú) v kapitole 1.7. Výklad vychádza z veľkého počtu príkladov a navyše obsahuje 20 ucelených programov pre počítače PMD 85 (ktoré sú bezprostredne použiteľné aj na Didaktiku Alfa) a pre počítače Atari XL a XE.

Vydavateľstvo Smena

František Turnovec

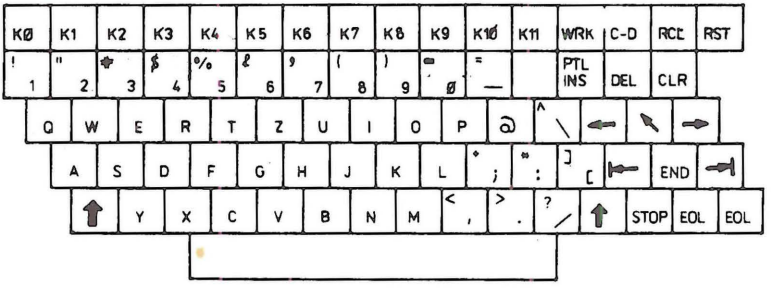
1.1 PRÁCA S OSOBNÝMI POČÍTAČMI PMD 85 A DIDAKTIK ALFA

Základnú zostavu počítačov PMD 85 a Didaktik Alfa tvoria: centrálna jednotka spolu s klávesnicou, obrazovka (k počítaču PMD 85 možno pripojiť bežný televízny prijímač, súčasťou vybavenia počítača Didaktik Alfa je špeciálne upravený televízny prijímač, tzv. *monitor*), zdroj (Didaktik Alfa má zdroj zabudovaný v monitore) a kazetový magnetofón. Túto zostavu možno rozšíriť pripojením tlačiarne.

S počítačom sa „zhovárame" prostredníctvom klávesnice a obrazovky. Prostredníctvom klávesnice vkladáme povely, príkazy a údaje, na obrazovke sa zjavujú „odpovede" počítača. Prácu počítača riadime pomocou špeciálneho programu, napevno zabudovaného v jeho pamäti, ktorý sa nazýva *monitor*. Nemýľte si ho s televíznym monitorom pre Didaktik Alfa! Monitor nám umožňuje privolať ďalší pevne zabudovaný program, nazývaný *interpret jazyka basic*. Pomocou povelov interpreta vytvárame, ladíme, opravujeme a štartujeme programy v basicu, hotové programy zapisujeme na kazetu magnetofónu a čítame ich z nej. S využitím riadiaceho programu (monitora) možno pracovať aj s programami v tzv. *strojovom jazyku* (v ňom je zostavený samotný monitor a interpret basicu), ako začiatočníci sa však v rámci tejto knihy uspokojíme s možnosťami, ktoré nám poskytuje interpret basicu.

1.1.1 KLÁVESNICA

Povely, príkazy a údaje vkladáme cez klávesnicu. Klávesnica počítača PMD 85 je na obr. 1.1.



1.1 Klávesnica počítača PMD 85

Klávesnica obsahuje tri typy klávesov:

a) Alfanumerické klávesy (veľké písmená latinskej abecedy, číslice od 0 do 9 a špeciálne znaky).

b) Systémové a editovacie klávesy.

c) Funkčné klávesy.

Pomocou *alfanumerických klávesov* vkladáme povely interpreta, programové riadky basicu a údaje požadované programom. Stlačením každého z týchto klávesov vložíme jeden znak. Dlhým neoznačeným klávesom v dolnej časti klávesnice vkladáme medzery (na tento kláves sa v ďalšom výklade odvolávame názvom SPACE). Niektoré z alfanumerických klávesov obsahujú dva znaky – horný a dolný. Jednoduchým stlačením takéhoto klávesu vložíme dolný znak, súčasným stlačením klávesu s horným a dolným znakom a klávesu ⇧ (na tento kláves sa odvolávame názvom SHIFT) vložíme horný znak. Vkladané znaky sa zobrazujú v poslednom riadku obrazovky, tzv. *dialógovom riadku*, v pozícii, v ktorej je zobrazený *kurzor* (znak podčiarknutia). Vloženie každého riadka ukončíme stlačením klávesu EOL, ktorým odošleme príslušný povel, príkaz alebo údaj na spracovanie.

Systémové a editovacie klávesy slúžia na riadenie niektorých činností a opravovanie (editovanie) vkladaných riadkov. Ďalej uvádzame účinok stlačenia jednotlivých klávesov:

EOL	ukončenie vkladania riadka a jeho odoslanie na spracovanie,
↑	SHIFT, prepnutie klávesnice na vkladanie horných znakov,
STOP	prerušenie vykonávania programu alebo výpisu programu na obrazovku,
RST	stlačenie tohto klávesu spolu s klávesom SHIFT spôsobí znovunaštartovanie monitora (napr. v prípade, keď počítač prestane reagovať),
RCL	privolanie obsahu posledného riadka, ktorý bol odoslaný stlačením klávesu EOL, do dialógového riadka,
C–D	zobrazenie poslednej správy počítača používateľovi v dialógovom riadku,
CLR	vymazanie obsahu dialógového riadka; ak stlačíme spolu s klávesom SHIFT, vymaže sa iba

časť dialógového riadka napravo od kurzora (vrátane znaku, pod ktorým je nastavený kurzor),

← (šípka vľavo), posunutie kurzora v dialógovom riadku o jednu pozíciu vľavo,

→ (šípka vpravo), posunutie kurzora v dialógovom riadku o jednu pozíciu vpravo,

END posunutie kurzora na posledný vložený znak dialógového riadka,

↖ (šípka šikmo hore), premiestnenie kurzora na začiatok dialógového riadka,

→| posúvanie obsahu dialógového riadka vpravo (na obrazovke sa zobrazuje iba 48 znakov v riadku, riadok však môže obsahovať až 80 znakov; stlačením tohto klávesu zabezpečíme, že obsah riadka na obrazovke sa posunie o jedno miesto vpravo, pričom kurzor zostáva na mieste),

|← posúvanie obsahu dialógového riadka vľavo,

DEL vymazanie znaku v dialógovom riadku, nachádzajúceho sa nad kurzorom,

PTL INS vsunutie medzery v dialógovom riadku v pozícii kurzora,
spolu s klávesom SHIFT zabezpečuje nastavenie režimu priebežného zobrazovania vkladaných riadkov na obrazovke „rolovaním" zdola nahor; opakované stlačenie tohto klávesu a klávesu SHIFT zruší režim zobrazovania vkladaných riadkov (po odoslaní vloženého riadka sa tento vymaže z obrazovky).

Funkčné klávesy KØ, K1, ..., K11 a kláves WRK používame na urýchlenie vkladania príkazov a na riadenie postupu práce pri vykonávaní programov. S ich využitím sa oboznámime v ďalších kapitolách.

Poznámka 1.1
Klávesnica počítača Didaktik Alfa používa iné označenia niektorých systémových, editovacích a funkčných klávesov (ich funkcie sú pritom rovnaké ako na PMD 85). Ďalej uvádzame porovnanie označení týchto klávesov na PMD 85 a Didaktiku Alfa:

PMD 85	Didaktik Alfa
RST	RESET
EOL	END OF LINE
C–D	MSG
↖	HOME
END	END LINE
→|	$\overrightarrow{\text{ROL}}$
|←	$\overleftarrow{\text{ROL}}$
PTL	PAL
INS	INS
WRK	◯
KØ, . . . , K11	Ø, . . . , 11

Okrem toho obsahuje klávesnica Didaktiku Alfa navyše 4 klávesy:

inverzia obrazu,

obraz bez blikania pri použití špeciálnej modifikácie systému,

blikanie zadaných častí zobrazenia pri použití špeciálnej modifikácie systému,

zrušenie inverzie obrazu.

Používanie týchto klávesov vysvetľujeme v kapitole 1.5.

1.1.2 OBRAZOVKA

Vkladané povely, príkazy a údaje, ako aj správy počítača používateľovi sa zobrazujú na *obrazovke*. V alfanumerickom režime (zobrazovanie textov) je obrazovka rozdelená do 26 riadkov, pričom prvých 25 riadkov slúži na zobrazovanie údajov, resp. programov a posledný, tzv. *dialógový riadok* je určený na zobrazovanie vkladaných povelov, príkazov a údajov, resp. na zobrazovanie správ počítača používateľovi. V každom riadku možno zobraziť 48 znakov.

Na začiatku dialógového riadka je umiestnený tzv. *kurzor* (zobrazený ako znak podčiarknutia). Kurzor označuje pozíciu v riadku, do ktorej sa zapíše znak vkladaný z klávesnice. Po vložení znaku sa kurzor posunie o jedno miesto vpravo. Pohyb kurzora môže riadiť aj používateľ pomocou špeciálnych klávesov.

V *grafickom režime* (pozri kapitolu 1.5) je obrazovka rozdelená na 256 stĺpcov a 243 riadkov, teda na 256 × 243 bodov, z ktorých každý

môže svietiť alebo nesvietiť, čím sa dosahuje možnosť kreslenia zložitejších obrázkov.

Základným spôsobom zobrazovania je „biele na čiernom", teda zobrazujú sa biele znaky na čiernom podklade. Tento spôsob možno zmeniť na „čierne na bielom" (tzv. inverzia obrazovky).

1.1.3 ZAČIATOK PRÁCE – PRIVOLANIE INTERPRETA BASICU

Po zapnutí televízora a zdroja (PMD 85), resp. monitora (Didaktik Alfa) sa v dialógovom riadku zjaví správa

*** PMD 85 READY /1.Ø ***

resp.

* DIDAKTIK ALFA READY *

Počítač je v tomto momente riadený základným obslužným programom – monitorom. Aby sme s počítačom mohli pracovať v jazyku basic, musíme privolať špeciálny program – interprét jazyka basic.

Interpret basicu privoláme vložením povelu

BASIC G

(na PMD 85), resp.

BASIC ALFA

(na Didaktiku Alfa) a stlačením klávesu EOL (END OF LINE). Na obrazovke sa zjaví správa

BASIC-G/V 1.Ø
OK

resp.

BASIC ALFA READY
OK

a počítač je pripravený na prácu v basicu (očakáva vloženie niektorého z povelov interpreta alebo programu).

Poznámka 1.2

Počítač PMD 85 je k dispozícii v dvoch modifikáciách: PMD 85–1 a PMD 85–2. Náš výklad vychádza z PMD 85–1 a na niektoré drobné odlišnosti PMD 85–2 upozorňujeme v poznámkach.

Na rozdiel od PMD 85–1 sa hneď po zapnutí počítača PMD 85–2 na obrazovke priamo ohlási interpret basicu správou

BASIC-G V2.Ø

v prvom hornom riadku obrazovky. V poslednom (dialógovom) riadku sa vypíše správa

OK

a používateľ môže začať vkladať povely interpreta alebo program v basicu.

Režim zobrazovania vkladaných riadkov na obrazovke je na PMD 85–2 upravený tak, že po každom odoslaní vloženého riadka sa tento automaticky zobrazí: prvých 25 riadkov sa zobrazuje zhora dole a ďalšie riadky potom „rolovaním" zdola nahor. Tento režim zobrazovania možno zrušiť súčasným stlačením klávesov SHIFT a PTL – vkladané riadky sa v tomto prípade po odoslaní nezobrazia. To isté platí aj pre výstupy z programov.

V porovnaní s PMD 85–1 má PMD 85–2 aj niektoré zvláštnosti v práci s klávesnicou. Okrem veľkých písmen na ňom môžeme pracovať aj s malými písmenami. Malé písmená sa vkladajú súčasným stlačením klávesu SHIFT a príslušného klávesu s veľkým písmenom (napr. ak stlačíme naraz SHIFT a kláves A, zobrazí sa malé písmeno a).

Súčasným stlačením klávesov SHIFT a RCL prejdeme z režimu práce s interpretom basicu do režimu monitora. V dialógovom riadku sa vypíše správa

** Os ready **

a počítač očakáva vkladanie povelov monitora. Do režimu interpreta basicu sa vrátime súčasným stlačením klávesov SHIFT a DEL.

Súčasným stlačením klávesov SHIFT a ↖ sa vymaže obrazovka (okrem neodosланého obsahu dialógového riadka). Súčasným stlačením SHIFT a ← sa zmení podklad obrazovky na inverzný (ak sa dovtedy písalo „biele na čiernom", bude sa po stlačení písať „čierne na bielom" a naopak). Opakovaným stlačením týchto dvoch klávesov sa vrátime k pôvodnému režimu zobrazovania. Základným režimom zobrazovania, nastaveným automaticky po zapnutí počítača, je „biele na čiernom".

Súčasným stlačením klávesov SHIFT a →| sa cyklicky nastavuje spôsob vypisovania na obrazovku: pri prvom stlačení polojasom, pri druhom stlačení blikanie plným jasom, pri treťom stlačení blikanie polojasom, pri štvrtom stlačení návrat k normálnemu spôsobu zobrazovania plným jasom atď.

1.2 POVELY INTERPRETA

Pri práci s interpretom basicu môžeme vkladať nové programy, opravovať ich, zapisovať programy z vnútornej pamäte počítača na kazetu magnetofónu, načítavať programy z kazety magnetofónu do vnútornej pamäte počítača, štartovať programy a pod. Na tieto účely používame povely interpreta.

1.2.1 VKLADANIE PROGRAMOV

Program v jazyku basic vkladáme po riadkoch. Každý riadok sa začína poradovým číslom, za ktorým nasleduje medzera a vlastný text programového riadka. Čísla programových riadkov volíme vzostupne. Vloženie programového riadka, ktorý sa zobrazuje v dialógovom riadku obrazovky, ukončujeme stlačením klávesu EOL.

Skôr ako začneme vkladať nový program, zabezpečíme vymazanie doterajšieho obsahu pamäte počítača povelom

```
NEW
```

Pri číslovaní programových riadkov je účelné voliť poradové čísla riadkov s istou rezervou, aby sme mohli v prípade potreby neskôr medzi už existujúce riadky vkladať nové riadky. Automatické číslovanie programových riadkov zabezpečíme povelom

```
AUTO
```

Po vložení tohto povelu sa automaticky nastaví číslo prvého riadka 10 a čísla riadkov sa budú zväčšovať s krokom 10.
Povel

```
AUTO m, n
```

kde *m* a *n* sú celé kladné čísla, zabezpečí, že sa automatické číslovanie vkladaných riadkov nastaví od čísla *m* s krokom *n*. Napr. po vložení povelu

```
AUTO 1ØØ, 5Ø
```

bude mať prvý vkladaný riadok číslo 100 a čísla riadkov sa budú zväčšovať s krokom 50. Čísla riadkov môžeme vkladať aj priamo; pôsobnosť povelu AUTO zrušíme napr. tak, že stlačíme kláves CLR a vložíme povel LIST, ktorý nám na obrazovke vypíše doteraz vloženú časť programu.

Priebežné zobrazovanie vkladaných povelov a programových riadkov na obrazovke zabezpečíme na PMD 85–1 tak, že pred začiatkom vkladania programu súčasne stlačíme klávesy SHIFT a PTL (resp. SHIFT a PAL na Didaktiku Alfa). Vkladané riadky sa zobrazujú „rolovaním" zdola nahor. Na PMD 85–2 je priebežné zobrazovanie vkladaných riadkov nastavené automaticky po zapnutí počítača (pozri poznámku 1.2).

Príklad 1.1

Vložíme krátky program, ktorý vypíše na obrazovke slovo AHOJ. Postupujeme takto:

a) Povelom NEW vymažeme pamäť počítača.

b) Povelom AUTO nastavíme automatické číslovanie riadkov od 10 s krokom 10.

c) Stlačením SHIFT a PTL nastavíme režim priebežného zobrazovania vkladaných riadkov. (Treba vykonať len na PMD 85–1!)

d) Vkladanie každého riadka ukončíme stlačením EOL.
Vložíme túto postupnosť povelov a programových riadkov:

```
NEW
AUTO
10 REM PRVY PROGRAM
20 PRINT "AHOJ"
30 END
40
LIST
```

Podčiarknuté časti riadkov sa vkladajú automaticky (čísla riadkov).
Po zobrazení čísla riadka 40 nezabudnite stlačiť kláves CLR!

1.2.2 PREZERANIE PROGRAMOV

Program v basicu, vložený do pamäte počítača, si vypíšeme na obrazovku povelom

```
LIST
```

ktorý spôsobí, že program sa postupne začne objavovať na obrazovke („roluje" zdola nahor). V prípade, že chceme výpis programu prerušiť, aby sme si mohli niektorú jeho časť dôkladnejšie prezrieť, zastavíme „rolovanie" programu stlačením klávesu STOP. Výpis programu môžeme obnoviť od ľubovoľného riadka povelom

```
LIST čr
```

kde *čr* je číslo programového riadka, od ktorého chceme program vypísať.

1.2.3 EDITOVANIE PROGRAMU

Opravy alebo zmeny už vloženého programu nazývame *editovaním programu*.

Ak zistíme, že program obsahuje chybné riadky, máme možnosť tieto riadky opraviť. Najjednoduchší spôsob opravy riadka je prepísanie celého riadka tak, že zapíšeme číslo chybného riadka a jeho nový text. Po stlačení EOL sa starý riadok prepíše novovloženým riadkom s rovnakým číslom.

Ak chceme niektorý programový riadok zrušiť, stačí vložiť jeho číslo a stlačiť kláves EOL.

Ak chceme vložiť nový riadok medzi dva už existujúce riadky, zvolíme si číslo vkladaného riadka tak, aby bolo väčšie ako číslo bezprostredne predchádzajúceho riadka a menšie ako číslo bezprostredne nasledujúceho riadka, vložíme toto číslo spolu s textom vkladaného riadka a stlačíme EOL. Ako sme už upozornili, riadky treba číslovať s dostatočnou rezervou, aby rozdiel medzi číslami susedných riadkov umožňoval neskoršie vsúvať prípadné nové riadky.

Možnosť pohodlnej opravy časti príkazového riadka bez nutnosti prepísať celý riadok poskytuje program *editor* spolu s *editovacími klávesmi*. Tento program je automaticky k dispozícii v rámci monitora a nemusíme ho osobitne privolávať. Pri editovaní riadka postupujeme nasledovne:

Povelom

```
LLIST čr
```

kde *čr* je číslo riadka, ktorý chceme opraviť, si vypíšeme programový riadok s číslom *čr* do dialógového riadka obrazovky. Pod prvým znakom dialógového riadka sa zobrazí kurzor, ktorý označuje pozíciu riadka, do ktorej editor vloží na základe stlačenia príslušného klávesu znak (písmeno, číslicu, znamienko).

Ak chceme opraviť niektorý znak v dialógovom riadku, nastavíme si najprv kurzor postupným stláčaním klávesu → (posun o jeden znak vpravo), resp. klávesu ← (posun o jeden znak vľavo) pod znak, ktorý chceme opraviť, a tento znak prepíšeme. Prepísaním znaku sa kurzor automaticky posunie o jednu pozíciu vpravo.

Ak chceme kurzor z ľubovoľnej pozície nastaviť na začiatok dialógového riadka, stlačíme kláves ↖ (HOME), ak chceme kurzor nastaviť na posledný vložený znak v dialógovom riadku, stlačíme kláves END (END LINE). Stlačením klávesu DEL vymažeme znak, pod ktorým sa momentálne nachádza kurzor, a celá časť dialógového riadka napravo od kurzora sa posunie o jeden znak doľava. Stlačením klávesu INS uvoľníme miesto na vsunutie nového znaku do už existujúceho textu (celá časť dialógového riadka, nachádzajúca sa vpravo od kurzora vrátane znaku, pod ktorým sa nachádza kurzor, sa posunie o jednu pozíciu doprava). Po vykonaní všetkých opráv v dialógovom riadku odošleme opravený riadok do pamäte stlačením EOL. Tento spôsob úprav používame aj v prípade „preklepov" pri vkladaní nového riadka, pri chybnom vložení povelu interpreta alebo údajov.

Príklad 1.2

Vypíšeme si program z príkladu 1.1. Vložíme povel LIST a na obrazovke sa zjaví výpis:

```
LIST
1Ø REM PRVY PROGRAM
2Ø PRINT "AHOJ"
3Ø END
OK
```

Správa OK informuje o tom, že požadovaná činnosť bola úspešne ukončená.

Predpokladajme, že program chceme upraviť tak, že medzi riadky s číslom 10 a 20 vložíme riadok s číslom 15, ktorý obsahuje „prázdny" príkaz PRINT na vynechanie jedného riadka na obrazovke, riadok s číslom 20 zmeníme tak, aby sa text AHOJ vypisoval s medzerami (teda v tvare A H O J), a zrušíme riadok s číslom 30, pretože program môže fungovať aj bez neho (spôsob používania príkazov REM, PRINT a END podrobnejšie vysvetlíme v kapitole 1.4). Postupujeme takto:

a) Vložíme nový riadok

```
15 PRINT
```

b) Vložíme povel

```
LLIST 2Ø
```

ktorý spôsobí, že do dialógového riadka sa vypíše programový riadok

```
2Ø PRINT "AHOJ"
```

pričom kurzor (znak podčiarknutia) sa nastaví pod prvým znakom. Posunieme 11-násobným stlačením klávesu → kurzor pod znak H a stlačíme INS, čím dostaneme

```
2Ø PRINT "A HOJ"
```

Ďalej posunieme kurzor pod písmeno O a stlačíme INS a pod písmeno J a stlačíme INS. Dostaneme požadovanú zmenu

```
2Ø PRINT "A H O J"
```

a opravený riadok odošleme späť do pamäte stlačením EOL.

c) Riadok s číslom 30 vymažeme z programu tak, že vložíme do dialógového riadka číslo 30 a stlačíme EOL.

Ak teraz znova vypíšeme upravený program povelom LIST, dostaneme

```
LIST
1Ø REM PRVY PROGRAM
15 PRINT
```

```
20 PRINT "A H O J"
OK
```

1.2.4 SPUSTENIE A POZASTAVENIE VYKONÁVANIA PROGRAMU

Vykonávanie programu, ktorý sa nachádza vo vnútornej pamäti počítača, odštartujeme povelom

```
RUN
```

V tomto prípade sa program začne vykonávať od prvého programového riadka. Ak chceme, aby sa program začal vykonávať od iného programového riadka, vložíme povel

```
RUN čr
```

kde *čr* je číslo riadka, od ktorého chceme program odštartovať.

Vykonávaný program vypisuje na obrazovke rôzne správy, požaduje od nás vkladanie údajov, realizuje výpočty alebo iné spôsoby spracovania údajov a zobrazuje výsledky spracovania. Ak chceme prácu programu počas jeho behu prerušiť, stlačíme kláves STOP. Vykonávanie programu sa pozastaví a môžeme si napríklad prezrieť doteraz zobrazené výsledky. Ak chceme vykonávanie programu obnoviť od miesta, v ktorom sme ho prerušili, vložíme povel

```
CONT
```

1.2.5 POVELY NA PRÁCU S MAGNETOFÓNOM

Program, ktorý vložíme do pamäte počítača, sa vypnutím počítača alebo vložením povelu NEW stráca, počítač ho „zabudne". Aby sme programy nemuseli vždy znovu vkladať cez klávesnicu, potrebujeme si ich uchovať v takej forme, aby sme ich mohli použiť aj v budúcnosti. Na tieto účely slúži kazetový magnetofón. Programy zapisujeme na kazetu podobne ako napríklad hudobný záznam a môžeme ich z kazety kedykoľvek načítať do pamäte počítača. Na počítačoch PMD 85–1 a Didaktik Alfa môže jedna kazeta obsahovať maximálne 64 súborov s programami alebo údajmi (na PMD 85–2 až 99). Tieto súbory označujeme číslami 0–63, ktoré im prideľujeme pri zápise na kazetu.

a) Záznam programu na kazetu

Z vnútornej pamäte počítača zapíšeme program na kazetu magnetofónu povelom

```
SAVE n "názov
```

kde *n* je číslo programového súboru, pod ktorým sa má tento uložiť na kazetu, *názov* je maximálne 8-znakový názov programového súboru, ktorý bude spolu s číslom súčasťou jeho hlavičky.

Predpokladajme napríklad, že chceme z vnútornej pamäte počítača zapísať na kazetu program z príkladu 1.2 pod názvom AHOJ ako súbor číslo 10. Kazetu nastavíme za posledný záznam, napíšeme povel

SAVE 1Ø "AHOJ

zapneme magnetofón na nahrávanie a až potom odošleme povel SAVE stlačením EOL. Po úspešnom zápise programu na kazetu sa v dialógovom riadku zobrazí správa OK, po ktorej zastavíme magnetofón.

Hoci nie je zakázané zapisovať na kazetu programové súbory s rovnakými číslami, odporúčame voliť vždy pre rôzne súbory rôzne čísla (a rôzne názvy) a pri zápise na kazetu voliť čísla súborov v prirodzenom vzostupnom poradí.

b) Načítanie programu z kazety

Program načítame z kazety magnetofónu do vnútornej pamäte počítača povelom

```
LOAD n
```

kde *n* je číslo programového súboru, pod ktorým je tento uložený na kazete.

Predpokladajme napríklad, že chceme načítať z kazety programový súbor s číslom 10 a názvom AHOJ. Nastavíme kazetu pred tento súbor, vložíme povel

LOAD 1Ø

odošleme ho stlačením EOL a zapneme magnetofón na prehrávanie. V dialógovom riadku sa zjaví hlavička čítaného súboru

1Ø>AHOJ

Po úspešnom načítaní programu do pamäte počítača sa v dialógovom riadku zobrazí správa OK. Vtedy vypneme magnetofón. Načítaný program potom môžeme odštartovať alebo editovať.

V prípade, že kazeta bola nastavená niekoľko súborov pred programovým súborom, ktorý chceme načítať, zobrazujú sa nám postupne v dialógovom riadku hlavičky všetkých týchto súborov, do pamäte počítača sa však načíta iba súbor požadovaný povelom LOAD.

c) Kontrola správnosti a vyhľadanie konca záznamu

Na kazete spravidla uchovávame viac súborov, a preto je dôležité vedieť si nastaviť kazetu na koniec určitého súboru. Na tieto účely slúži povel

```
CHECK n
```

kde *n* je číslo súboru, za koniec ktorého chceme nastaviť kazetu.

Povel CHECK má pritom ešte jeden účinok: kontroluje správnosť záznamu na kazete. Po nájdení konca požadovaného súboru a zistení správnosti záznamu sa v dialógovom riadku zjaví správa OK. V opačnom prípade (záznam je chybný) sa v dialógovom riadku zjaví správa

FILE ERROR

(chyba súboru). Hneď po zápise programu na kazetu povelom SAVE odporúčame skontrolovať správnosť záznamu povelom CHECK a v prípade chybného záznamu zápis zopakovať. Pred zapisovaním nového súboru povelom SAVE alebo čítaním povelom LOAD treba najprv povelom CHECK nastaviť kazetu za súbor, za ktorý chceme nový súbor zapísať, alebo pred súbor, ktorý chceme načítať. Predpokladajme napr., že programový súbor s číslom 10 chceme zapísať za súbor s číslom 9. Vložíme najprv povel

CHECK Ø9

odošleme ho stlačením EOL a zapneme magnetofón na nahrávanie. Podobne ako v prípade povelu LOAD sa v dialógovom riadku obrazovky postupne zjavujú hlavičky s číslami a názvami všetkých súborov, ktoré predchádzajú hľadanému súboru. Po nájdení konca súboru s číslom 9 a vypísaní správy OK zastavíme magnetofón. Potom môžeme pristúpiť k zápisu súboru s číslom 10 povelom SAVE.

Poznámka 1.3

1. Na PMD 85–2 možno čítať programové a údajové súbory zapísané na kazetu na PMD 85–1, predpokladá to však väčšie medzery na kazete medzi jednotlivými súbormi. Súbory s číslami menšími ako 64 vytvorené na PMD 85–2 možno čítať aj na PMD 85–1.

2. Na PMD 85–2 existuje možnosť ochrany programov proti prezeraniu a kopírovaniu. Program, ktorý chceme chrániť, treba nahrať na kazetu tak, aby prvým znakom názvu bola hviezdička, napr.

SAVE 1Ø "*AHOJ

Keď sa povelom LOAD prečíta program, ktorého názov sa začína hviezdičkou, program sa automaticky odštartuje. Beh programu nemožno zastaviť stlačením klávesu STOP. Keď sa vykonávanie programu skončí, program sa automaticky vymaže z pamäte počítača, takže ho nemôžeme vypísať povelom LIST.

1.2.6 VYUŽITIE FUNKČNÝCH KLÁVESOV

Jednou z možností využívania funkčných klávesov KØ, K1, ..., K11 (na PMD 85), resp. Ⓘ, ①, ..., ⑪ (na Didaktiku Alfa) je zjednodušenie vkladania povelov interpreta, resp. príkazov basicu. Pomocou týchto klávesov možno do vyhradenej časti pamäte vložiť najčastejšie používané kľúčové slová – názvy niektorých povelov a príkazov – a tieto potom vkladať do dialógového riadka jednoduchým stlačením príslušných funkčných klávesov. Postupujeme takto:

a) Do dialógového riadka vložíme príslušné kľúčové slovo, napr. povel LIST.

b) Stlačíme kláves WRK (na PMD 85), resp. kláves ◯ (na Didaktiku Alfa).

c) Stlačíme príslušný funkčný kláves, do ktorého chceme uložiť kľúčové slovo, napr. KØ na PMD 85 alebo Ⓘ na Didaktiku Alfa.

Tým si zabezpečíme, že vždy, keď stlačíme funkčný kláves KØ, resp. Ⓘ , zapíše sa do dialógového riadka (od pozície, na ktorú je nastavený kurzor) povel LIST. Stlačením EOL potom tento povel odošleme na spracovanie.

Príklad 1.3

Predpokladajme, že do funkčného klávesu K1 chceme vložiť kľúčové slovo PRINT. Vložíme toto slovo najprv do dialógového riadka, stlačíme WRK a kláves K1. Ak potom chceme vložiť napr. príkazový riadok

2Ø PRINT "AHOJ"

môžeme postupovať nasledovne:

Vložíme číslo riadka a medzeru, teda

2Ø

(podčiarknutie znázorňuje polohu kurzora). Stlačíme kláves K1, čím sa dialógový riadok doplní o kľúčové slovo PRINT, teda

20 PRINT

Z klávesnice doplníme zvyšok príkazového riadka:

20 PRINT "AHOJ"

a príkazový riadok odošleme stlačením EOL.

Klávesnica obsahuje 12 funkčných klávesov, ktoré môžeme využiť na vloženie kľúčových slov alebo iných textov. V prípade potreby (po vyčerpaní základných 12 klávesov) môžeme počet takto vložených kľúčových slov rozšíriť na 24 tým, že pri ich vkladaní a pri ich privolávaní do dialógového riadka stlačíme s príslušným funkčným klávesom aj kláves SHIFT. Do každého funkčného klávesu teda môžeme vložiť dve kľúčové slová: jedno bez stlačenia klávesu SHIFT a druhé s jeho stlačením.

Po vypnutí počítača alebo stlačení klávesov SHIFT a RST sa obsah funkčných klávesov stráca.

1.3 ZÁKLADNÉ PRVKY JAZYKA BASIC

Jazyk basic na počítačoch PMD 85 a Didaktik Alfa je orientovaný na riešenie výpočtových úloh, prácu s textami a „kreslenie na obrazovke" (tzv. *počítačovú grafiku*). Poskytuje aj určité možnosti na prácu so súbormi údajov uchovávanými na kazete. V prípade pripojenia vhodnej tlačiarne umožňuje aj tlačenie výsledkov a grafických zobrazení na tlačiarni.

Počas práce s počítačom sa program, údaje a výsledky ich spracovania uchovávajú v pamäti počítača. Pre správne plánovanie riešenia úloh treba mať jasnú predstavu o tom, koľko pamäte máme k dispozícii. Jednotkou, pomocou ktorej meriame rozsah pamäte počítača, je *slabika* čiže *bajt* (B, anglicky Byte). Ide o pamäťovú jednotku, ktorá umožňuje uchovať jeden znak (napr. písmeno). Maximálny rozsah pamäte sa vyjadruje pomocou vyššej mernej jednotky, nazývanej *kilobajt* (KB). Platí

$1\ KB = 2^{10}\ B = 1024\ B$

teda jeden kilobajt obsahuje 1024 bajtov. Basic na počítačoch PMD 85 a Didaktik Alfa poskytuje používateľovi na jeho programy a údaje približne 19 KB vnútornej pamäte počítača (vyše 19 000 znakov), na Atari až 37 KB.

V tejto kapitole sa oboznámime so základnými poznatkami potrebnými na správny zápis programov v basicu: s formátom príkazových riadkov, konštantami a premennými, štandardnými funkciami a výrazmi a s prehľadom všetkých príkazov jazyka.

1.3.1 PRIAMY A PROGRAMOVÝ REŽIM

Osobné počítače možno používať ako lepšiu „kalkulačku". Každý riadok bez poradového čísla interpret basicu pokladá za príkaz, ktorý sa má okamžite vykonať. V tomto prípade hovoríme o *priamom režime* práce s interpretom basicu.

Príklad 1.4

Ak chceme napríklad vypočítať hodnotu výrazu 376:125, zapíšeme do dialógového riadka

```
PRINT 376/125
```

a stlačíme EOL. Na obrazovke sa okamžite zjaví výsledok a správa OK, teda

```
3.ØØ8
OK
```

Základným spôsobom práce s osobnými počítačmi je však tzv. *programový režim*, keď zostavujeme a spúšťame ucelené programy. Riadok, ktorý sa začína číslom, interpret pokladá za programový riadok a uloží ho do programovej časti pamäte počítača. Postupnosť takýchto riadkov nazývame programom. Vykonávanie programov sa štartuje povelom RUN. V tejto knihe sa zaoberáme iba programovým režimom práce.

1.3.2 FORMÁT PROGRAMOVÝCH RIADKOV

Každý programový riadok sa začína číslom, ktoré určuje poradie, v ktorom sa riadok zaraďuje do pamäte počítača. Ide o celé kladné číslo v rozsahu 0 až 65535 (na PMD 85–2 od 0 do 32767).

Jeden programový riadok môže obsahovať jeden alebo viac príkazov (v tomto zmysle hovoríme o jednopríkazových a viacpríkazových riadkoch). Ak riadok obsahuje viac príkazov, jednotlivé príkazy oddeľujeme znakom : (dvojbodka). Príkaz musí byť ukončený v programovom riadku (nemožno pokračovať v príkaze v ďalšom programovom riadku).

Maximálny počet znakov programového riadka je 80. V jednom riadku obrazovky môžeme zobraziť iba 48 znakov; ak vkladáme ďalšie znaky, posúva sa dialógový riadok postupne vľavo za okraj obrazovky. Pri vložení 70. znaku v riadku počítač upozorní zvukovým signálom („pípnutím") na blížiaci sa koniec programového riad-

ka. Pri zápise prvých programov odporúčame využívať iba 48 znakov, teda viditeľnú časť obrazovky.

```
Príklad 1.5
  Jednopríkazové riadky:
  1Ø REM PRIKLAD
  2Ø PRINT "VLOZ CISLA A,B,C"
  3Ø INPUT A,B,C
  Viacpríkazové riadky:
  5Ø PRINT "VLOZ X"; : INPUT X
  6Ø FOR I=1 TO 5 : S=S+A(I) : NEXT I
  9Ø A=1 : B=3 : Z=Ø.5 : K=Ø
```

Na zápis programových riadkov používame 26 písmen latinskej abecedy, číslice, operačné znamienka (napr. +, –, /, * a pod.), interpunkčné znamienka a niektoré špeciálne znaky. Aby sme odlíšili číslo nula od písmena O, nulu vždy prečiarkneme, teda O je písmeno a Ø je nula. Rovnakým spôsobom sa nula zobrazuje aj na obrazovke.

1.3.3 KONŠTANTY, PREMENNÉ A POLIA

Údaje, ktoré chceme programom spracovávať, sa v programe zapisujú buď priamo svojou hodnotou, alebo symbolicky názvom. Podľa toho rozlišujeme *konštanty* (zápis hodnotou) a *premenné* (zápis názvom). Môžeme pracovať aj so skupinami premenných, nazývanými *polia*.

a) Konštanty

V jazyku basic pre počítače PMD 85 a Didaktik Alfa používame dva typy konštánt: číselné konštanty a textové konštanty (reťazce).

Číselné konštanty môžeme vkladať buď v tvare bežného desiatkového čísla (znamienko a číslo s prípadnou desatinnou bodkou), alebo v exponenciálnom tvare

$$aEb$$

ktorý vyjadruje číslo

$$a \times 10^b$$

kde *a* je mantisa, zapísaná ako číslo v bežnom dekadickom tvare, *E* je vyhradený znak a *b* je jednociferný alebo dvojciferný exponent s prípadným znamienkom. V bežnom dekadickom tvare možno za-

písať maximálne šesťmiestne čísla, v exponenciálnom tvare možno zapísať čísla s absolútnou hodnotou od 10^{-38} do 10^{38}.

Príklad 1.6

Príklady zápisu číselných konštánt v bežnom dekadickom tvare:

```
2.14
-27
128.55
0
.125
```

Všimnime si, že pri zápise konštanty menšej ako 1 možno vynechať nulu pred desatinnou bodkou.

Príkazy zápisu číselných konštánt v exponenciálnom tvare:

zápis	hodnota
1.12E+6	1120000
-2.605E+2	-260.5
5.7E-11	0.000000000057

V exponenciálnom tvare zapisujeme konštanty s veľmi veľkou alebo veľmi malou absolútnou hodnotou.

Textová konštanta (reťazec) je postupnosť znakov uzatvorená úvodzovkami. Môže obsahovať akékoľvek znaky prípustné v jazyku basic (teda aj číslice!) s výnimkou ohraničujúcich úvodzoviek.

Príklad 1.7

Zápis textových konštánt:

```
"DIDAKTIK ALFA"
"STLAC KLAVES EOL!"
"VYSLEDKY VYPOCTU V %"
"X ="
"1989"
```

b) Premenné

Na rozdiel od konštánt sa premenné v programe nezapisujú hodnotou, ale menom a svoju hodnotu môžu počas vykonávania programu meniť. Podobne ako v prípade konštánt rozlišujeme číselné a reťazcové (textové) premenné.

Meno *číselnej premennej* obsahuje jeden alebo dva znaky, pričom prvým znakom musí byť písmeno a druhým znakom písmeno alebo číslica.

Meno *reťazcovej premennej* sa skladá z dvoch až troch znakov,

pričom prvým znakom je vždy písmeno a posledným znakom $ (dolár). Ak je názov reťazcovej premennej trojznakový, potom druhým znakom môže byť písmeno alebo číslica.

V prípade, že použijeme dlhší názov číselnej alebo reťazcovej premennej, interpret basicu G z neho akceptuje prvé dva znaky (v prípade číselnej premennej), alebo prvé dva znaky a posledný znak $ (v prípade reťazcovej premennej). Napríklad premennú CISLO spracuje ako CI, premennú SLOVO$ ako SL$ a názvy CISLO, CISLA, CIS, ale aj CIEL teda označujú tú istú premennú CI. S ohľadom na úsporu pamäte a odstránenie chýb, ktoré by mohli vzniknúť z rovnakého pomenovania rôznych premenných, neodporúčame používať v basicu G dlhšie názvy.

Ako mená premenných nepoužívajte ani kľúčové slová basicu (názvy logických operátorov, príkazov a funkcií).

Jedna číselná premenná zaberá v pamäti počítača 4 B, maximálna dĺžka reťazcovej premennej je 255 B (t. j. 255 znakov). V Atari basicu môžu reťazcové premenné dosahovať aj väčšiu dĺžku.

Príklad 1.8

Správne zapísané mená číselných a reťazcových premenných:

X	číselná
A1	číselná
Y0	číselná
ZZ	číselná
Q$	reťazcová
T1$	reťazcová
PP$	reťazcová

Nesprávne zapísané mená číselných premenných:

0X	začína sa číslicou
Q$	nesprávny druhý znak
!C	nesprávny prvý znak

Nesprávne zapísané mená reťazcových premenných:

TXT	nesprávny koncový znak
R?$	nesprávny druhý znak
G $	meno nesmie obsahovať medzeru

Poznámka 1.4

1. Na rozdiel od PMD 85–1 a Didaktiku Alfa sa na počítači PMD 85–2 môžu názvy premenných tvoriť z veľkých aj z malých písmen, pričom Y1 a y1 sú názvy dvoch rôznych číselných premenných a T$ a t$ sú názvy dvoch rôznych reťazcových premenných.

2. Na PMD 85–2 môžu textové konštanty obsahovať aj malé písmená, napr. "Vloz cislo Z:".

3. Okrem číselných a textových konštánt možno v basicu na PMD 85–2 pracovať aj so šestnástkovými (hexadecimálnymi) konštantami, ktoré sa zapisujú v tvare

'kons

t. j. začínajú znakom apostrofu, za ktorým nasleduje maximálne štvorciferné šestnástkové číslo (tvorené šestnástkovými číslicami 0, 1, 2, . . . , 9, A, B, C, D, E, F). Rozsah šestnástkových konštánt je od 'Ø do 'FFFF.

c) Polia

Premenné zapísané iba menom nazývame jednoduchými premennými. Pri programovaní zložitejších úloh, v ktorých pracujeme s veľkými skupinami premenných, by sme s jednoduchými premennými nevystačili. Napríklad na označenie všetkých prvkov tabuľky, ktorá má 50 riadkov a 50 stĺpcov, by sme potrebovali 2500 rôznych názvov jednoduchých premenných. Programy na spracovanie údajov takejto tabuľky by boli príliš ťažkopádne a vo väčšine prípadov by sa pravdepodobne ani nezmestili do pamäte počítača. Na prácu s väčšími skupinami údajov rovnakého typu používame v basicu a v ďalších jazykoch tzv. *polia* a *indexované premenné*.

Poľom nazývame usporiadanú skupinu premenných s rovnakým menom. V jazyku basic na PMD 85 a Didaktiku Alfa môžeme používať jednorozmerné, dvojrozmerné a trojrozmerné polia. Na prvky poľa sa v programe odvolávame menom poľa a uvedením jedného, dvoch alebo troch indexov (čiže celých čísel určujúcich polohu prvku v poli) oddelených navzájom čiarkami a uzavretých v okrúhlych zátvorkách. Prvky polí preto nazývame indexovanými premennými.

Prvky polí v jednotlivých rozmeroch číslujeme od nuly. Napríklad jednorozmerné pole s menom A, ktoré obsahuje 5 prvkov, si môžeme predstaviť ako riadok

A(Ø)	A(1)	A(2)	A(3)	A(4)

Dvojrozmerné pole 9 prvkov s menom B si môžeme predstaviť ako tabuľku

B(Ø,Ø)	B(Ø,1)	B(Ø,2)
B(1,Ø)	B(1,1)	B(1,2)
B(2,Ø)	B(2,1)	B(2,2)

Trojrozmerné pole si môžeme predstaviť ako skupinu tabuliek, kde tretí rozmer vždy udáva poradové číslo tabuľky.

Podobne ako jednoduché premenné aj polia môžu byť dvoch typov: číselné a reťazcové. To isté pole nemôže zároveň obsahovať číselné a reťazcové premenné. Typ poľa je určený jeho menom, vytváraným podľa rovnakých pravidiel ako mená jednoduchých číselných a reťazcových premenných.

Na to, aby sme interpretu basicu oznámili, že chceme pracovať s poľami, a rezervovali potrebné miesto v pamäti počítača, používame príkaz

DIM zoznam

kde *zoznam* obsahuje tzv. deklarácie mien a rozsahov polí oddelené čiarkami. Deklarácia poľa má tvar

$meno_1(r_1)$

alebo

$meno_2(r_1,r_2)$

alebo

$meno_3(r_1,r_2,r_3)$

kde $meno_1$, $meno_2$ a $meno_3$ sú mená polí a r_1, r_2, r_3 sú kladné celočíselné konštanty, ktoré udávajú maximálny rozsah prvého, druhého a tretieho rozmeru poľa.

Príklad 1.9

Príkaz

```
2Ø DIM A(2Ø),B(15,5),T$(6Ø)
```

deklaruje jednorozmerné číselné pole A, ktoré rezervuje miesto pre 21 premenných, dvojrozmerné číselné pole B, ktoré rezervuje miesto pre 96 premenných (16 riadkov a 6 stĺpcov) a jednorozmerné reťazcové pole T$, ktoré rezervuje miesto pre 61 premenných (nezabúdajte, že číslovanie premenných v poli sa pre každý rozmer začína od nuly).

1.3.4 ŠTANDARDNÉ FUNKCIE

Na rýchlu realizáciu niektorých výpočtov, operácií s reťazcami a ďalších činností používame *štandardné funkcie* jazyka basic.

Všeobecný tvar štandardnej funkcie:

meno (argumenty)

kde *meno* je kľúčové slovo, ktoré sa skladá z 3–6 znakov, a v zozname argumentov v zátvorkách uvádzame 1–3 *argumenty*

(v závislosti od konkrétnej funkcie). Výnimočne môže byť funkcia aj bez argumentov. Funkciu používame tak, že ju zaradíme na príslušnom mieste do programu a vyhodnotíme dosadením konkrétnych hodnôt (konštánt, premenných alebo výrazov) za argumenty.

Súčasťou jazyka basic pre PMD 85 a Didaktik Alfa sú tri typy štandardných funkcií:

- aritmetické funkcie,
- funkcie na prácu s reťazcami,
- špeciálne funkcie.

a) Aritmetické funkcie

Aritmetické funkcie majú trojznakové meno a obsahujú iba jeden argument. Meno aritmetickej funkcie je nositeľom číselnej hodnoty. Ide o tieto funkcie:

SIN(X)	sin x, pre x v radiánoch,
COS(X)	cos x, pre x v radiánoch,
TAN(X)	tg x, pre x v radiánoch,
ATN(X)	arctg x, výsledná hodnota v radiánoch,
LOG(X)	prirodzený logaritmus x (základ logaritmu je e=2.71828),
EXP(X)	exponenciálna funkcia e^x, kde e je základ prirodzených logaritmov,
ABS(X)	absolútna hodnota x,
SQR(X)	druhá odmocnina z x,
INT(X)	najväčšie celé číslo menšie alebo rovné x
SGN(X)	znamienko x, kde SGN(X)=−1, ak x<Ø, SGN(X)= Ø, ak x=Ø, SGN(X)= 1, ak x>Ø.

Pri používaní goniometrických funkcií pracujeme s radiánmi. *Radián* je jednotka oblúkovej miery uhla. Vzťah medzi radiánmi a stupňami je daný vzorcom

$$\text{radiány} = \frac{\text{stupne} \cdot \pi}{180}$$

kde π je oblúková miera priameho uhla (používa sa približná hodnota 3.14159).

Ak v programe požadujeme výpočet ďalších matematických funkcií, musíme si ich vytvoriť na základe známych vzťahov, napr.

$$\text{cotg } x = \frac{1}{\text{tg } x} = \frac{\cos x}{\sin x}$$

alebo

$$\log_a x = \frac{\log_e x}{\log_e a}$$

Na PMD 85–2 máme možnosť pracovať s goniometrickými funkciami v radiánoch aj v stupňoch. Na prepínanie z radiánov na stupne a zo stupňov na radiány slúžia príkazy RAD a DEG. Po zapnutí počítača a privolaní interpreta je automaticky nastavený režim radiánov. Na stupňový režim ho zmeníme vložením príkazu DEG v príslušnom mieste programu. K režimu radiánov sa v prípade potreby vrátime vložením príkazu RAD.

b) Funkcie na prácu s reťazcami

Funkcie na prácu s reťazcami majú jeden až tri argumenty (reťazcové alebo číselné) a ich výsledkom je reťazec alebo číslo. Ide o tieto funkcie:

ASC(X$)	číselná hodnota v desiatkovej sústave, ktorá zodpovedá ASCII (čítaj aski) kódu prvého znaku reťazca X$,
CHR$(I)	znak, ktorý zodpovedá desiatkovej hodnote I ASCII kódu (I je celé číslo z intervalu 0–255),
LEFT$(X$,I)	vyberie z reťazca X$ prvých I znakov (ľavý krajný podreťazec v dĺžke I znakov),
RIGHT$(X$,I)	vyberie z reťazca X$ posledných I znakov (pravý krajný podreťazec v dĺžke I znakov),
MID$(X$,I,J)	vyberie z reťazca X$ časť v dĺžke J znakov, ktorá sa začína I-tym znakom,
LEN(X$)	počet znakov reťazca X$,
STR$(X)	premení číslo X na reťazec číselných znakov (znamienko, číslice, desatinná bodka, znak E),
VAL(X$)	premení reťazec číselných znakov X$ na číslo.

Po vykonaní funkcie sa názvu funkcie priradí príslušný reťazec (funkcie, ktorých mená sa končia znakom $) alebo číslo (funkcie, ktorých mená sa končia iným znakom ako $).

Počítač spracováva jednotlivé znaky v číselnom tvare pomocou tzv. ASCII kódu (American Standard Code for Information Interchange). Číselná hodnota znaku vyjadrená v desiatkovej sústave je z intervalu 0 až 255. ASCII kód znakov, používaných pri programo-

vaní v basicu na počítačoch PMD 85 a Didaktik Alfa (a zobraziteľných na týchto počítačoch) sa začína od hodnoty 32 (medzera) a končí sa hodnotou 95 (znak _). Znaky s hodnotami 96 až 127 zodpovedajú malým písmenám a niektorým špeciálnym znamienkam (používajú sa na PMD 85–2). V intervale 0 až 31 sú špeciálne znaky, ktoré sa nezobrazujú a slúžia na riadenie niektorých činností. Pre hodnoty kódu z intervalu 128 až 255 sa na PMD 85 a Didaktiku Alfa všetky znaky znova opakujú. Tabuľku ASCII kódu uvádzame v prílohe 3. Hodnoty ASCII kódu jednotlivých zobraziteľných znakov možno zistiť funkciou ASC, znaky priradené jednotlivým hodnotám ASCII kódu môžeme zistiť funkciou CHR$.

Poznámka 1.5
Na počítači PMD 85–1 umožňuje funkcia VAL previesť reťazec číslic na desiatkové číslo. Na PMD 85–2 táto funkcia navyše vyhodnocuje aritmetický výraz, t. j. pre zadaný vzorec vráti jeho hodnotu (pozri kapitolu 1.4).

c) Špeciálne a pomocné funkcie

Tieto funkcie umožňujú regulovať úpravu výstupov na obrazovku, generovať pseudonáhodné čísla, získavať informácie o rozsahu nevyužitej pamäte, využívať v programe funkčné klávesy a vykonávať niektoré činnosti s pamäťou počítača. Ide o nasledujúce funkcie:

TAB(X)	vynechanie X pozícií od začiatku riadka pri výpise na obrazovku,
SPC(X)	vynechanie X pozícií od posledného zobrazeného znaku v riadku,
FRE(X)	zistenie rozsahu voľnej pamäťovej oblasti pre program a číselné údaje (v B),
FRE(X$)	zistenie rozsahu voľnej pamäťovej oblasti pre reťazce (v B),
RND(X)	generuje pseudonáhodné čísla z otvoreného intervalu 0 až 1; ak X=0, generuje sa pri každom použití to isté číslo; ak X>0, generuje sa pri každom použití iné číslo,
INKEY	načítanie hodnoty stlačeného funkčného klávesu KØ až K11; pri stlačení klávesu sa menu funkcie priradí číselná hodnota 0 až 11 podľa čísla stlačeného klávesu; ak nebol stlačený žiaden kláves, priraďuje sa hodnota 255,
PEEK(X)	načítanie hodnoty z pamäťového miesta X,

INP(I)	prečítanie obsahu portu I,
USR(I)	privolanie a spustenie podprogramu v strojovom kóde, uloženého v pamäti počítača od adresy I,
BIT X,I	určenie hodnoty bitu I v bajte X (argument I je z intervalu 0 až 7, výsledkom je 0 alebo 1).

Poznámka 1.6

Okrem funkcií TAB a SPC, používaných na úpravu výstupov, poskytuje PMD 85–2 dve nové funkcie: AT a INK.

Funkcia AT umožňuje nastaviť kurzor (a tým aj miesto, kde sa bude vypisovať) na ľubovoľné miesto obrazovky. Používa sa v tvare

AT r,s

kde *r* je číslo riadka obrazovky (od 0 do 24), *s* je číslo stĺpca obrazovky (od 0 do 47).

Funkcia INK umožňuje riadiť jas a blikanie pri výpisoch na obrazovku. Používa sa v tvare

INK(p)

kde *p* je číselný parameter, pomocou ktorého sa zadáva spôsob výpisu:

p=0	plný jas,
p=1	polojas,
p=2	blikanie,
p=4	inverzný výpis (na bielom podklade tmavé písmená).

Na začiatku práce s počítačom je automaticky nastavený výpis plným jasom (na čiernom pozadí biele písmená) bez blikania. Kombinovanie spôsobov výstupu riadime súčtom príslušných hodnôt. Ak chceme napríklad blikanie polojasom, použijeme parameter $p=1+2=3$. Teda napríklad príkazom

```
10Ø PRINT AT 2,1Ø:INK(3);"Ahoj"
```

vypíšeme od desiatej pozície druhého riadka obrazovky slovo Ahoj polojasom s blikaním.

1.3.5 VÝRAZY A RELÁCIE

Z konštánt, premenných a funkcií vytvárame pomocou aritmetických a logických operátorov a zátvoriek *aritmetické*, *logické* a *reťazcové* výrazy. Výsledkom vyhodnotenia aritmetického výrazu je číslo, výsledkom vyhodnotenia logického výrazu je logická 1 (pravda) alebo logická 0 (nepravda), výsledkom vyhodnotenia reťazcového výrazu je reťazec. Výrazy porovnávame v *reláciách*.

a) Aritmetické výrazy

Na zápis aritmetických výrazov používame nasledujúce aritmetické operátory:

+	sčítanie	(napr. A+B)
–	odčítanie	(napr. A–B)

*	násobenie	(napr. A*B)
/	delenie	(napr. A/B)
^	umocňovanie	(napr. A^ B)

Povolená je tiež operácia „unárne mínus", napr. –A, resp. unárne plus, napr. +A.

Aritmetický výraz vytvárame pomocou číselných konštánt, číselných premenných (jednoduchých aj indexovaných), funkcií s číselným výsledkom, aritmetických operátorov a ľavých a pravých okrúhlych zátvoriek. Počet ľavých a pravých zátvoriek v aritmetickom výraze sa musí zhodovať. Na rozdiel od algebraického zápisu vzorcov musíme v aritmetickom výraze vždy uvádzať operátor násobenia.

V aritmetickom výraze bez zátvoriek sa operácie vykonávajú podľa týchto priorít:

1. funkcie,
2. umocnenie,
3. unárne mínus, resp. unárne plus,
4. násobenie a delenie,
5. sčítanie a odčítanie.

V prípade použitia zátvoriek sa najprv vyhodnocujú časti výrazu v najvnútornejších zátvorkách, potom v ďalšej dvojici zátvoriek atď. Pri rovnakých prioritách (napr. násobenie a delenie) sa pri vyhodnocovaní postupuje zľava doprava.

Príklad 1.10

Správne zapísané aritmetické výrazy:

A/B*X	algebraicky $\frac{ax}{b}$
ABS(2*X–Y)	algebraicky $\lvert 2x-y \rvert$
B^2–4*A*C	algebraicky $b^2 - 4ac$
X^(1/3)	algebraicky $\sqrt[3]{x}$
((X+1)^ 2+B)^ 3	algebraicky $((x+1)^2+b)^3$

Nesprávne zapísané aritmetické výrazy:

–X*–2	dva aritmetické operátory, správne –X*(–2)
A(B+C)	chýba operátor násobenia, správne A*(B+C)
(1+ABS(X)	chýba pravá zátvorka, správne (1+ABS(X))

Na porovnávanie aritmetických výrazov používame nasledujúce relačné operátory:

=	rovný	(napr. A=B)
<	menší	(napr. A<B)
<=	menší alebo rovný	(napr. A<=B)
>	väčší	(napr. A>B)
>=	väčší alebo rovný	(napr. A>=B)
<>	nerovný	(napr. A<>B)

Pomocou relačných operátorov vytvárame relácie.

Príklad 1.11

Zápis aritmetických relácií:

X<=2*Y	algebraicky $x \leqq 2y$
3*X-1>0	algebraicky $3x-1>0$
A*B-C*D<>0	algebraicky $ab-cd \neq 0$

b) Logické výrazy

Logický výraz vytvárame z konštánt, premenných, funkcií, aritmetických, relačných a logických operátorov a ľavých a pravých okrúhlych zátvoriek. Pri tvorbe logických výrazov používame nasledujúce logické operátory:

NOT	negácia	(napr. NOT A)
AND	logický súčin	(napr. A AND B)
OR	logický súčet	(napr. A OR B)

Význam jednotlivých logických operátorov charakterizuje nasledovná „pravdivostná" tabuľka (1 označuje pravdu, 0 nepravdu):

A	B	NOT A	A AND B	A OR B
Ø	Ø	1	Ø	Ø
Ø	1	1	Ø	1
1	Ø	Ø	Ø	1
1	1	Ø	1	1

Výsledkom vyhodnotenia logického výrazu je logická 1 (pravda), alebo logická 0 (nepravda). Priority operácií v logických výrazoch:

1. aritmetické operátory,
2. relačné operátory,

3. logické operátory v poradí
   NOT
   AND, OR

Príklad 1.12
Logické výrazy:

X<1 OR X>5	pravda, ak x<1, alebo x>5
X>3 AND X<9	pravda, ak 3<x<9
NOT X>=5	pravda, ak x<5

c) Reťazcové výrazy

Reťazcový výraz vytvárame z reťazcových konštánt, reťazcových premenných a reťazcových funkcií, oddelených operátorom + (alebo &), ktorý spája dva znakové reťazce.

Príklad 1.13
Reťazcové výrazy a ich hodnoty:

"ABC"+"DEF"	hodnota "ABCDEF"
"X"+"+"+"Y"	hodnota "X+Y"

ak A$="ABCD", B$="EFGH", potom
LEFT$(A$,2)+B$ má hodnotu "ABEFGH"

Reťazce možno porovnávať v reláciách. Reťazcové relačné operátory sa zapisujú rovnako ako aritmetické relačné operátory, majú však iný význam: porovnávajú reťazce podľa abecedného usporiadania (resp. podľa hodnôt ASCII kódu jednotlivých znakov).

Príklad 1.14
Reťazcové relácie:

A$=B$	reťazce A$ a B$ sú zhodné,
A$<B$	reťazec A$ abecedne predchádza reťazcu B$, napr. "NOVAK"<"NOVOTNY".

Pri porovnávaní dvoch reťazcov rôznej dĺžky sa kratší reťazec doplní sprava medzerami na dĺžku dlhšieho reťazca.

Reťazcové relácie môžu byť súčasťou logických výrazov.

1.3.6 PREHĽAD PRÍKAZOV JAZYKA BASIC G

Príkazy jazyka basic tvoria kľúčové slová (názvy príkazov) a parametre (spravidla výrazy oddelené interpunkčnými znamienkami).

V tabuľke 1.1 uvádzame kľúčové slová a stručné charakteristiky príkazov jazyka basic pre počítače PMD 85 a Didaktik Alfa (pre tento dialekt basicu sa často používa názov basic G). Všetky tieto príkazy možno používať v programovom režime, niektoré z nich možno používať v priamom („kalkulačkovom") režime. Príkazy, ktoré sú k dispozícii iba na PMD 85–2, označujeme v tabuľke hviezdičkou (*).

Tabuľka 1.1

1. Štandardné príkazy	
BEEP	zvukový signál
CLEAR	vynulovanie číselných premenných a nastavenie všetkých reťazcových premenných na prázdny znak
DATA	deklarácia zoznamu konštánt
DEF FNC	definícia používateľskej funkcie
DIM	deklarácia polí
DISP	výpis do dialógového riadka obrazovky
DLOAD	načítanie údajov z kazety magnetofónu
DSAVE	zápis údajov na kazetu magnetofónu
END	koniec programu
FOR TO STEP	začiatočný príkaz cyklu
GCLEAR	vymazanie obrazovky
GOSUB	príkaz skoku do podprogramu
GOTO	príkaz nepodmieneného skoku
IF THEN	príkaz podmieneného skoku
INPUT	vstup údajov z klávesnice
NEXT	koncový príkaz cyklu
ON GOSUB	prepínač podprogramov
ON GOTO	prepínač skokov
PAUSE	zastavenie vykonávania programu na zadaný časový interval
PRINT	výstup na obrazovku
READ	priradenie hodnôt zo zoznamu konštánt premenným
REM	poznámka
RESTORE	nastavenie ukazovateľa na začiatok alebo na zadaný riadok zoznamu konštánt

RETURN	návrat z podprogramu
STOP	prerušenie vykonávania programu
?	výpis údajov do dialógového riadka s prerušením spracovávania programu
DEG	prepnutie na stupne pri spracovaní goniometrických funkcií (*)
RAD	prepnutie na radiány pri spracovaní goniometrických funkcií (*)

2. Príkazy pre grafiku

AXES	vykreslenie súradnicových osí
BMOVE	nastavenie „pera“ pre kreslenie v bajtovom režime
BPLOT	vykreslenie motívu v bajtovom režime
FILL	vykreslenie plošných obrazcov v bodovom režime
LABEL	zväčšovanie znakov v bodovom režime
MOVE	nastavenie „pera“ pre kreslenie v bodovom režime
PLOT	vykreslenie bodu alebo úsečky v bodovom režime
SCALE	nastavenie mierky pre kreslenie v bodovom režime
PEN	nastavenie spôsobu kreslenia (*)

3. Špeciálne príkazy

CODE	vykonanie krátkeho programu v strojovom kóde
CONTROL	nastavenie portu pre výstup (napr. pri výstupe na tlačiareň)
OUT	odoslanie hodnoty na zadaný port
OUTPUT	výstup zadaným kanálom (napr. na tlačiareň)
POKE	zápis celočíselnej hodnoty z intervalu 0 až 255 do zadaného pamäťového miesta (jeden bajt)
ROM	načítanie obsahu ROM-pamäte
ON ERR	prepínač v prípade výskytu chyby (*)

Poznamenávame, že väčšina štandardných príkazov (DATA, DIM, END, FOR TO STEP, GOSUB, GOTO, IF THEN, INPUT, ON GOSUB, ON GOTO, NEXT, RETURN, PRINT, READ, REM, RESTORE, STOP) je spoločná pre všetky dialekty jazyka basic na rôznych počítačoch. Ostatné príkazy (grafika, spracovanie zvukov, práca s kazetovým magnetofónom, špeciálne príkazy na využívanie možností konkrétneho monitora alebo operačného systému) sa môžu na rôznych počítačoch odlišovať.

1.4 PROGRAMOVANIE VÝPOČTOV A PRÁCE S TEXTAMI

V tejto kapitole uvádzame stručný výklad štandardných príkazov jazyka basic G, ktoré nám umožnia programovať jednoduchšie výpočtové úlohy a úlohy spracovania textov. Pri väčšine príkladov uvedených v tejto kapitole nezáleží na konkrétnom dialekte jazyka basic. V prípadoch, ktoré sú závislé od dialektu, uvádzame spravidla riešenie pre počítače PMD 85 a Didaktik Alfa aj pre počítače Atari XL a XE.

1.4.1 ZAČIATOK A KONIEC PROGRAMU, KOMENTÁRE

Komentárový príkaz

```
REM poznámka
```

kde *poznámka* je text, ktorý obsahuje ľubovoľné znaky prípustné v jazyku basic G, slúži na vkladanie komentárov a vysvetľujúcich poznámok do programu. Pri vykonávaní programu sa tento príkaz ignoruje. Môžeme ho použiť kdekoľvek v programe, vo viacpríkazovom riadku však musí byť posledným príkazom riadka.

Program v basicu spravidla začíname komentárovým príkazom REM, v ktorom uvádzame názov programu, dátum jeho zostavenia a prípadné ďalšie informácie o programe a jeho autorovi.

Koncový príkaz

```
END
```

označuje koniec programu. Účinkom jeho vykonania program ukončí svoju činnosť. Nie je povinný; ak nie je uvedený, vykonávanie

programu sa ukončí po spracovaní programového riadka s najvyšším číslom.

Prerušovací príkaz

```
STOP
```

slúži na prerušenie vykonávania programu. Po jeho spracovaní sa v dialógovom riadku obrazovky vypíše správa

STOP AT LINE čr

kde *čr* je číslo riadka s príkazom STOP, ktorý spôsobil prerušenie. Činnosť programu možno v tomto prípade obnoviť od riadka, v ktorom došlo k prerušeniu, vložením povelu

```
CONT
```

Príkaz STOP je vhodné využívať pri ladení programu na zisťovanie logicky chybných úsekov programu.

Príklad 1.15

Nový program začneme vkladať od príkazu

1Ø REM PROGRAM 1.1 – PERCENTA, 1. 4. 88

v ktorom zapíšeme názov programu (PROGRAM 1.1 – PERCENTA) a dátum jeho zostavenia (napr. 1. 4. 88). Môžeme hneď vložiť aj koncový príkaz (s dostatočne vysokým číslom riadka, aby zostalo miesto na čísla ostatných programových riadkov), napr.

2ØØ END

Ak vložené programové riadky vypíšeme povelom LIST, vypíše sa na obrazovke

1Ø REM PROGRAM 1.1 – PERCENTA, 1. 4. 88
2ØØ END

1.4.2 VSTUPY A VÝSTUPY

Údaje, ktoré chceme spracovať, odovzdávame programu prostredníctvom klávesnice, výsledky a správy vypisuje program na obrazovku. Môžeme naprogramovať i zvukový výstup a časovo obmedzené pozastavenie vykonávania programu.

a) Vymazanie obrazovky príkazom GCLEAR

Príkaz

```
GCLEAR
```

vymaže obrazovku. Používa sa v priamom aj v programovom režime. Ak ho použijeme v priamom režime, napr. pred začiatkom vkladania nového programu, vymažeme z obrazovky zvyšky starého programu.

Príkaz GCLEAR je špeciálnym príkazom dialektu basicu pre PMD 85 a Didaktik Alfa. V iných dialektoch sa na vymazanie obrazovky spravidla používa príkaz CLS. V Atari basicu vymažeme obrazovku príkazom

PRINT CHR$(125);

b) Programovanie výstupov na obrazovku príkazom PRINT

Príkaz výstupu na obrazovku má tvar

```
PRINT zoznam
```

kde *zoznam* je zoznam údajov, ktoré chceme vypísať – číselných a reťazcových konštánt, premenných alebo celých výrazov oddelených čiarkami alebo bodkočiarkami. Ak je súčasťou zoznamu výraz, vypisuje sa jeho hodnota. Prvkami zoznamu príkazu PRINT môžu byť aj funkcie TAB a SPC.

Príkaz PRINT začína vypisovať do predposledného riadka obrazovky (25. riadok), pričom prv zobrazené riadky sa posunú o jeden riadok hore. Keď jednotlivé prvky zoznamu príkazu PRINT oddeľujeme bodkočiarkami, vypisujú sa jednotlivé im zodpovedajúce údaje bezprostredne vedľa seba. Keď prvky zoznamu oddeľujeme čiarkami, vypisujú sa takto: prvý údaj od prvej pozície riadka, druhý údaj od 15. pozície, tretí údaj od 29. pozície a štvrtý údaj od 43. pozície (prípadné ďalšie údaje do ďalšieho riadka). Ak je zoznam príkazu PRINT ukončený bodkočiarkou, potom bezprostredne nasledujúci príkaz PRINT zabezpečí výstup údajov do toho istého riadka za posledný v ňom zobrazený údaj. Vynechanie zoznamu v príkaze PRINT spôsobí výstup prázdneho riadka (t. j. vynechanie riadka).

Príklad 1.16

Nech A = 12.5, B = 0.01, C = –20200000, potom príkaz

```
100 PRINT A,B,C
```

vypíše do 25. riadka obrazovky

m12.5mmmmmmmmmm.01mmmmmmmmmm–2.02E+7m

(písmenom m označujeme medzery). Ak použijeme príkaz

```
100 PRINT A;B;C
```

(prvky zoznamu oddeľujeme bodkočiarkami), vypíšu sa príslušné údaje takto:

m12.5mm.Ø1m–2,Ø2E+ 7m

Všimnime si, že pri výstupe číselných údajov sa zobrazenie každého čísla začína medzerou alebo znamienkom mínus (v prípade záporného čísla) a končí medzerou. Ak je absolútna hodnota vystupujúceho čísla menšia ako 1, nezobrazuje sa nula pred desatinnou bodkou. Ak absolútna hodnota vystupujúceho čísla je z intervalu od 0.01 do 999999, potom sa toto číslo zobrazuje v obyčajnom desatinnom tvare (t. j. ako desatinné číslo s prípadným znamienkom a desatinnou bodkou). Číslo väčšie ako 999999 alebo menšie ako 0.01 sa zobrazuje v exponenciálnom tvare. Každé vystupujúce číslo sa zaokrúhli na 6 miest.

Príklad 1.17

Príkaz

8Ø PRINT "VLOZ CISLO X"

vypíše text uvedený v úvodzovkách. Príkaz

14Ø PRINT "P =";P;"%"

vypíše text v úvodzovkách, za ním hodnotu premennej P a znak %, teda napr.

P = 34.1583 %

Príkaz

5Ø PRINT

vynechá na obrazovke jeden riadok.

Na grafickú úpravu výstupov na obrazovku môžeme v príkaze PRINT využiť funkcie TAB a SPC. Funkcia TAB(n) zapísaná v zozname príkazu PRINT spôsobí, že pred vystupujúcou veličinou zapísanou v zozname za touto funkciou sa vynechá n pozícií od začiatku riadka. Funkcia SPC(n) spôsobí, že vystupujúca veličina zapísaná v zozname za touto funkciou sa začne vypisovať po vynechaní n medzier od konca predchádzajúcej vystupujúcej veličiny.

Príklad 1.18

Príkaz

1ØØ PRINT TAB(1Ø);X;SPC(5);Y

spôsobí, že v príslušnom riadku sa vynechá 10 medzier od začiatku,

vypíše sa hodnota premennej X, vynechá sa 5 medzier a vypíše sa hodnota Y.

Pri programovaní výstupov na obrazovku si treba uvedomiť, že v jednom riadku možno zobraziť maximálne 48 znakov (vrátane medzier).

Príkaz PRINT môžeme používať aj v priamom („kalkulačkovom") režime. Do jeho zoznamu pritom spravidla zapisujeme aritmetické výrazy, ktorých hodnota sa okamžite vypočíta a vypíše.

Príkaz PRINT je súčasťou všetkých dialektov basicu. Rozdiely v jeho používaní sa vyskytujú v súvislosti s funkciami TAB a SPC. Napr. Atari basic nepozná funkcie TAB a SPC. Grafickú úpravu výstupov na Atari regulujeme pomocou špeciálneho príkazu POSITION, určujúceho miesto na obrazovke, kde sa má vypisovať. Počítač PMD 85–2 poskytuje na grafickú úpravu výstupov príkazom PRINT navyše funkcie AT a INK (pozri poznámku 1.6).

c) Výstup do dialógového riadka príkazom DISP

Príkazom

```
DISP zoznam
```

kde pre *zoznam* vystupujúcich údajov platia rovnaké pravidlá ako pre zoznam príkazu PRINT, vypisujeme údaje uvedené v zozname do dialógového riadka (posledný riadok obrazovky). Vzhľadom na to, že výstup v tomto prípade „neroluje" zdola nahor po obrazovke, každý nový príkaz DISP spôsobí prepísanie doterajšieho obsahu dialógového riadka.

Podobný účinok ako príkaz DISP má príkaz

```
? zoznam
```

Na rozdiel od príkazu DISP však príkaz ? po vypísaní veličín zo zoznamu do dialógového riadka spôsobí pozastavenie vykonávania programu. Vykonávanie programu obnovíme (napr. po prečítaní si vypísanej správy) stlačením ľubovoľného klávesu.

Príkaz DISP je špeciálnym príkazom dialektu basicu pre počítače PMD 85 a Didaktik Alfa. V Atari basicu, ktorý nepozná dialógový riadok, sa nepoužíva.

d) Programovanie vstupu údajov príkazom INPUT
Príkaz

```
INPUT zoznam
```

kde *zoznam* obsahuje zoznam premenných, ktorým chceme priradiť hodnoty vkladané z klávesnice, používame na programovanie vstupu údajov počas vykonávania programu.

Pri spracovní príkazu INPUT sa vykonávanie programu preruší, ozve sa zvukový signál a počítač očakáva vloženie údajov. Údaje vkladáme z klávesnice do dialógového riadka obrazovky v poradí, ktoré zodpovedá zoznamu v príkaze INPUT. Vložené údaje odošleme do pamäte počítača stlačením EOL.

Príklad 1.19
Príkaz
90 INPUT X
požaduje vloženie jednej číselnej hodnoty, ktorú priradí premennej X. Príkaz
200 INPUT A,B,C
požaduje vloženie troch číselných hodnôt, ktoré sa priradia premenným A, B a C. Tieto hodnoty môžeme vložiť buď tak, že do dialógového riadka vložíme tri čísla oddelené čiarkami a stlačíme EOL, alebo postupne vložíme prvé číslo a stlačíme EOL, druhé číslo a stlačíme EOL a tretie číslo a stlačíme EOL.
Príkaz
300 INPUT T$
očakáva vloženie reťazca do reťazcovej premennej T$. Napíšeme napr.
BRATISLAVA
alebo
"KYSUCKA 12, 831 01 BRATISLAVA"
a údaj odošleme do pamäte stlačením EOL. Reťazec, ktorý neobsahuje čiarky a dvojbodky, môžeme vkladať bez úvodzoviek (prvý prípad); ak reťazec obsahuje uvedené oddeľovacie znamienka, musíme ho vložiť v úvodzovkách (druhý prípad).

Vždy pred použitím príkazu INPUT by sme mali použiť príkaz PRINT, DISP alebo ?, ktorým na obrazovku vypíšeme upozornenie, aké údaje sa majú vkladať.

Príklad 1.20
Dvojica príkazov
80 PRINT "VLOZ HODNOTU X"
90 INPUT X
vypíše najprv na obrazovku správu
VLOZ HODNOTU X
a potom zvukový signál upozorní, že počítač očakáva vstup z klávesnice. Po vložení hodnoty X a jej odoslaní stlačením EOL sa pokračuje vo vykonávaní ďalších príkazov.

Ak na začiatku práce s počítačom stlačíme klávesy SHIFT a PTL (na PMD 85–1), resp. SHIFT a PAL (na Didaktiku Alfa), zostanú vstupujúce údaje zobrazené na obrazovke aj po ich odoslaní do pamäte (z dialógového riadka, v ktorom boli vložené, sa presunú do predposledného riadka obrazovky a postupne „rolujú" po obrazovke). V opačnom prípade zobrazenie vložených údajov v dialógovom riadku zmizne z obrazovky po odoslaní údajov do pamäte stlačením EOL. Na počítači PMD 85–2 je to presne naopak: po spustení počítača sa automaticky nastaví režim priebežného zobrazovania a „rolovania" vstupujúcich údajov a stlačením SHIFT a PTL sa tento režim zruší.

Príkaz INPUT sa používa vo všetkých dialektoch basicu.

e) Programovanie zvukového signálu, príkaz BEEP

Príkaz

```
BEEP
```

slúži na spustenie zvukového signálu (tzv. „pípnutia"). Môžeme ho umiestniť kdekoľvek v programe, keď chceme upozorniť na nejakú dôležitú skutočnosť (napr. nesprávne vložený údaj, ukončenie výpočtu a pod.).

Príklad 1.21
Programový riadok
230 BEEP : BEEP
spôsobí dve po sebe idúce „pípnutia".

Atari basic príkaz BEEP nepozná, môžeme ho však nahradiť príkazom

PRINT CHR$(253)

f) Prerušenie programu príkazom PAUSE
Príkaz

```
PAUSE n
```

kde *n* je číselná premenná, konštanta alebo aritmetický výraz, slúži na pozastavenie vykonávania programu na vopred stanovený čas. Dĺžku prerušenia určuje hodnota *n*, ktorá môže byť z intervalu od 0 do 255, čo zodpovedá času prerušenia od 0 do 25.5 sekundy. V prípade, že parameter *n* v príkaze PAUSE chýba, preruší sa spracovanie programu na 25.5 sekundy. Vykonávanie programu môžeme obnoviť pred uplynutím zadaného času prerušenia stlačením klávesu SPACE.

Príklad 1.22
Úsek programu

```
500 DISP "PREZRI SI NAVOD (MAS NA TO 20 SEKUND)"
510 PAUSE 200
520 BEEP : BEEP
530 GCLEAR
```

má tento účinok: Vypíše do dialógového riadka správu
PREZRI SI NAVOD (MAS NA TO 20 SEKUND)
(predpokladáme, že na obrazovke sme predtým zobrazili nejaké informácie, potrebné pre ďalší postup), pozastaví vykonávanie programu na 20 sekúnd, upozorní na uplynutie času prerušenia dvojnásobným „pípnutím" a vymaže obrazovku.

Atari basic príkaz PAUSE nepozná.

1.4.3 PRIRAĎOVACÍ PRÍKAZ LET

Jednoduché výpočty a manipulácie s textami programujeme pomocou priraďovacieho príkazu

```
LET p = v
```

kde *p* je meno číselnej alebo reťazcovej premennej (jednoduchej alebo indexovanej) a *v* je aritmetický alebo reťazcový výraz, ktorého hodnota sa má priradiť premennej *p*.

Výraz na pravej strane priraďovacieho príkazu musí byť rovnakého typu ako premenná na ľavej strane (teda hodnotu aritmetického výrazu nemôžeme priradiť reťazcovej premennej a hodnotu reťazco-

vého výrazu číselnej premennej). Kľúčové slovo LET môžeme vynechať.

Príklad 1.23
Príkaz
110 LET P=(X/Z)*100
priradí premennej P hodnotu aritmetického výrazu
(X/Z)*100
Rovnaký účinok bude mať príkaz
110 P=(X/Z)*100

Príklad 1.24
Príkazy
300 A$="NESPRAVNE"
310 B$=RIGHT$(A$,7)
majú tento účinok: Reťazcovej premennej A$ sa priradí reťazec NESPRAVNE a pomocou funkcie RIGHT$ sa z tohto reťazca vyberie 7 znakov sprava, teda text SPRAVNE, ktorý sa priradí reťazcovej premennej B$.

Priraďovací príkaz LET je súčasťou všetkých dialektov basicu, uvedenie kľúčového slova LET v zápise príkazu je však povinné len v niektorých dialektoch.

Priraďovací príkaz možno využiť aj v priamom režime pri vykonávaní zložitejšieho výpočtu.

Príklad 1.25
Povrch rotačného valca s polomerom podstavy r=2,7 a výškou v=11,25 môžeme v priamom režime vypočítať s použitím vzorca $S=2\pi r(r+v)$ takto:

Vložíme najprv postupne priraďovacie príkazy
PI=3.14159
R=2.7
V=11.25
pričom každý z nich odošleme stlačením EOL. Potom vložíme príkaz
PRINT 2*PI*R*(R+V)
a po stlačení EOL sa na obrazovke zjaví výsledok a správa OK.

PROGRAM 1.1

Pristúpime k zostaveniu prvého uceleného programu, ktorý využíva príkazy REM, GCLEAR, PRINT, INPUT, LET a END. Program si vyžiada vloženie hodnôt Z a X a podľa vzorca

$$P = 100 \frac{X}{Z}$$

vypočíta, koľko percent zo základu Z je hodnota X. Výpis 1.1

```
10 REM PROGRAM 1.1 - PERCENTA, 1. 4. 88
20 GCLEAR
30 PRINT "X JE P PERCENT ZO ZAKLADU Z"
40 PRINT "ZADAVAME Z, X; POCITAME P"
50 PRINT
60 PRINT "VLOZ ZAKLAD Z"
70 INPUT Z
80 PRINT "VLOZ HODNOTU X"
90 INPUT X
100 PRINT
110 LET P=(X/Z)*100
120 PRINT "Z ="$Z
130 PRINT "X ="$X
140 PRINT "P ="$P$"%"
200 END
```

obsahuje výpis programu pre počítače PMD 85 a Didaktik Alfa, na obr. 1.2 uvádzame priebeh dialógu na obrazovke pri vkladaní hodnôt Z a X a zobrazení výsledku.

```
X JE P PERCENT ZO ZAKLADU Z
ZADAVAME Z, X; POCITAME P

VLOZ ZAKLAD Z
325
VLOZ HODNOTU X
111

Z = 325
X = 111
P = 34.1538 %

OK
```

1.2 Priebeh dialógu s programom PERCENTÁ

Aby program z výpisu 1.1 fungoval aj na počítačoch Atari XL a XE, stačí nahradiť programový riadok 20 s príkazom GCLEAR na vymazanie obrazovky riadkom

2Ø PRINT CHR$(125);

Pred odštartovaním programu povelom RUN na počítačoch PMD 85–1 a Didaktik Alfa stlačte klávesy SHIFT a PTL (resp. SHIFT a PAL), aby vstupujúce údaje zostali na obrazovke aj po ich odoslaní stlačením EOL.

František Turnovec

1.4.4 PRÍKAZY SKOKOV

Program v basicu sa normálne vykonáva v poradí čísel programových riadkov. Toto prirodzené poradie vykonávania jednotlivých príkazov možno zmeniť a rozhodovať sa v programe o ďalšom postupe príkazmi skokov GOTO, ON GOTO a IF THEN.

a) Príkaz nepodmieneného skoku GOTO
Príkaz

GOTO čr

kde *čr* je číslo programového riadka, ktorého príkazy sa majú vykonať ako bezprostredne nasledujúce, mení prirodzené poradie spracovania programových riadkov skokom na zadaný riadok. Pri použití príkazu GOTO vo viacpríkazovom riadku musí byť GOTO posledným príkazom viacpríkazového riadka.

Príklad 1.26
V úseku programu
```
50 PRINT "VLOZ NOVU HODNOTU"
60 INPUT A
```

```
100 GOTO 50
```
programový riadok
```
100 GOTO 50
```
spôsobí, že po jeho spracovaní sa vykoná príkaz z programového riadka s číslom 50, teda uskutoční sa skok nazad v programe.

Príkaz GOTO možno použiť aj v priamom režime. V tomto prípade spôsobí odštartovanie programu od zadaného riadka, podobne ako povel
```
RUN čr
```
Na rozdiel od príkazu RUN však zachová obsah premenných podľa posledného stavu spracovania programu (odštartovanie programu povelom RUN automaticky nuluje všetky premenné).

Príkaz GOTO je súčasťou všetkých dialektov basicu. Upozorňujeme, že dobrý programátor by mal tento príkaz používať čo najme-

nej, pretože časté nepodmienené skoky znižujú prehľadnosť programu.

b) Príkaz prepínača ON GOTO

Príkaz prepínača umožňuje viacnásobné vetvenie programu. Používa sa v tvare

ON v GOTO $čr_1$,$čr_2$,...,$čr_n$

kde v je aritmetický výraz, ktorý nadobúda hodnoty 1, 2, ..., n a $čr_j$ je číslo programového riadka, ktorý sa má spracovať ako bezprostredne nasledujúci, ak $v=j$. Zoznam čísel riadkov v príkaze ON GOTO môže obsahovať iba čísla existujúcich riadkov. V zozname sa môže viackrát vyskytovať rovnaké číslo riadka.

Ak hodnota výrazu v je menšia ako 1 alebo väčšia ako n, potom spracovanie programu po vykonaní príkazu ON GOTO pokračuje bezprostredne nasledujúcim programovým riadkom. Vo viacpríkazovom riadku musí byť príkaz ON GOTO posledným príkazom riadka.

Príklad 1.27

Nasledujúci úsek programu uvádza do súladu tvar slova BODY, uloženého v reťazcovej premennej B$, s použitým počtom bodov (napr. pri vyhodnocovaní nejakého dialógového testu):

```
1010 ON B GOTO 1030,1040,1040,1040
1020 B$="BODOV" : GOTO 1050
1030 B$="BOD" : GOTO 1050
1040 B$="BODY"
1050 PRINT "ZISKALI STE";B;B$
```

Predpokladajme, že premenná B obsahuje počet získaných bodov (nejaké celé nezáporné číslo). Podľa hodnoty premennej B sa program vetví takto:

Ak B=0, potom po spracovaní riadka 1010 sa prejde k bezprostredne nasledujúcemu riadku (hodnota výrazu v v príkaze ON GOTO je menšia ako 1), v ktorom sa premennej B$ priradí text BODOV a príkaz nepodmieneného skoku GOTO odovzdá riadenie riadku s číslom 1050, teda vypíše sa

ZISKALI STE 0 BODOV

Ak B=1, potom po spracovaní príkazu v riadku 1010 sa odovzdá

Vydavateľstvo Smena

riadenie riadku 1030, premennej B$ sa priradí text BOD a príkaz GOTO odovzdá riadenie riadku 1050, teda vypíše sa

ZISKALI STE 1 BOD

Ak B sa rovná 2, 3 alebo 4, príkaz v riadku 1010 odovzdá riadenie riadku 1040, teda premennej B$ sa priradí text BODY a riadok 1050 vypíše na obrazovku napr.

ZISKALI STE 2 BODY

Ak B>4, teda v príkaze ON GOTO nie je uvedené číslo riadka, zodpovedajúce tejto hodnote premennej B, odovzdá sa riadenie bezprostredne nasledujúcemu riadku (ako v prípade B=0), t. j. premennej B$ sa priradí text BODOV a príkaz GOTO odovzdá riadenie riadku 1050, ktorý vypíše na obrazovku napr.

ZISKALI STE 5 BODOV

Príklad ilustruje aj účelné využitie príkazu nepodmieneného skoku GOTO vo vylučujúcich sa vetvách programu.

Príkaz ON GOTO je súčasťou všetkých dialektov basicu.

c) Príkaz podmieneného skoku IF THEN

Príkaz podmieneného skoku mení prirodzené poradie spracovania programových riadkov v závislosti od splnenia určitej podmienky. Používa sa v tvare

```
IF pod THEN čr
```

alebo

```
IF pod THEN príkaz
```

kde *pod* je nejaká podmienka, zadaná v tvare aritmetickej alebo reťazcovej relácie, resp. logického výrazu, *čr* je číslo riadka, ktorý sa má spracovať ako bezprostredne nasledujúci v prípade, že podmienka je splnená, *príkaz* je príkaz, ktorý sa má vykonať v prípade, že podmienka je splnená.

Ak je podmienka v príkaze IF THEN splnená, potom sa vykoná skok na riadok s číslom *čr* (v prvom prípade), alebo sa vykoná príkaz uvedený za slovom THEN a prípadné ďalšie príkazy uvedené v tom istom riadku (v druhom prípade). Ak podmienka nie je splnená, spracovanie programu pokračuje bezprostredne nasledujúcim riadkom.

Príkaz IF THEN je súčasťou vštkých dialektov basicu.

Príklad 1.28

Príkaz

```
21Ø IF R=1 THEN 6Ø
```

spôsobí, že v prípade, keď hodnota premennej R je 1, odovzdá sa riadenie programovému riadku s číslom 60, zatiaľ čo pre iné hodnoty premennej R sa pokračuje vo vykonávaní príkazov bezprostredne nasledujúceho riadka. Rovnaký účinok bude mať príkaz

```
21Ø IF R=1 THEN GOTO 6Ø
```

Príklad 1.29

Príkazový riadok

```
22Ø IF R=Ø THEN PRINT "KONIEC VYPOCTOV" : GOTO
    25Ø
```

spôsobí, že pri hodnote R=0 sa v príslušnom riadku obrazovky vypíše text

```
KONIEC VYPOCTOV
```

a vykoná sa skok na programový riadok 250. Pre iné hodnoty premennej R sa príkazy PRINT a GOTO nevykonajú a pokračuje sa v spracovaní bezprostredne nasledujúceho riadka.

(Ako ste iste postrehli, programový riadok 220 je z typografických dôvodov rozdelený do dvoch riadkov. Podobných prípadov sa v tejto knihe vyskytuje viac, vy však nezabúdajte, že vždy ide o jediný súvislý riadok, ktorý musí byť vložený do počítača na jeden raz.)

PROGRAM 1.2

Pomocou príkazov GOTO, IF THEN a BEEP rozšírime program 1.1 tak, aby umožnil opakované vloženie údajov a výpočet (pozri výpis 1.2). Všimnime si, ako funguje záverečný úsek programu:

```
19Ø PRINT "CHCES POKRACOVAT? (1=ANO, Ø=NIE)"
2ØØ INPUT R
21Ø IF R=1 THEN 6Ø
22Ø IF R=Ø THEN PRINT "KONIEC VYPOCTOV" : GOTO
    25Ø
23Ø BEEP : BEEP
24Ø GOTO 19Ø
25Ø END
```

Ak na dotaz CHCEŠ POKRACOVAT? odpovieme 1 (áno), vráti sa program k riadku 60 a celý postup sa opakuje (vložíme nové údaje a dostaneme nový výsledok). Ak odpovieme 0 (nie), vypíše sa správa KONIEC VYPOCTOV a vykoná sa skok na programový riadok 250. Ak sa pomýlime a vložíme inú číselnú odpoveď ako 1 alebo 0, upozorní nás počítač na chybu dvojnásobným „pípnutím", vykoná sa skok na riadok 190 a očakáva sa vloženie správnej odpovede.

Aby program fungoval aj na Atari XL a XE, musíme riadok 20 nahradiť riadkom

```
2Ø PRINT CHR$(125);
```

a riadok 230 nahradiť riadkom

```
23Ø PRINT CHR$(253) : PRINT CHR$(253)
```

```
10 REM PROGRAM 1.2 - PERCENTA1, 1. 4. 88
20 GCLEAR
30 PRINT "X JE P PERCENT ZO ZAKLADU Z"
40 PRINT "ZADAVAME Z, X; POCITAME P"
50 PRINT
60 LET I=I+1 : PRINT
70 PRINT "PRIKLAD C."$I
80 PRINT
90 PRINT "VLOZ ZAKLAD Z"
100 INPUT Z
110 PRINT "VLOZ HODNOTU X"
120 INPUT X
130 PRINT
140 LET P=(X/Z)*100
150 PRINT "Z ="$Z
160 PRINT "X ="$X
170 PRINT "P ="$P$"%"
180 PRINT
190 PRINT "CHCES POKRACOVAT? (1=ANO, 0=NIE)"
200 INPUT R
210 IF R=1 THEN 60
220 IF R=0 THEN PRINT "KONIEC VYPOCTOV" : GOTO 250
230 BEEP : BEEP
240 GOTO 190
250 END
```

1.4.5 PROGRAMOVANIE CYKLOV

Cyklom nazývame úsek programu, ktorý sa pri vykonávaní programu môže viackrát opakovať. Cykly môžeme programovať buď pomocou skokových príkazov IF THEN a GOTO, alebo pomocou špeciálnej dvojice príkazov cyklu FOR TO STEP a NEXT.

a) Vytváranie cyklov pomocou skokových príkazov

Predpokladajme, že určitý úsek programu sa má vykonať viackrát po sebe. Pomocou príkazu GOTO sa môžeme po vykonaní posledného príkazu tohto úseku vrátiť na jeho začiatok. Príkaz podmieneného skoku IF THEN, vhodne umiestnený vnútri tohto úseku programu, nám pri splnení určitej podmienky umožní z tohto cyklu „vyskočiť".

Príklad 1.30

V niektorom mieste programu chceme vložiť n čísel a vypočítať ich súčet. Príslušný úsek programu môže vyzerať napr. takto:

```
70 I=0 : S=0
80 I=I+1 : IF I>N THEN 120
90 PRINT "VLOZ ";I;". CISLO: "; : INPUT A
100 S=S+A
110 GOTO 80
120 PRINT "SUCET = ";S
```

Opakujúci sa úsek programu (cyklus) v tomto prípade tvoria programové riadky 80 až 110. V riadku 80 sa hodnota premennej I, ktorá udáva, koľké číslo ideme vkladať, zväčší o jednotku. Ďalší príkaz tohto riadka sa „pýta", či už je hodnota premennej I väčšia ako počet sčítancov N (predpokladáme, že číslo N už bolo definované predtým v programe). Ak I je menšie alebo rovné N, vykonajú sa bezprostredne nasledujúce programové riadky, t. j. na obrazovke sa vypíše výzva na vloženie ďalšieho sčítanca, ktorý sa po vložení pripočíta k doterajšej hodnote súčtu v premennej S. Vykonaním príkazu nepodmieneného skoku v riadku 110 sa vrátime na začiatok cyklu do riadka 80. Ak sme už vložili N čísel a vypočítali ich súčet, je v riadku 80 v príkaze podmieneného skoku splnená podmienka I > N a riadenie sa odovzdá riadku 120, ktorý vypíše hodnotu súčtu. Tým sa ukončí vykonávanie cyklu.

Riadok 90 obsahuje príkaz PRINT s navádzacím textom na vloženie ďalšieho čísla. Zoznam tohto príkazu je ukončený bodkočiar-

kou, čo spôsobí, že po vykonaní bezprostredne nasledujúceho príkazu INPUT sa vložená hodnota zobrazí v tom istom riadku za navádzacím textom (ak sme v režime priebežného zobrazovania vstupov). Ak by zoznam príkazu PRINT nebol ukončený bodkočiarkou, zobrazila by sa vložená hodnota v nasledujúcom riadku obrazovky.

Všimnime si, že podobným spôsobom je konštruovaný program 1.2 (pozri výpis 1.2), kde cyklus zabezpečuje opakovanie výpočtu percent. V tomto prípade sme na vytvorenie cyklu použili dva príkazy podmieneného skoku IF THEN a priamy dotaz na pokračovanie alebo ukončenie.

b) Programovanie cyklu príkazmi FOR TO STEP a NEXT

Na programovanie cyklov možno použiť aj špeciálnu dvojicu príkazov cyklu: FOR TO STEP a NEXT. Cyklom v tomto prípade nazývame úsek programu, ktorý sa začína príkazom FOR TO STEP a končí sa príkazom NEXT.

Prvý príkaz cyklu má tvar

```
FOR p=v1 TO v2 STEP v3
```

kde p je tzv. *riadiaca premenná cyklu*, v_1 je aritmetický výraz, určujúci začiatočnú hodnotu riadiacej premennej p, v_2 je aritmetický výraz, ktorý určuje koncovú hodnotu riadiacej premennej p a v_3 je aritmetický výraz, ktorý určuje krok zmeny premennej p. Ak krok $v_3 = 1$, možno časť STEP v príkaze vynechať.

Za príkazom FOR TO STEP nasleduje tzv. *telo cyklu*, t. j. úsek programu, ktorý sa má viackrát vykonať. Posledný príkaz je

```
NEXT p
```

kde p je názov riadiacej premennej cyklu, použitej v začiatočnom príkaze FOR.

Ak krok v_3 v príkaze FOR TO STEP je kladný, potom cyklus funguje takto: Pri vstupe do cyklu (vykonanie príkazu FOR TO STEP) sa riadiacej premennej p priradí začiatočná hodnota v_1 a skontroluje sa, či je hodnota riadiacej premennej menšia alebo rovná koncovej hodnote v_2. Ak je táto podmienka splnená, vykonajú sa všetky príkazy, ktoré sa nachádzajú medzi príkazmi FOR TO STEP a NEXT. Po ich vykonaní sa hodnota riadiacej premennej

zväčší o krok v_3, znovu sa vykoná kontrola, či hodnota riadiacej premennej je menšia alebo rovná koncovej hodnote, a ak je táto podmienka splnená, vykonajú sa znova príkazy medzi FOR TO STEP a NEXT atď. Cyklus sa ukončí vtedy, keď nie je splnená podmienka, že hodnota riadiacej premennej je menšia alebo rovná koncovej hodnote. V tomto prípade program pokračuje vykonaním príkazu bezprostredne nasledujúceho za posledným príkazom cyklu NEXT. Ak krok v_3 v príkaze FOR TO STEP je záporný, potom sa uvedený postup mení iba v tom, že pred každým priechodom cyklu sa kontroluje, či je hodnota riadiacej premennej väčšia alebo rovná koncovej hodnote v_2. Cyklus sa ukončí vtedy, keď táto podmienka nie je splnená.

Príklad 1.31

Ukážeme, ako naprogramovať cyklus z príkladu 1.30 pomocou príkazov FOR TO STEP a NEXT. Príslušný úsek programu môže vyzerať napr. takto:

```
60 S=0
70 FOR I=1 TO N STEP 1
80 PRINT "VLOZ ";I;". CISLO: ": : INPUT A
90 S=S+A
100 NEXT I
110 PRINT "SUCET = ";S
```

Vidíme, že v tomto prípade netreba „strážiť" počet „otáčok" cyklu – príkazy FOR TO STEP a NEXT to robia automaticky.

Keďže krok zmeny riadiacej premennej je v našom prípade 1, v riadku 70 stačí uviesť iba

```
70 FOR I=1 TO N
```

teda časť STEP 1 možno vynechať.

Do cyklu vstupujeme iba prostredníctvom príkazu FOR TO STEP, skok dovnútra cyklu z inej časti programu je neprípustný. Z cyklu možno vystúpiť aj pred jeho ukončením (použitím niektorého skokového príkazu).

Príklad 1.32

Nasledujúci úsek programu je nesprávny:

```
100 FOR I=1 TO N
110 INPUT A
```

```
120 PRINT A
130 NEXT I
     .
     .
     .
200 GOTO 110
```

(v programovom riadku 200 požadujeme skok dovnútra cyklu).

Príklad 1.33

```
Cyklus
100 F=1
110 FOR I=1 TO N
120 F=F*I
130 IF F>999999 THEN 170
140 NEXT I
150 PRINT N;"! = ";F
160 GOTO 180
170 PRINT "NEVIEM VYPOCITAT PRESNU HODNOTU
    ";N;"!"
180 PRINT
```

slúži na výpočet faktoriálu

$$n! = 1 \times 2 \times \ldots \times (n-1) \times n$$

Keďže faktoriál je vždy celočíselný a najväčšie celé číslo, ktoré môžeme v basicu G presne zobraziť, je 999999, v riadku 130 kontrolujeme, či pre dané N je faktoriál ešte presne zobraziteľný. V prípade, že hodnota faktoriálu je väčšia ako 999999, vyskočíme z cyklu pred jeho ukončením.

Poznamenajme, že v Atari basicu je najväčším celým číslom, ktoré môžeme presne zobraziť, 999999999, riadok 130 preto upravíme takto:

```
130 IF F>999999999 THEN 170
```

(Doplnením úseku programu o vstup hodnoty N a jeho vykonaním na počítačoch PMD 85, resp. Didaktik Alfa alebo Atari XL, resp. XE sa presvedčte o tom, že na počítačoch PMD 85 a Didaktik Alfa môžeme presne vypočítať najviac 9! a na počítačoch Atari XL a XE najviac 12!)

Riadiaca premenná cyklu nemusí byť kladná ani celočíselná. To umožňuje programovať tabeláciu funkcií v ľubovoľnom „rozumnom" intervale a s ľubovoľným krokom.

Príklad 1.34

Hodnoty polynómu

$y = 2x^3 - 3x^2 + x + 2$

v intervale premennej x od –5 do 5 s krokom 0,5 tabeluje nasledujúci cyklus:

```
100 FOR X=-5 TO 5 STEP .5
110 Y=2*X^3-3*X^2+X+2
120 PRINT X,Y
130 NEXT X
```

Všimnime si, že v tomto prípade musí byť v príkaze FOR TO STEP uvedená časť STEP, pretože krok zmeny riadiacej premennej sa nerovná 1.

Rovnaký výsledok pri opačnom usporiadaní tabelovaných funkčných hodnôt získame takto:

```
200 FOR X=5 TO -5 STEP -.5
210 Y=2*X^3-3*X^2+X+2
220 PRINT X,Y
230 NEXT X
```

V tomto prípade je začiatočná hodnota riadiacej premennej cyklu väčšia ako koncová hodnota, a preto musíme použiť záporný krok.

PROGRAM 1.3

Program, ktorý uvádzame vo výpise 1.3, vypočíta súčet a aritmetický priemer n čísel. Program si vyžiada vloženie N a cyklom

```
7Ø FOR I=1 TO N
8Ø PRINT "UDAJ C.";I;": "; : INPUT A
9Ø S=S+A
1ØØ NEXT I
```

postupne požaduje vloženie N čísel, pričom každé z nich pripočíta k doteraz dosiahnutej hodnote súčtovej premennej S. Po skončení cyklu sa vypočíta priemer a na obrazovke sa vypíše počet sčítancov a hodnota súčtu a priemeru.

```
10 REM PROGRAM 1.3 - PRIEMERY, 2. 4. 88
20 REM SUCET A ARITMETICKY PRIEMER N CISEL
30 GCLEAR
40 PRINT "ZADAJ POCET UDAJOV N = "; : INPUT N
50 PRINT
60 S=0
70 FOR I=1 TO N
80 PRINT "UDAJ C."#I#": "; : INPUT A
90 S=S+A
100 NEXT I
110 P=S/N
120 PRINT
130 PRINT "POCET SCITANCOV       N = "#N
140 PRINT "SUCET                 S = "#S
150 PRINT "PRIEMER           P=S/N = "#P
160 END
```

Príkazy FOR TO STEP a NEXT sú súčasťou všetkých dialektov basicu.

1.4.6 DEKLARÁCIA POLÍ A PRÁCA S POĽAMI

V odseku 1.3.3 sme vysvetlili pojem poľa. Jedno, dvoj a trojrozmerné polia deklarujeme príkazom DIM. Pri práci s poľami je osobitne výhodné využívať cykly FOR TO STEP a NEXT.

a) Príkaz deklarácie polí DIM

Polia *deklarujeme* (t. j. prideľujeme im mená a rezervujeme miesto v pamäti) príkazom

DIM $p_1(r_1),p_2(r_1,r_2),p_3(r_1,r_2,r_3),\ldots$

kde p_1, p_2, p_3,... sú mená polí, r_1 je konštanta, ktorá udáva prvý rozmer poľa, r_2 je konštanta, ktorá udáva druhý rozmer poľa a r_3 je konštanta, ktorá udáva tretí rozmer poľa.

Typ poľa (číselný alebo reťazcový) sa určuje jeho menom. Mená číselných polí sú jedno alebo dvojznakové, pričom prvým znakom musí byť písmeno a druhým znakom písmeno alebo číslica. Mená reťazcových polí môžu byť až trojznakové, pričom prvým znakom musí byť písmeno, posledným znakom $ (dolár) a prípadným prostredným znakom môže byť písmeno alebo číslica.

V jednom príkaze DIM môžeme deklarovať viac polí rôznych rozmerov a typov. Každé pole smie byť deklarované iba raz. Deklaračné príkazy DIM môžu byť v programe na ľubovoľnom mieste pred prvým použitím príslušných polí. V prípade, že pole nedeklarujeme v príkaze DIM, pridelí mu interpret basicu pri prvom použití príslušných indexovaných premenných automaticky pre každý rozmer rozsah 11 (od 0 do 10). Pre lepšiu prehľadnosť programu odporúčame deklarovať všetky použité polia (teda aj polia s rozsahom jednotlivých rozmerov do 11 – deklarovať tieto je vhodné aj pre úsporu pamäte) v príkazoch DIM a deklaračné príkazy DIM sústrediť na začiatku programu.

Príklad 1.35

Úsek programu

```
2Ø DIM A(5Ø),B(1Ø,2Ø), P$(6Ø)
3Ø DIM Z1(2,1Ø,5)
```

deklaruje štyri polia: jednorozmerné číselné pole A, jednorozmerné reťazcové pole P$, dvojrozmerné číselné pole B a trojrozmerné číselné pole Z1.

Po odštartovaní programu príkazom RUN sa všetkým prvkom číselných polí priradia hodnoty 0 a všetkým prvkom reťazcových polí prázdne znaky (prázdny znak je reťazec "", ktorý obsahuje 0 znakov). Príkazom

```
CLEAR
```

môžeme v prípade potreby v programe vynulovať všetky premenné a polia.

V Atari basicu pracujeme iba s jednorozmernými alebo dvojrozmernými číselnými poľami. Pre číselné polia sa príkaz DIM používa rovnako. Vzhľadom na iný spôsob práce s reťazcami má príkaz DIM v Atari basicu pre reťazce iný význam ako v basicu G (pozri kapitolu 1.7).

b) Programovanie práce s poľami

Pomocou príkazov FOR TO STEP a NEXT môžeme s výhodou programovať vstupy údajov do jednotlivých polí a výstupy polí na obrazovku. Spôsob práce s poľami vysvetlíme na príkladoch.

Príklad 1.36

Nasledujúci úsek programu umožňuje vložiť prvky jednorozmerného číselného poľa:

```
20 DIM A(100)
    .
    .
    .
100 FOR I=1 TO N
110 PRINT "VLOZ A(";I;") ";
120 INPUT H : A(I)=H
130 NEXT I
```

```
100 FOR K=D TO H
```

začiatok vonkajšieho cyklu

```
200 FOR J=L TO P
```

telo vnútorného cyklu

```
300 NEXT J
```

pokračovanie vonkajšieho cyklu

```
400 NEXT K
```

1.3 Štruktúra vložených cyklov

Program si postupne vyžiada vloženie N prvkov číselného jednorozmerného poľa A (predpokladá sa, že hodnota N bola definovaná pred začiatkom cyklu a nie je väčšia ako 100, t. j. rozsah poľa deklarovaný v riadku 20). Programový riadok

```
120 INPUT H : A(I)=H
```

môžeme v basicu G zapísať aj takto:

```
120 INPUT A(I)
```

Uvedenú formu vstupu prvku poľa sme použili s ohľadom na to, aby bola prípustná aj v Atari basicu (Atari basic nepripúšťa indexované premenné v zozname príkazu INPUT).

Cyklus môže byť vložený v inom cykle, v tomto prípade hovoríme o tzv. *viacnásobných cykloch*. Vnútorný cyklus musí byť v tomto prípade ukončený skôr ako vonkajší cyklus, napr. podľa schémy z obr. 1.3. Riadiaca premenná vnútorného cyklu sa pritom musí líšiť od riadiacej premennej vonkajšieho cyklu.

Príklad 1.37

Ukážeme, ako s využitím viacnásobných cyklov možno naprogramovať vstup prvkov tabuľky (dvojrozmerného poľa) a vypísanie tabuľky na obrazovku.

```
20 DIM T(10,10)
.
.
.
100 REM VSTUP PRVKOV POLA T PO RIADKOCH
110 FOR I=0 TO 10
120 PRINT I;". RIADOK"
130 FOR J=0 TO 10
140 PRINT "VLOZ T(";I;",";J;") ";
150 INPUT H : T(I,J)=H
160 NEXT J
170 NEXT I
180 REM VYPIS POLA T NA OBRAZOVKU
190 PRINT
200 FOR I=0 TO 10
210 FOR J=0 TO 10
220 PRINT T(I,J);
230 NEXT J
240 PRINT
250 NEXT I
```

PROGRAM 1.4

Predpokladajme, že máme tabuľku s maximálnym rozsahom 26 riadkov a 16 stĺpcov. V programe ju deklarujeme ako dvojrozmerné pole A príkazom

DIM A(25,15)

(nezabúdajme, že prvky polí sa indexujú od nuly). Vložíme do tejto tabuľky číselné údaje. Chceme vypočítať priemery v jednotlivých riadkoch a stĺpcoch a celú tabuľku aj s vypočítanými priemermi zobraziť na obrazovke.

Program (pozri výpis 1.4) si vyžiada údaje o skutočnom počte riadkov a stĺpcov (tento skutočný počet nesmie byť väčší ako dekla-

```
10 REM PROGRAM 1.4 - TABULKA, 2. 4. 88
20 DIM A(25,15)
30 GCLEAR
31 REM V ATARI BASICU - PRINT CHR$(125);
100 REM VLOZENIE POCTU RIADKOV A STLPCOV
110 PRINT "VLOZTE POCET RIADKOV TABULKY (1 - 25): ";
120 INPUT M
130 PRINT "VLOZTE POCET STLPCOV TABULKY (1 - 15): ";
140 INPUT N
150 PRINT
200 REM VLOZENIE TABULKY PO RIADKOCH
210 PRINT "VLOZTE UDAJE TABULKY PO RIADKOCH"
220 PRINT
230 FOR I=1 TO M
240 PRINT "RIADOK C.";I
250 FOR J=1 TO N
260 PRINT "A(";I;",";J;") = ";
270 INPUT H : A(I,J)=H
280 NEXT J
290 NEXT I
300 REM VYPOCET RIADKOVYCH PRIEMEROV
310 FOR I=1 TO M
320 A(I,0)=0
330 FOR J=1 TO N
340 A(I,0)=A(I,0)+A(I,J)
350 NEXT J
360 A(I,0)=A(I,0)/N
370 NEXT I
400 REM VYPOCET STLPCOVYCH PRIEMEROV
410 FOR J=1 TO N
420 A(0,J)=0
430 FOR I=1 TO M
440 A(0,J)=A(0,J)+A(I,J)
450 NEXT I
460 A(0,J)=A(0,J)/M
470 NEXT J
500 REM ZOBRAZENIE TABULKY
510 GCLEAR
511 REM V ATARI BASICU - PRINT CHR$(125);
520 FOR I=1 TO M
530 FOR J=1 TO N
540 PRINT A(I,J);
550 NEXT J
560 PRINT "     ";A(I,0)
570 NEXT I
580 PRINT
590 FOR J=1 TO N
600 PRINT A(0,J);
610 NEXT J
620 END
```

rovaný rozsah poľa A), vyžiada si vloženie jednotlivých údajov po riadkoch, vypočíta riadkové a stĺpcové priemery a vypíše celú tabuľku spolu s riadkovými a stĺpcovými priemermi na obrazovku (riadkové priemery sú uložené v nultom prvku každého riadka, stĺpcové priemery sú uložené v nultom prvku každého stĺpca).

V riadkoch 31 a 511 v komentárových príkazoch REM uvádzame, ako treba zmeniť bezprostredne predchádzajúci príkaz, aby program fungoval aj na počítačoch Atari XL a XE.

1.4.7 DEKLARÁCIA A ČÍTANIE ZOZNAMU KONŠTÁNT

Údaje možno uchovávať priamo v programe v tzv. zoznamoch konštánt deklarovaných príkazmi DATA. Hodnoty konštánt z týchto zoznamov možno priraďovať premenným príkazmi READ. Miesto v zozname, od ktorého sa čítajú konštanty príkazom READ, možno nastaviť príkazom RESTORE.

a) Deklarácia zoznamu konštánt príkazmi DATA

Príkaz DATA slúži na vkladanie zoznamu konštánt do programu. Používa sa v tvare

```
DATA zoznam
```

kde *zoznam* je zoznam číselných a reťazcových konštánt, oddelených čiarkami.

Príkazy DATA môžu byť umiestnené kdekoľvek v programe, pre väčšiu prehľadnosť ich však odporúčame sústrediť na konci programu za príkazom END.

Príklad 1.38

Nasledujúci úsek programu definuje zoznam konštánt:

```
1010 DATA 15,12
1020 DATA ALENA, 1,2,1,1,1,1,3,2,1,2,2,1
1030 DATA "PRIEZVISKO, MENO, DATUM
     NARODENIA","C,T,"
```

Všimnime si, že v tých istých príkazoch DATA môžeme zapisovať číselné aj reťazcové konštanty. Pri zápise reťazcových konštánt v príkazoch DATA je nutné uvádzať úvodzovky iba vtedy, ak tieto konštanty obsahujú začiatočné alebo koncové medzery, čiarky alebo dvojbodky.

b) Čítanie zoznamu konštánt príkazmi READ

Hodnoty konštánt zo zoznamu deklarovaného príkazmi DATA priraďujeme premenným príkazmi READ. Príkaz READ má tvar

```
READ zoznam
```

kde *zoznam* je zoznam premenných, ktorým sa majú priradiť hodnoty zo zoznamu konštánt. V zozname príkazu READ sa môžu vyskytovať číselné a reťazcové premenné, oddelené čiarkami.

Príkazy READ spolupracujú s príkazmi DATA takto:

Konštanty zo všetkých príkazov DATA v danom programe tvoria akoby jeden súvislý zoznam v tom poradí, v akom sú v programe uložené príkazy DATA. Príkazy READ postupne priraďujú premenným hodnoty konštánt z tohto zoznamu v tom poradí, v akom sú vykonávané (teda prvý príkaz READ začne čítať konštanty zo začiatku zoznamu, druhý príkaz READ číta konštanty zo zoznamu od miesta, v ktorom skončil čítanie prvý príkaz READ, atď.). Vzájomne si zodpovedajúce premenné zo zoznamu príkazov READ a konštanty zo zoznamu definovaného príkazmi DATA musia byť rovnakého typu (teda číselné konštanty priraďujeme číselným premenným a reťazcové konštanty reťazcovým premenným).

Príklad 1.39

Výsledkom úseku programu

```
100 READ M,N
      .
      .
      .
1000 REM ZACIATOK ZOZNAMU KONSTANT
1010 DATA 15,12
```

v ktorom príkaz v riadku 100 je prvým vykonávaným príkazom READ a príkaz v riadku 1010 je prvým príkazom DATA, je nasledujúce priradenie konštánt premenným:

M = 15, N = 12

Výsledkom úseku programu

```
300 READ A,B,C
310 READ T1$,T2$
      .
      .
      .
```

```
500 DATA 10,140.5,120
510 DATA "PRIEZVISKO, MENO, DATUM
    NARODENIA","C.T. "
```

v ktorom príkaz v riadku 300 je prvým vykonávaným príkazom READ a príkaz v riadku 500 je prvým príkazom DATA, je priradenie

```
A = 10, B = 140.5, C = 120
T1$ = "PRIEZVISKO, MENO, DATUM NARODENIA"
T2$ = "C.T. "
```

Všimnime si, že príkazy DATA nemusia byť v programe umiestnené pred príkazmi READ. Príkazy DATA slúžia iba na uloženie konštánt do pamäte a nevykonávajú v programe žiadnu operáciu, nezáleží preto na ich umiestnení v programe, iba na ich poradí. Rovnaký výsledok ako vo vyššie uvedenom úseku programu by sme dostali napr. aj takto:

```
10 DATA 10,140.5,120
20 DATA "PRIEZVISKO, MENO, DATUM
   NARODENIA","C.T. "
30 READ A,B,C,T1$,T2$
```

c) Nastavenie ukazovateľa v zozname konštánt príkazom RESTORE

Príkaz RESTORE umožňuje opakované čítanie konštánt zo zoznamu definovaného príkazmi DATA alebo zmenu poradia čítania. Pri vykonávaní programu si počítač „pamätá" miesto v zozname konštánt, od ktorého by mal čítať nasledujúci príkaz READ (pre jednoduchosť hovoríme, že posúva ukazovateľ v zozname konštánt). Príkazom

```
RESTORE čr
```

si môžeme tento ukazovateľ nastaviť na miesto zoznamu, ktoré zodpovedá prvej konštante uvedenej v príkaze DATA v programovom riadku *čr*. Znamená to, že bezprostredne nasledujúci príkaz READ bude čítať zo zoznamu od tohto miesta. Ak príkaz použijeme v tvare

```
RESTORE
```

t. j. bez čísla riadku, nastaví sa ukazovateľ na začiatok zoznamu,

t. j. bezprostredne nasledujúci príkaz READ bude čítať konštanty od prvej konštanty prvého príkazu DATA. Po odštartovaní programu sa ukazovateľ automaticky nastaví na začiatok zoznamu.

Príklad 1.40

Výsledkom vykonania úseku programu

```
20 READ A,B,C
30 RESTORE
40 READ X,Y,D,E,F
50 DATA 28,62,63,60,52
```

bude priradenie

A = 28, B = 62, C = 63

X = 28, Y = 62, D = 63, E = 60, F = 52

Po vykonaní riadku 20 (priradenie hodnôt zo začiatku zoznamu premenným A, B, C) sa účinkom príkazu RESTORE nastaví ukazovateľ znovu na začiatok zoznamu a ďalší príkaz READ bude čítať od začiatku.

Príklad 1.41

Výsledkom vykonania úseku programu

```
100 RESTORE 500
110 READ X
120 RESTORE 510
130 READ Y,Z
140 RESTORE 500
150 READ A,B,C
      .
      .
      .
500 DATA 28,30,15
510 DATA 11,36
```

bude priradenie

X = 28

Y = 11, Z = 36

A = 28, B = 30, C = 15

Príkazy DATA, READ a RESTORE sú súčasťou všetkých dialektov basicu.

PROGRAM 1.5

Program 1.4 pozmeníme tak, aby slúžil na spracovanie prehľadov o klasifikácii žiakov. Spracovávanými údajmi budú mená žiakov, názvy predmetov a známky žiakov z jednotlivých predmetov. V jednorozmernom reťazcovom poli Z$ uchovávame mená žiakov, v jednorozmernom reťazcovom poli P$ uchovávame názvy predmetov, v dvojrozmernom číselnom poli A uchovávame známky. Údaje (mená žiakov, názvy predmetov a známky) nevkladáme príkazom INPUT, ale zapisujeme ich do programu pomocou príkazov DATA. Všetky príkazy DATA uložíme na konci programu od riadka 1000. Prvý príkaz DATA obsahuje údaje o počte žiakov a predmetov (v našom prípade 15 žiakov a 12 predmetov). Ďalšie príkazy DATA obsahujú mená žiakov a ich známky. Posledná skupina príkazov DATA obsahuje názvy predmetov. Poradie známok musí zodpovedať poradiu predmetov.

Program (pozri Výpis 1.5) najprv načíta údaje zo zoznamu defino-

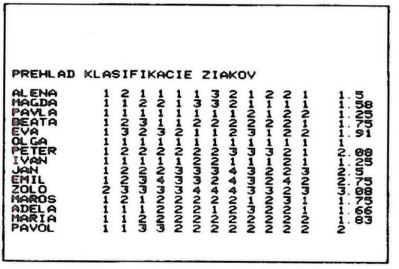
```
10 REM PROGRAM 1.5 - PROSPECH, 3. 4. 88
20 DIM A(25,15),Z$(25),P$(15)
30 @CLEAR : DISP "PREBIEHA SPRACOVANIE UDAJOV"
100 REM PRECITANIE UDAJOV ZO ZOZNAMU DATA
110 READ M,N
120 FOR I=1 TO M : READ Z$(I)
130 FOR J=1 TO N : READ H : A(I,J)=H : NEXT J
140 NEXT I
150 FOR J=1 TO N : READ P$(J) : NEXT J
200 REM VYPOCET RIADKOVYCH PRIEMEROV
210 FOR I=1 TO M : A(I,J)=0
220 FOR J=1 TO N : A(I,0)=A(I,0)+A(I,J) : NEXT J
230 A(I,0)=A(I,0)/N : NEXT I
300 REM VYPOCET STLPCOVYCH PRIEMEROV
310 FOR J=1 TO N : A(0,J)=0
320 FOR I=1 TO M : A(0,J)=A(0,J)+A(I,J) : NEXT I
330 A(0,J)=A(0,J)/M : NEXT J
400 REM ZOBRAZENIE PREHLADU PODLA ZIAKOV
410 @CLEAR
420 PRINT "PREHLAD KLASIFIKACIE ZIAKOV" : PRINT
430 FOR I=1 TO M
440 PRINT LEFT$(Z$(I),10);TAB(12);
450 FOR J=1 TO N
460 PRINT STR$(A(I,J));
470 NEXT J
480 R$=STR$(A(I,0))
490 PRINT TAB(2*N+14);LEFT$(R$,5) : NEXT I
500 REM ZOBRAZENIE PREHLADU PODLA PREDMETOV
510 DISP "PRECHOD K PREHLADU PREDMETOV - STLAC EOL!"
520 INPUT Q$
530 @CLEAR
540 PRINT "PREHLAD KLASIFIKACIE PODLA PREDMETOV"
550 PRINT
560 FOR J=1 TO N
570 PRINT LEFT$(P$(J),20);TAB(23);
580 R$=STR$(A(0,J))
590 PRINT LEFT$(R$,5) : NEXT J
600 END
1000 REM UDAJE O ZIAKOCH, ZNAMKACH A PREDMETOCH
1010 DATA 15,12
1020 DATA ALENA,1,2,1,1,1,1,3,2,1,2,2,1
```

```
1030 DATA MAGDA,1,1,2,2,1,3,3,2,1,1,1,1
1040 DATA PAVLA,1,1,1,1,1,1,1,1,2,1,2,2
1050 DATA BEATA,1,2,3,1,1,2,2,2,2,2,2,1
1060 DATA EVA,  1,3,2,3,2,1,1,2,3,1,2,2
1070 DATA OLGA, 1,1,1,1,1,1,1,1,1,1,1,1
1080 DATA PETER,1,2,2,2,2,2,2,3,3,3,2,1
1090 DATA IVAN, 1,1,1,1,1,2,2,1,1,1,2,1
1100 DATA JAN,  1,2,2,2,3,3,3,4,3,2,2,3
1110 DATA EMIL, 1,2,3,4,3,3,2,4,3,2,4,2
1120 DATA ZOLO, 2,3,3,3,3,4,4,4,3,3,2,3
1130 DATA MAROS,1,2,1,2,2,2,2,2,1,2,3,1
1140 DATA ADELA,1,1,1,2,2,2,1,2,3,2,2,1
1150 DATA MARIA,1,1,2,2,2,2,2,2,2,2,2,2
1160 DATA PAVOL,1,1,3,3,2,2,2,2,2,2,2,2
1170 DATA "SPRAVANIE"
1180 DATA "SLOVENSKY JAZYK"
1190 DATA "RUSKY JAZYK"
1200 DATA "OBCIANSKA VYCHOVA"
1210 DATA "DEJEPIS"
1220 DATA "ZEMEPIS"
1230 DATA "MATEMATIKA"
1240 DATA "PRIRODOPIS"
1250 DATA "TELESNA VYCHOVA"
1260 DATA "VYTVARNA VYCHOVA"
1270 DATA "HUDOBNA VYCHOVA"
1280 DATA "PRACOVNE VYUCOVANIE"
```

vaného príkazmi DATA do polí Z$, A a P$ a vyráta priemery podľa jednotlivých žiakov a predmetov. V dialógovom riadku sa pritom zjaví správa

PREBIEHA SPRACOVANIE UDAJOV

Po skončení výpočtu sa na obrazovke zobrazí tabuľka s menami žiakov a ich priemermi (obr. 1.4). V dialógovom riadku sa vypíše správa



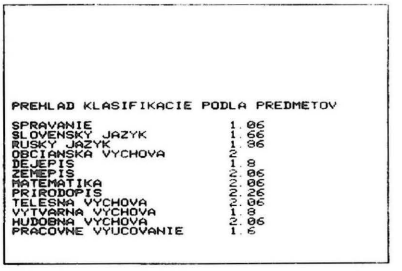
1.4 Výstup programu PROSPECH. Prehľad klasifikácie žiakov

PRECHOD K PREHLADU PREDMETOV – STLAC EOL

a po stlačení klávesu EOL sa zobrazia názvy predmetov a celkové priemery triedy z týchto predmetov (obr. 1.5).

Aby program mal širšie uplatnenie (napr. na vyhodnocovanie prospechu v rôznych polrokoch), stačí pre každé nové použitie zme-

niť údaje v príkazoch DATA pomocou prostriedkov na editovanie programov (pozri odsek 1.2.3).



1.5 Výstup programu PROSPECH. Prehľad klasifikácie podľa predmetov

V programe sme použili štandardné funkcie TAB, STR$ a LEFT$ na grafickú úpravu výstupu (tabuľky) a zaokrúhľovanie vypočítaných priemerov na 2 desatinné miesta. Napr. v príkaze

44Ø PRINT LEFT$(Z$(I),1Ø);TAB(12);

nám funkcia LEFT$ zabezpečí, že sa vypíše iba 10 znakov z každého mena a funkcia TAB nastaví výpis prvej známky na 12. pozíciu príslušného riadka. V príkaze

48Ø R$=STR$(A(I,Ø))

funkcia STR$ prevedie priemer, uložený v číselnej indexovanej premennej A(I,Ø), na reťazec číselných znakov. Príkazom

49Ø PRINT TAB(2*N+14);LEFT$(R$,5)

potom funkciou TAB nastavíme výpis riadkových priemerov na pozíciu 2*N+14 príslušného riadka (pre N = 12 na 38. pozíciu) a funkciou LEFT$ zabezpečíme vytlačenie priemeru iba s 2 desatinnými miestami.

Vzhľadom na odlišný spôsob práce s reťazcami nie je program 1.5 priamo použiteľný na počítačoch Atari XL a XE.

1.4.8 PRÍKAZY FUNKCIÍ A PODPROGRAMOV

V jazyku basic máme k dispozícii celý rad tzv. štandardných funkcií (aritmetické funkcie, funkcie na prácu s reťazcami, špeciálne a pomocné funkcie), ktoré nám zjednodušujú programovanie niektorých výpočtov a činností (prehľad funkcií pozri v odseku 1.3.4). Tieto funkcie sa používajú zápisom ich mena a argumentov v príslušnom mieste programu. Okrem toho si používateľ môže definovať

vlastné funkcie (tzv. používateľské funkcie) príkazom DEF FNC. Pomocou príkazov GOSUB, ON GOSUB a RETURN možno v basicu programovať a využívať tzv. podprogramy, t. j. úseky programov, na ktoré sa možno opakovane odvolávať v rôznych miestach programu.

a) Využívanie štandardných funkcií

Spôsob používania štandardných funkcií ilustrujeme na niekoľkých príkladoch.

Príklad 1.42

Predpokladajme, že v jednorozmernom reťazcovom poli A sme uložili n čísel. Chceme určiť, koľko z nich je záporných, koľko kladných a koľko rovných nule. S použitím funkcie SGN môžeme postupovať napr. takto:

```
20 DIM A(100)

90 S1=0 : S2=0 : S3=0
100 FOR I=1 TO N
110 IF SGN(A(I))=1 THEN S1=S1+1 : GOTO 140
120 IF SGN(A(I))=0 THEN S2=S2+1 : GOTO 140
130 IF SGN(A(I))=-1 THEN S3=S3+1
140 NEXT I
```

Po vykonaní tohto úseku programu bude v premennej S1 uložený počet kladných čísel, v premennej S2 počet núl a v premennej S3 počet záporných čísel.

Pomocou funkcie RND môžeme v programe vytvárať náhodné čísla z intervalu ⟨0, 1). Vhodnou úpravou generovaných čísel môžeme vytvárať čísla s požadovanými vlastnosťami.

Príklad 1.43

Nasledujúci úsek programu vytvorí 100 celých čísel z intervalu od 1 do 100 a uloží ich do jednorozmerného poľa B.

```
20 DIM B(100)
```

```
100 FOR I=1 TO 100
110 B(I)=INT(100*RND(1)+1)
120 NEXT I
```

Poznamenajme, že kladná konštanta dosadená za argument funkcie RND spôsobí, že pri každom privolaní funkcie RND sa vytvorí nové náhodné číslo. Aritmetický výraz

```
100*RND(1)+1
```

transformuje číslo z intervalu (0, 1) na číslo z intervalu od 1 do 100. Jeho dosadením za argument funkcie INT zabezpečíme, že sa budú vytvárať celé čísla.

Užitočné služby nám poskytujú reťazcové funkcie. Pomocou nich môžeme z reťazcov vyberať ich časti (tzv. podreťazce) a analyzovať ich. Ukážeme napr., ako použiť reťazcové funkcie na hľadanie v reťazcoch.

Príklad 1.44

Predpokladajme, že zo zoznamu uloženého v reťazcovom poli T$ chceme vytlačiť iba tie jeho položky (napr. mená), ktoré sa začínajú písmenom A. S využitím reťazcovej funkcie LEFT$ môžeme postupovať takto:

```
20 DIM T$(100)
    .
    .
    .
100 FOR I=1 TO 100
110 IF LEFT$(T$(I),1)<>"A" THEN 130
120 PRINT T$(I)
130 NEXT I
```

Príklad 1.45

Chceme zistiť, či reťazec R$ obsahuje slovo PRAHA. Použijeme na to funkcie LEN a MID$:

```
200 IF LEN(R$)<5 THEN 250
210 FOR I=1 TO LEN(R$)-4
220 R1$=MID$(R$,I,5)
230 IF R1$="PRAHA" THEN 260
240 NEXT I
250 K=0 : GOTO 270
260 K=1
```

V riadku 200 pomocou funkcie LEN zistíme, koľko má reťazec znakov. Ak má menej ako 5 znakov, nemôže obsahovať slovo PRAHA a uskutoční sa skok na riadok 250. V opačnom prípade vyberáme funkciou MID$ postupne zľava doprava podreťazce reťazca R$, ktoré obsahujú 5 znakov, a porovnávame ich s reťazcom "PRAHA". V prípade, že existuje podreťazec s požadovaným obsahom, vykoná sa skok na riadok 260 a premennej K sa priradí hodnota 1. V opačnom prípade sa v riadku 250 priradí premennej K hodnota 0 a vykoná sa skok na riadok 270 v ďalšej časti programu. Podľa hodnoty K sa potom rozhodujeme o ďalšom spracovaní reťazca.

Na zistenie rozsahu pamäte, ktorú máme k dispozícii na vkladanie programov a údajov, používame funkciu FRE. Táto funkcia sa spravidla používa v priamom režime.

Príklad 1.46

Ak chceme v priebehu vytvárania programu zistiť, koľko máme ešte k dispozícii voľnej pamäte na vkladanie ďalších príkazov a údajov, vložíme príkaz

```
PRINT FRE(X)
```

a stlačíme EOL. V dialógovom riadku obrazovky sa vypíše rozsah voľnej pamäte v kilobajtoch. Príkazom

```
PRINT FRE(X$)
```

a stlačením EOL zistíme voľnú pamäťovú oblasť na vkladanie reťazcov.

Na počítači PMD 85–2 môžeme funkciu VAL použiť na vyhodnocovanie aritmetických výrazov (na PMD 85–1 a Didaktiku Alfa táto funkcia iba transformuje reťazec číslic na číslo).

Príklad 1.47

Na počítači PMD 85–2 má úsek programu

```
100 PRINT "ZADAJ FUNKCIU PREMENNEJ X: "
110 INPUT F$
120 PRINT "ZADAJ HODNOTU PREMENNEJ X: "
130 INPUT X
140 Y=VAL(F$)
```

tento účinok: Pre správne zapísaný aritmetický výraz vložený pri vykonávaní programu do reťazcovej premennej F$ a hodnotu X vypočíta hodnotu príslušného výrazu a priradí ju premennej Y.

Poznámka 1.7

V Atari basicu chýbajú niektoré štandardné funkcie, ktoré máme k dispozícii v basicu G. Sú to predovšetkým reťazcové funkcie RIGHT$, LEFT$ a MID$ (čo vyplýva z iného spôsobu spracovania reťazcov) a ďalej funkcie TAB a SPC na formátovanie výstupov.

b) Používateľské funkcie

Popri bohatej ponuke štandardných funkcií si môžeme definovať aj vlastné funkcie. Na tieto účely používame príkaz

```
DEF FNCz(p)=v
```

kde DEF je kľúčové slovo, FNC je nemenná časť mena funkcie, *z* je jeden abecedný znak A až Z, ktorý si volí programátor a pomocou ktorého zadáva úplný názov svojej funkcie, *p* je jeden argument v tvare jednoduchej číselnej premennej a *v* je aritmetický výraz.

Takto definovanú funkciu potom použijeme rovnako ako štandardné funkcie, t. j. zápisom mena funkcie a dosadením konkrétnej hodnoty za argument funkcie. Funkciu treba definovať pred jej prvým použitím.

Príklad 1.48

Príkaz

```
3Ø DEF FNCA(X)=INT(X*RND(1)+1)
```

definuje funkciu, ktorá vytvára celé kladné číslo z intervalu od 1 do X. Vidíme, že v definícii používateľskej funkcie možno využiť aj štandardnú funkciu (v našom prípade RND). Nami definovanú funkciu FNCA použijeme napr. takto:

```
1ØØ FOR I=1 TO 1ØØ
11Ø B(I)=FNCA(1ØØ)
12Ø NEXT I
```

(pozri tiež príklad 1.43).

Príklad 1.49

Podľa vzorca

$$\operatorname{cotg} x = \frac{\cos x}{\sin x}$$

si môžeme definovať vlastnú funkciu pre kotangens

```
2Ø DEF FNCC(X)=COS(X)/SIN(X)
```

PROGRAM 1.6

Ukážeme, ako určiť deň v týždni pre ľubovoľný dátum 20. storočia. Nech r je rok, m je mesiac a d je deň nášho dátumu. Jednotlivé dni v týždni označíme poradovými číslami: 1 – pondelok, 2 – utorok, 3 – streda atď. Nech m_j je počet dní v j-tom mesiaci (teda napr. $m_1 = 31$ je počet dní v januári, $m_2 = 28$ alebo 29 je počet dní vo februári podľa toho, či r je, alebo nie je priestupný rok atď). Označme p počet dní, ktoré uplynuli k nami zadanému dátumu (vrátane dňa, ktorého dátum sme zadali) od začiatku 20. storočia. Určíme ho podľa vzorca

$$p = 365(r-1901) + \text{int}((r-1901)/4) + k + d$$

kde int(x) označuje celú časť čísla x a k je počet dní v prvých $m-1$ mesiacoch daného roku. Poradové číslo dňa v týždni potom určíme podľa vzorca

$$p - 7.\text{int}(p/7) + 1$$

Vzorec vychádza z toho, že 1. 1. 1901 bol utorok.

```
10 REM PROGRAM 1.6 - DATUM, 5. 4. 88
20 DIM D$(7),MD(12)
30 DEF FNCA(R)=365*(R-1901)+INT((R-1901)/4)
40 DEF FNCB(P)=P-7*INT(P/7)+1
50 FOR I=1 TO 12 : READ MD(I) : NEXT I
60 FOR I=1 TO 7 : READ D$(I) : NEXT I
100 GCLEAR
110 PRINT "VLOZTE DATUM"
120 PRINT "ROK (1901-2000): "; : INPUT R
130 PRINT "MESIAC (1-12):   "; : INPUT M
140 PRINT "DEN (1-31):      "; : INPUT D
150 IF R/4-INT(R/4)=0 THEN MD(2)=29
160 DR=0
170 IF M-1=0 THEN 210
180 FOR I=1 TO M-1 : DR=DR+MD(I) : NEXT I
210 DS=FNCA(R)+DR+D
220 K=FNCB(DS)
230 PRINT
240 PRINT D;".";M;".";R;"  - ";D$(K)
250 DISP "CHCETE POKRACOVAT? (A, N):"
260 INPUT Q0$ : M(2)=28 : IF Q0$="A" THEN 100
270 PRINT : PRINT "KONIEC PROGRAMU"
280 END
300 DATA 31,28,31,30,31,30,31,31,30,31,30,31
310 DATA PONDELOK,UTOROK,STREDA,STVRTOK
320 DATA PIATOK,SOBOTA,NEDELA
```

Program pre výpočet dňa v týždni uvádzame vo výpise 1.6. Využívame v ňom používateľské funkcie

3Ø DEF FNCA(R)=365*(R-19Ø1)+INT((R-19Ø1)/4)

na určenie počtu dní, ktorý uplynul od začiatku 20. storočia do začiatku roka R, a

4Ø DEF FNCB(P)=P-7*INT(P/7)+1

na určenie poradového dňa v týždni, ak k danému dátumu uplynulo od začiatku storočia P dní.

Atari basic neposkytuje možnosť definovať používateľské funkcie. Vzhľadom na to a na odlišnosť práce s reťazcami nie je program 1.6 použiteľný na počítačoch Atari XL a XE.

c) Podprogramy (príkazy GOSUB, RETURN a ON GOSUB)

Podprogram v jazyku basic je skupina programových riadkov, ktorá sa končí príkazom

```
RETURN
```

Podprogram privolávame z ľubovoľného miesta programu príkazom

```
GOSUB čr
```

kde *čr* je číslo riadka s prvým príkazom podprogramu, ktorý chceme privolať.

Po vykonaní príkazu GOSUB sa odovzdá riadenie prvému riadku podprogramu. Len čo sa pri vykonávaní podprogramu vykoná príkaz RETURN, riadenie sa odovzdá príkazu, ktorý bezprostredne nasleduje za príkazom GOSUB, z ktorého sme podprogram privolali.

Program v basicu môže obsahovať väčší počet podprogramov. Všetky podprogramy je vhodné umiestniť na koniec programu (za príkaz END a pred príkazy DATA). Podprogram môže obsahovať viac príkazov RETURN. Z jedného podprogramu možno privolať iný podprogram.

Príklad 1.50

Nasledujúci úsek programu ilustruje možnosť použitia podprogramu na skloňovanie slova BOD (napr. pri vyhodnocovaní výsledkov nejakého bodovo ohodnoteného testu):

```
10 REM TEST
    .
    .
    .
500 GOSUB 2000
510 PRINT "ZISKALI STE ";B;W$
    .
    .
    .
900 END
```

```
2000 REM PODPROGRAM SKLONOVANIA
2010 T$="BOD"
2020 IF ABS(B)=1 THEN W$=T$ : RETURN
2030 IF ABS(B)>1 AND ABS(B)<5 THEN W$=T$+"Y"
     : RETURN
2040 IF ABS(B)>4 OR B=0 THEN W$=T$+"OV"
2050 RETURN
```

Pred zobrazením správy o počte získaných bodov privoláme príkazom

```
500 GOSUB 2000
```

podprogram skloňovania, ktorý podľa hodnoty B vytvorí v reťazcovej premennej W$ správny tvar. Po spracovaní príkazu RETURN sa riadenie odovzdá príkazu

```
510 PRINT "ZISKALI STE ";B;W$
```

ktorý na obrazovke vypíše príslušnú správu, napr.

```
ZISKALI STE 12 BODOV
```

Podprogram v záujme väčšej univerzálnosti pracuje s funkciou ABS(B). Ak by sme namiesto nej použili premennú B, fungoval by správne len pre nezáporné čísla.

Na podmienečné privolanie jedného z viacerých podprogramov slúži príkaz

ON v GOSUB čr$_1$,čr$_2$,...,čr$_n$

kde v je aritmetický výraz, ktorý nadobúda hodnoty 1, 2, ..., n a $čr_j$ je číslo riadka s prvým príkazom podprogramu, ktorý sa má privolať, ak $v = j$. Ak hodnota výrazu v je menšia ako 1 alebo väčšia ako n, spracovanie programu pokračuje bezprostredne nasledujúcim riadkom.

Po ukončení privolávaného podprogramu (vykonaní príkazu RETURN) sa riadenie odovzdá príkazu, ktorý bezprostredne nasleduje za príkazom ON GOSUB.

Rozdiel medzi príkazmi ON GOTO a ON GOSUB spočíva v tom, že prepínač ON GOTO iba odovzdá riadenie požadovanému riadku bez návratu, zatiaľ čo prepínač podprogramov ON GOSUB po vykonaní podprogramu zabezpečuje návrat na bezprostredne nasledujúci riadok.

Príklad 1.51

Nasledujúci úsek programu ilustruje využitie prepínača ON GOSUB pri vyhodnocovaní nejakého testu (predpokladáme, že pre-

menná B obsahuje počet získaných bodov, pričom pri testovaní možno získať od 0 do 10 bodov):

```
600 K=INT(B/3)+1
610 ON K GOSUB 1100,1200,1300,1400
620 PRINT T$
      .
      .
      .
      .
1000 END
1100 REM HODNOTENIE PRE B=0,1,2
1110 T$="ZE SA NEHANBIS, TVOJE ZNALOSTI SU
     NULOVE!"
1120 RETURN
1200 REM HODNOTENIE PRE B=3,4,5
1210 T$="TVOJE ZNALOSTI SU VELMI SLABE!"
1220 RETURN
1300 REM HODNOTENIE PRE B=6,7,8
1310 T$="TVOJE ZNALOSTI SU PRIEMERNE!"
1320 RETURN
1400 REM HODNOTENIE PRE B=9,10
1410 T$="GRATULUJEM, TVOJE ZNALOSTI SU VELMI
     DOBRE!"
1420 RETURN
```

Uvedený úsek programu funguje tak, že premenná K nadobúda hodnoty

1 pre B = 0, 1, 2
2 pre B = 3, 4, 5
3 pre B = 6, 7, 8
4 pre B = 9, 10

a v príkaze

```
610 ON K GOSUB 1100,1200,1300,1400
```

sa v súlade s dosiahnutým počtom bodov privolá podprogram, ktorý naplní reťazcovú premennú T$ príslušným textom. Príkaz RETURN potom odovzdá riadenie bezprostredne nasledujúcemu príkazu

```
620 PRINT T$
```

ktorý vypíše hodnotiaci text. Dvojicu príkazov 600 a 610 môžeme nahradiť jediným príkazom

```
610 ON INT(B/3)+1 GOSUB 1100,1200,1300,1400
```

Všimnime si, že aritmetický výraz INT(B/3)+1 využíva funkciu INT na výpočet celých čísel, ktoré zodpovedajú stanoveným intervalom získaných bodov, napr.

INT(2/3) = 0, teda INT(2/3) + 1 = 1

Príkazy GOSUB, RETURN a ON GOSUB sa rovnakým spôsobom používajú v Atari basicu i v ďalších dialektoch basicu.

1.4.9 PRÍKAZY NA PRÁCU SO SÚBORMI NA KAZETÁCH

Údaje, ktoré odovzdávame programu vkladaním cez klávesnicu príkazom INPUT, zostanú v pamäti počítača iba počas behu programu. Pri novom odštartovaní programu príkazom RUN sa všetky číselné premenné vynulujú a všetky reťazce vymažú. Zachovanie údajov pre rôzne spustenia programu nám umožňujú zoznamy konštánt DATA, tie však znižujú univerzálnosť programu, pretože sú jeho súčasťou. Pri práci s väčšími súbormi údajov je výhodné uchovávať vložené údaje v tvare údajových súborov na kazete a pri spracovaní v programe ich „čítať" z kazety. Na tieto účely používame príkazy DSAVE (zapísanie údajového súboru na kazetu) a DLOAD (načítanie údajového súboru z kazety do pamäte počítača).

a) Zápis údajov na kazetu príkazom DSAVE

Údaje uložené v číselnom alebo reťazcovom poli môžeme zapísať na kazetu magnetofónu príkazom

```
DSAVE čz;pole "komentár
```

kde *čz* je číslo záznamu na kazete (celé číslo od 0 do 63), pomocou ktorého sa na záznam odvolávame pri jeho čítaní, *pole* je indexovaná premenná, označujúca prvý prvok poľa, od ktorého chceme obsah príslušného poľa zapísať na kazetu, *komentár* je maximálne 8-znakový text, ktorý charakterizuje údaje.

Pole, ktoré chceme príkazom DSAVE zapísať na kazetu, musíme v programe deklarovať príkazom DIM. Komentár nie je povinný, odporúčame ho však používať, pretože uľahčuje hľadanie záznamu na kazete.

Príprava magnetofónu na zápis údajového súboru príkazom DSAVE je rovnaká ako pri zápise programu povelom SAVE. Pred príkaz DSAVE v programe vkladáme príkaz PRINT, DISP alebo ? s upo-

zornením, že treba pripraviť magnetofón na zápis, a na čas prípravy magnetofónu zastavíme beh programu (napr. príkazom INPUT, ktorý očakáva vloženie správy, že používateľ už uviedol magnetofón do chodu na nahrávanie).

PROGRAM 1.7

Predpokladajme, že chceme vytvoriť súbor otázok a ponúkaných odpovedí pre nejaký test s výberovými odpoveďami. Súbor má nasledujúcu štruktúru: je uložený v jednorozmernom reťazcovom poli, v ktorom nultý prvok obsahuje názov testu, prvý prvok počet otázok (testových položiek) a druhý prvok počet ponúkaných odpovedí na každú otázku (počet otázok a počet ponúkaných odpovedí sa ukladajú ako reťazce číslic). Od tretieho prvku poľa začínajú jednotlivé testové položky tak, že najprv ide otázka (tretí prvok), za ňou ponúkané odpovede (toľko prvkov, koľko je odpovedí) atď. Program, ktorý zabezpečuje vloženie súboru a jeho uchovanie na kazete, uvádzame vo výpise 1.7.

```
10 REM PROGRAM 1.7 - TEST1, 10. 4. 88
20 DIM T$(300)
30 GCLEAR
100 GOSUB 1000
990 END
1000 REM VLOZENIE SUBORU TESTOVYCH POLOZIEK
1010 DISP "VLOZTE NAZOV TESTU:"
1020 INPUT N$ : T$(0)=N$
1030 DISP "VLOZTE POCET VYBEROVYCH ODPOVEDI:"
1040 INPUT M : T$(2)=STR$(M)
1050 I=0
1060 I=I+1
1070 K=2+(I-1)*(M+1)+1
1080 DISP "VLOZTE TEXT OTAZKY C.";I
1090 INPUT P$ : T$(K)=P$
1100 FOR J=1 TO M
1110 L=2+(I-1)*(M+1)+1+J
1120 DISP "VLOZTE VYBEROVU ODPOVED C.";J
1130 INPUT P$ : T$(L)=P$
1140 NEXT J
1150 DISP "CHCETE UKONCIT VKLADANIE? (A, N):"
1160 INPUT Q0$
1170 IF Q0$="A" THEN 1200
1180 IF Q0$="N" THEN 1060
1190 GOTO 1150
1200 N=I : T$(1)=STR$(N)
1300 REM ZAPIS POLA T$ NA KAZETU
1310 GCLEAR
1320 PRINT "1. NASTAVTE KAZETU NA ZACIATOK"
1330 PRINT "2. PRIPRAVTE MAGNETOFON NA NAHRAVANIE"
1340 PRINT "3. STLACTE EOL"
1350 INPUT Q0$
1360 DSAVE 05;T$(0),"TEST
1370 PRINT : PRINT "ZAPIS NA DISKETU UKONCENY"
1380 PRINT "ZASTAVTE MAGNETOFON A STLACTE EOL!"
1390 INPUT Q0$
1400 RETURN
```

Program funguje takto:

Najprv vložíme do poľa T$ súbor testových položiek (počet testových položiek N nevkladáme, pretože program si sám eviduje, koľko testových položiek sme vložili, a po ukončení vkladania zapíše skutočný počet testových položiek do prvku T$(1) poľa T$). Stačí zadať názov testu a počet výberových odpovedí M, ktorý je pre

všetky položky rovnaký. Predpokladáme, že vždy iba jedna z výberových odpovedí je správna; správnu odpoveď budeme vkladať ako prvú. Ak má náš súbor N testových položiek s M výberovými odpoveďami, potrebujeme na jeho uchovanie deklarovať pole T$ rozsahu aspoň 2 + N*(M+1).

Zápis súboru na kazetu vykoná úsek programu v riadkoch 1300 až 1390. Predpokladáme, že zapisovaný súbor je prvým súborom na kazete. Pred začiatkom zápisu program vypíše pokyn o príprave magnetofónu.

Pri tvorbe otázok a výberových odpovedí treba vychádzať z formátu obrazovky (tak, aby sa jedna testová položka mohla zobraziť v peknej grafickej úprave na jednej obrazovke).

Program 1.7 sme usporiadali tak, aby sa dal ľahko rozširovať a dopĺňať. Úsek programu na vloženie a zápis súboru testových položiek má formu podprogramu, ktorý privoláme príkazom

100 GOSUB 1000

V číslovaní riadkov máme dostatočnú rezervu na to, aby sme mohli privolávať ďalšie podprogramy – na prečítanie súboru z kazety, jeho opravy a rozširovanie, na vlastné testovanie a vyhodnocovanie výsledkov.

b) Načítanie súboru údajov z kazety príkazom DLOAD

Údajový súbor na kazete vyhľadáme a načítame do pamäte počítača príkazom

```
DLOAD čz;pole
```

kde *čz* je číslo záznamu na kazete, pod ktorým bol súbor na kazetu zapísaný príkazom DSAVE, *pole* je indexovaná premenná, označujúca prvok poľa, od ktorého chceme súbor z kazety načítať do príslušného poľa programu.

Pole, do ktorého chceme príkazom DLOAD načítať súbor z kazety, musí byť v programe deklarované príkazom DIM. Rozmery poľa, do ktorého chceme načítať náš súbor, nesmú byť menšie ako rozmery poľa, zapísaného na kazetu. Príprava magnetofónu na načítanie údajového súboru príkazom DLOAD je rovnaká ako pri čítaní programu povelom LOAD.

Príklad 1.52

V programe 1.7 sme vytvorili súbor testových položiek v poli T$ a zapísali ho ako prvý súbor na kazetu. Zostavíme podprogram na

prečítanie tohto súboru z kazety tak, aby sme s ním mohli pracovať (opraviť alebo doplniť testové položky, vykonať vlastné testovanie žiakov).

```
2000 REM PODPROGRAM VSTUPU POLA T$ Z KAZETY
2010 GCLEAR
2020 PRINT "1. VLOZTE A NASTAVTE KAZETU"
2030 PRINT "2. PRIPRAVTE MAGNETOFON NA
     CITANIE"
2040 PRINT "3. PO ZAPNUTI MAGNETOFONU STLACTE
     EOL"
2050 INPUT Q0$
2060 DLOAD 01;T$(0)
2070 PRINT
2080 PRINT "CITANIE Z MAGNETOFONU UKONCENE"
2090 PRINT "ZASTAVTE MAGNETOFON A STLACTE
     EOL"
2100 INPUT Q0$
2110 RETURN
```

Príkazy DSAVE a DLOAD sú špeciálnymi príkazmi dialektu basic G. Príkazy na prácu so súbormi sú aj v iných dialektoch basicu (napr. v Atari basicu príkazy OPEN, CLOSE, INPUT #n, PRINT #n), používajú sa však odlišným spôsobom.

1.4.10 NIEKOĽKO UŽITOČNÝCH POZNATKOV

V tomto odseku uvádzame výber poznatkov, ktoré nám pomôžu pri vytváraní a overovaní správnosti programov a pri pokusoch o náročnejšie programy.

a) Ladenie programov

Basic vznikol ako interpretačný jazyk. Znamená to, že na rozdiel od väčšiny iných programovacích jazykov, ktoré umožňujú vykonávať iba ucelené, skompilované programy (po ich preložení do strojového kódu), v basicu sa programy vykonávajú po jednotlivých príkazoch. Interpret pri vykonávaní programu postupne „prekladá" jednotlivé príkazy a hneď ich vykonáva, pričom po vykonaní každého príkazu „zabudne" jeho preloženú formu. Má to svoje výhody a nevýhody. Nevýhodou interpretačného basicu v porovnaní s inými jazykmi je to, že programy sa vykonávajú podstatne pomalšie. Výhodou však je možnosť štartovať program po častiach už v etape jeho

vytvárania a jednoduchým spôsobom si overovať správnosť jeho jednotlivých úsekov.

Overovanie správnosti programu nazývame ladením programu. V rámci ladenia programu si overujeme dve veci – či program neobsahuje formálne chyby (napr. nesprávne zapísané príkazy) a či neobsahuje logické chyby (t. j. či robí to, čo sme plánovali pri jeho zostavovaní).

V basicu môžeme už po vložení prvých príkazov skúšať, či doteraz vložená časť programu nemá závady.

Príklad 1.53

Predpokladajme, že chceme zostaviť program na výpočet k-tého člena geometrickej postupnosti

$a_k = a_1 q^{k-1}$

Pri zostavovaní programu môžeme postupovať takto:

1. Vložíme príkazy

```
10 REM GEOMETRICKA POSTUPNOST
20 PRNT "VLOZTE KVOCIENT Q" : INPUT Q
```

a povelom RUN ich vykonáme. Keďže sme sa pomýlili v zápise riadka 20 (namiesto PRINT sme napísali PRNT), vykonávanie programu sa pri spracovaní tohto riadka zastaví a vypíše sa správa o chybe

```
SYNTAX ERROR
```

Ihneď chybu opravíme a znovuspustením si overíme správnosť tohto úseku programu.

2. Vložíme ďalšiu časť programu, napr.

```
30 PRINT "VLOZTE PRVY CLEN" : INPUT A1
40 PRINT "VLOZTE CISLO K" : INPUT K
```

a znovu spustíme celú doteraz vloženú časť programu povelom RUN a skontrolujeme jej správnosť.

3. Vložíme zvyšok programu

```
50 PRINT "K-TY CLEN GEOMETRICKEJ POSTUPNOSTI"
60 PRINT "KDE K = ";K
70 PRINT "S PRVYM CLENOM A1 = ";A1
80 PRINT "A S KVOCIENTOM Q = ";Q
90 PRINT "JE"
100 AK=A1*Q^(K-1)
110 PRINT "AK = ";AK
```

Pre spustenie celého programu si zvolíme údaje q, a_1, k tak, aby sme poznali výsledok, napr.

$a_1 = 5, q = 2, k = 7$

pre ktoré

$a_7 = 5.2^6 = 320$

a presvedčíme sa o správnosti programu. Pri overovaní správnosti programu je vhodné vykonať ho viackrát s rôznymi údajmi, pre ktoré poznáme výsledky.

4. Keď sa presvedčíme o tom, že program „robí" to, čo od neho požadujeme, pristúpime k jeho vylepšovaniu. Nepáči sa nám napríklad grafická úprava výstupov, ktoré nasledujú bezprostredne za sebou a sú neprehľadné. Doplníme preto program tak, aby jednotlivé časti výstupov boli oddelené prázdnymi riadkami. Vložíme príkazy

```
45 PRINT
95 PRINT
```

Po overení správnosti vykonaných úprav môžeme program ďalej zdokonaľovať a rozširovať, napr. pred začiatkom výstupov vymazať príkazom GCLEAR obrazovku, doplniť program tak, aby vypočítal súčet prvých k členov geometrickej postupnosti, a pod.

b) Využívanie kľúčov pomocou funkcie INKEY

V hornej časti klávesnice počítačov PMD 85, resp. Didaktik Alfa máme k dispozícii 12 funkčných klávesov KØ, K1, ..., K11 (resp. Ø, ①, ..., ⑪ na Didaktiku Alfa). Funkcia

```
INKEY
```

umožňuje načítať hodnotu stlačeného funkčného klávesu od 0 do 11 podľa čísla stlačeného klávesu. Ak nie je stlačený žiadny kláves, funkcia INKEY má hodnotu 255.

Funkciu INKEY používame predovšetkým na programovanie riadenia chodu programu.

Príklad 1.54

Predpokladajme, že program konštruujeme tak, aby si používateľ mohol po odštartovaní programu a v priebehu jeho vykonávania vyberať z určitej ponuky činností (napr. vstup údajov, prezeranie údajov, oprava údajov, zápis údajov na kazetu, ukončenie práce). Na výber z ponuky použijeme funkčné klávesy napr. takto:

```
1Ø REM VYBER Z PONUKY FUNKCNYMI KLAVESMI
2Ø REM S POUZITIM FUNKCIE INKEY
```

```
100 GCLEAR
110 PRINT "MATE TIETO MOZNOSTI:" : PRINT
120 PRINT "    1. VSTUP UDAJOV   (KLAVES K1) "
130 PRINT "    2. PREZERANIE UDAJOV  (KLAVES K2) "
140 PRINT "    3. OPRAVA UDAJOV   (KLAVES K3) "
150 PRINT "    4. ZAPIS UDAJOV NA KAZETU  (KLAVES
    K4) "
160 PRINT "    5. UKONCENIE PRACE   (KLAVES K5) "
170 DISP "STLACTE PRISLUSNY FUNKCNY KLAVES!"
180 IF INKEY<1 OR INKEY>5 THEN 180
190 IF INKEY=5 THEN 220
200 ON INKEY GOSUB 1000,2000,3000,4000
210 GOTO 100
220 END
1000 REM PODPROGRAM VSTUPU UDAJOV
      .
      .
      .
1990 RETURN
2000 REM PODPROGRAM PREZERANIA UDAJOV
      .
      .
      .
2990 RETURN
3000 REM PODPROGRAM OPRAVY UDAJOV
      .
      .
      .
3990 RETURN
4000 REM PODPROGRAM ZAPISU NA KAZETU
      .
      .
      .
4990 RETURN
```

Pri stlačení funkčného klávesu netreba odosielať načítanú hodnotu do pamäte stlačením EOL. Hodnota funkčného klávesu je uložená v premennej INKEY. V prípade, že v čase vykonávania príkazu 180 nie je stlačený niektorý z funkčných klávesov K1, K2, K3, K4,

K5, príkaz sa zopakuje, teda čaká sa na stlačenie príslušného funkčného klávesu. Po vykonaní zvoleného podprogramu sa odovzdá riadenie príkazu 210, ktorý obnoví ponuku možností.

c) Priame čítanie stlačeného klávesu

Ukázali sme, ako priamo čítať funkčné klávesy pomocou funkcie INKEY. Podobným spôsobom môžeme čítať aj ostatné klávesy. Použijeme na to podprogram v strojovom kóde, ktorý je súčasťou programu monitora počítačov PMD 85 a Didaktik Alfa.

Podprogram v strojovom kóde privoláme a vykonáme v rámci programu v basicu funkciou

```
USR(adresa)
```

kde *adresa* je celé číslo, ktoré udáva adresu (dekadicky) začiatku podprogramu v pamäti počítača. Pamäť počítačov PMD 85 a Didaktik Alfa môžeme adresovať od 0 do 65536. Najväčšie celé číslo, s ktorým môžeme v basicu pracovať, je však iba 32767. Vzhľadom na to adresy od 0 do 32767 uvádzame vo funkcii USR priamo a adresy väčšie ako 32767 v tvare

adresa – 65536

Podprogram na čítanie stlačeného klávesu má adresu 33953, preto sa naň vo funkcii USR odvolávame číslom

33953 – 65536 = –31583

Po vykonaní funkcie

USR(–31583)

bude premenná USR obsahovať desiatkovú číselnú hodnotu znaku reprezentovaného príslušným klávesom v ASCII kóde (pozri prílohu 3). Aby sme túto hodnotu previedli na príslušný znak, použijeme funkciu

```
CHR$(kód)
```

kde kód je desiatková hodnota ASCII kódu (od 0 do 255). Po vykonaní príkazu

Z$=CHR$(USR(–31583))

bude teda premenná Z$ obsahovať jeden znak, vložený stlačením niektorého klávesu. Príkaz funguje tak, že pri vykonávaní funkcie USR sa spracovanie programu preruší, až pokým nestlačíme niektorý z klávesov. Týmto spôsobom môžeme teda nahradiť príkaz INPUT, ktorý požaduje vloženie jedného znaku z klávesnice, pričom vložený znak nemusíme odosielať stlačením EOL.

Príklad 1.55

V dialógových programoch sa často vyskytujú dotazy, na ktoré očakávame odpoveď v tvare A (áno) alebo N (nie). Nasledujúci úsek programu ukazuje, ako spracovávať odpoveď na takéto dotazy pomocou priameho čítania stlačeného klávesu:

```
200 DISP "CHCETE POKRACOVAT? (A, N):"
210 Z$=CHR$(USR(-31583))
220 IF Z$="A" THEN 250
230 IF Z$="N" THEN 500
240 GOTO 210
250 REM POKRACOVANIE


500 END
```

PROGRAM 1.8

Výpis 1.8 obsahuje užitočný program, pomocou ktorého zistíme desiatkové hodnoty ASCII kódu jednotlivých znakov zodpovedajúcich príslušným klávesom. Po stlačení klávesu sa na obrazovke vypíše príslušný znak a jemu zodpovedajúci ASCII kód.

```
10 REM PROGRAM 1.8 - KOD ASCII, 11. 4. 88
20 GCLEAR
30 DISP "STLAC POZADOVANY KLAVES"
40 K=USR(-31583) : K$=CHR$(K)
50 PRINT "ZNAK  "$K$$TAB(10)$"ASCII KOD  "$K
60 DISP "CHCES POKRACOVAT? (A, N):"
70 Z$=CHR$(USR(-31583))
80 IF Z$="A" THEN PRINT : GOTO 30
90 IF Z$="N" THEN 110
100 GOTO 60
110 END
```

d) Príkaz ON ERR na spracovanie chýb

V prípade výskytu chyby interpret jazyka basic preruší vykonávanie programu a vypíše na obrazovke hlásenie o chybe v tvare

* * * správa IN LINE čr * * *

kde *čr* je číslo programového riadka, pri vykonávaní ktorého došlo k chybe. Hlásenia chýb na počítačoch PMD 85–1, PMD 85–2 a Didaktik Alfa sú rovnaké (zoznam chýb pozri v prílohe 1).

Na počítači PMD 85–2 máme možnosť potlačiť hlásenie chýb a ošetriť výskyt chyby priamo v programe. Slúži na to príkaz

ON ERR príkaz

kde *príkaz* je nejaký príkaz jazyka basic (spravidla skok), ktorý sa má vykonať v prípade výskytu chyby. Vykonávanie programu sa teda v prípade chyby, ktorá vznikne nesprávnym zásahom používateľa, nepreruší, ale pokračuje spôsobom stanoveným v príkaze ON ERR.

Príklad 1.56

Úsek programu

```
100 ON ERR GOTO 200
110 DISP "VLOZ HODNOTU A:"
120 INPUT A
130 PRINT A;TAB(10);A^2;TAB(20);A^3
140 DISP "CHCES POKRACOVAT? (A, N):"
150 Z$=CHR$(USR(-31583))
160 IF Z$="A" THEN 110
170 IF Z$="N" THEN STOP
180 GOTO 140
```

```
200 REM OSETRENIE CHYBY
210 DISP "CHYBNY VSTUP, ZADAJ ZNOVA"
220 BEEP : BEEP : BEEP
230 GOTO 100
```

funguje takto: V prípade, že pri vkladaní hodnoty A dôjde k chybe (vložením iného ako číselného údaja), odovzdá sa riadenie riadku 200, ktorý vypíše správu o chybe. Na chybu upozorní aj trojnásobný zvukový signál. Program si potom znovu vyžiada vloženie číselnej hodnoty. Po každom spracovaní chyby treba znovu nastaviť príkazom ON ERR režim programového spracovania chýb.

V zložitejších prípadoch možno zistiť číslo riadka, v ktorom sa vyskytla chyba, a kód chyby (podľa kódov uvedených v prílohe so zoznamom chybových hlásení) pomocou príkazov

```
R=APEEK '5E5D
```

(riadok s výskytom chyby) a

```
K=PEEK '26
```

(kód chyby) a v závislosti od toho v programe bližšie špecifikovať spôsob spracovania chýb.

Poznámka 1.8
V Atari basicu na tieto účely používame príkaz

```
TRAP čr
```

kde *čr* je číslo riadka, ktorému sa má odovzdať riadenie v prípade výskytu chyby. Číslo riadka, v ktorom sa vyskytla chyba, zistíme príkazom

```
R=PEEK(186)+256*PEEK(187)
```

a kód chyby príkazom

```
K=PEEK(195)
```

Zoznam chýb a ich kódov v Atari basicu uvádzame v prílohe 2.

1.5 GRAFIKA

Rôzne dialekty basicu pre osobné počítače obsahujú tzv. grafické príkazy, ktoré umožňujú programovať kreslenie grafov, obrázkov a pohybujúcich sa objektov. V tejto časti sa jednotlivé dialekty basicu najviac odlišujú. Oboznámime sa s grafickými príkazmi jazyka basic G. Ku grafike v Atari basicu, ktorá je založená na iných princípoch, sa vrátime v kapitole 1.7.

1.5.1 FORMÁTY OBRAZOVKY

Obrazovka, prostredníctvom ktorej komunikujeme s počítačmi PMD 85 a Didaktik Alfa, je rozdelená na dve časti: na pracovnú oblasť a dialógový riadok.

Do dialógového riadka sa zobrazujú údaje, ktoré programátor vkladá z klávesnice, výstupy programované príkazmi DISP a ? a správy počítača používateľovi. Dialógový riadok je posledným riadkom obrazovky a umožňuje zobraziť 48 znakov. Zvyšok obrazovky tvorí tzv. pracovná oblasť, ktorá má dva formáty: alfanumerický a grafický.

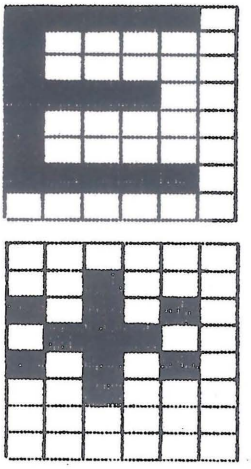
Alfanumerický formát umožňuje zobraziť 25 riadkov (číslovaných od 0 do 24 zhora dole) po 48 znakov v riadku (číslovaných od 0 do 47 zľava doprava). V tomto formáte sa zobrazujú výpisy programov povelom LIST a výstupy číselných a textových údajov príkazom PRINT.

Grafický formát umožňuje využívať obrazovku na kreslenie obrázkov, funkcií a grafov. Jazyk basic G obsahuje príkazy na využívanie grafického formátu obrazovky v dvoch režimoch.

V *bodovom režime* si môžeme pracovnú oblasť obrazovky predstaviť ako tabuľku s 243 riadkami (číslovanými od 0 do 242 zdola hore) a 256 stĺpcami (číslovanými od 0 do 255 zľava doprava). Na jednotlivé body obrazovky sa v grafických príkazoch odvolávame dvojicou súradníc (x, y), kde x je vodorovná súradnica (číslo stĺpca) a y je zvislá súradnica (číslo riadka). Grafiku v bodovom režime programujeme príkazmi SCALE, MOVE, PLOT, LABEL, AXES a FILL.

V *bajtovom režime* je obrazovka rozdelená na 243 riadkov (číslovaných od 0 do 242 zhora dole – teda opačne ako v bodovom režime) a 48 stĺpcov (číslovaných od 0 do 47 zľava doprava). Jednotkou zobrazenia je v tomto prípade vodorovná šestica bodov, na ktorú sa odvolávame zadaním čísla riadka a stĺpca. Spôsob, akým sa vykreslí táto šestica bodov, zadávame pomocou 1 bajtu špeciálne zostrojeného reťazca (preto hovoríme o bajtovom režime). Bajtový režim teda umožňuje programovať zobrazovanie 288 × 243 bodov. Grafiku v bajtovom režime programujeme príkazmi BMOVE a BPLOT.

Základným spôsobom zobrazovania je možnosť riadiť stav každého bodu pracovnej oblasti obrazovky – buď „svieti“, alebo „nesvieti“. Rozdiel medzi alfanumerickým a grafickým formátom spočíva iba v tom, že určité „obrázky“ (písmená, číslice a iné znaky na klávesnici) interpret basicu vykresľuje automaticky, pričom na každý znak používa 6 bodov vodorovne a 8 bodov zvisle, zatiaľ čo zložitejšie obrázky si musíme „nakresliť“ sami. Na obr. 1.6 sme pre ilustráciu znázornili spôsob, akým sa v alfanumerickom formáte zobrazuje písmeno E a znak * (hviezdička) – každý štvorček tu predstavuje jeden bod obrazovky.



1.6 Zobrazovanie štandardných znakov

1.5.2 JEDNODUCHÉ KRESLENIE PRÍKAZOM PRINT

Jednoduchšie obrázky v alfanumerickom formáte môžeme programovať pomocou už známych príkazov.

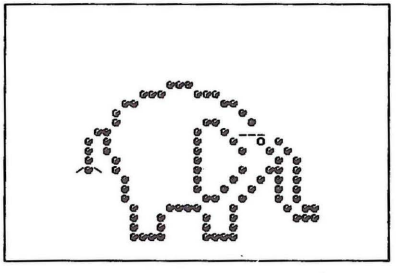
František Turnovec

PROGRAM 1.9

Výpis 1.9 obsahuje program na vykreslenie schematického obrázka slona. Kresbu, ktorá sa po odštartovaní programu zjaví na obrazovke, uvádzame na obr. 1.7.

```
10 REM PROGRAM 1.9 - SLON, 15. 4. 88
20 GCLEAR
30 READ S$
40 IF S$="#" THEN 100
50 PRINT TAB(10);S$
60 GOTO 30
100 END
1000 DATA "           @@@"
1010 DATA "        @@@   @@@"
1020 DATA "      @@        @@"
1030 DATA "    @             @"
1040 DATA "    @          @@   @"
1050 DATA "  @@          @  @ ___"
1060 DATA " @ @          @    @  @ @"
1070 DATA " @ @          @      @  @ @"
1080 DATA " @  @         @          @ @"
1090 DATA "/@\ @         @      @  @@ @"
1100 DATA "     @        @    @  @ @ @"
1110 DATA "     @        @  @  @  @ @"
1120 DATA "     @         @@  @    @ @"
1130 DATA "       @    @@@@   @      @ @@"
1140 DATA "      @ @     @  @       @@@"
1150 DATA "      @  @    @  @"
1160 DATA "      @@@@    @@@@"
1170 DATA "#"
```

Pri tomto spôsobe kreslenia, na ktorý používame príkaz PRINT, musíme vystačiť s formátom 48 stĺpcov a 25 riadkov, pretože obrázky sa vytvárajú pomocou štandardných znakov a najmenšou jednotkou zobrazenia je obdĺžnik 6 × 8 bodov. Na jemnejšie kresby s väčšou rozlišovacou schopnosťou je využitie príkazu PRINT nevhodné. Vzhľadom na to sa budeme ďalej zaoberať iba grafickými príkazmi bodového a bajtového režimu.



1.7 Výstup programu SLON

1.5.3 PRÍKAZY NA KRESLENIE V BODOVOM REŽIME

V bodovom režime (formát obrazovky 256 × 243 bodov) používame tieto grafické príkazy:

SCALE	nastavenie mierky zobrazenia
MOVE	nastavenie „pera" do konkrétneho bodu
PLOT	vykreslenie bodu alebo úsečky
LABEL	zväčšovanie a vykresľovanie znakov
AXES	kreslenie súradnicových osí
FILL	kreslenie plošných obrazcov

Na počítačoch PMD 85–1 a Didaktik Alfa tieto príkazy používame buď v jednopríkazových riadkoch, alebo ako posledné príkazy viacpríkazových riadkov. V programoch s grafikou pritom nesmieme používať premenné XØ, X1, X2, X3, YØ, Y1, Y2 a Y3. Na počítači PMD 85–2 tieto obmedzenia odpadajú.

a) Nastavenie mierky príkazom SCALE

Príkaz SCALE nastavuje mierku zobrazenia, t. j. vykonáva transformáciu medzi bodovým grafickým formátom obrazovky 256 × 243 bodov a súradnicami, ktoré chceme používať v našom programe. Používa sa v tvare

```
SCALE x1,x2,y1,y2
```

kde x_1 je dolná hranica nášho súradnicového systému na vodorovnej osi, x_2 je horná hranica nášho súradnicového systému na vodorovnej osi, y_1 je dolná hranica nášho súradnicového systému na zvislej osi a y_2 je horná hranica nášho súradnicového systému na zvislej osi.

Príklad 1.57

Príkaz

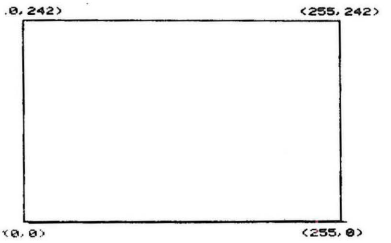
```
2Ø SCALE Ø,255,Ø,242
```

spôsobí, že body na vodorovnej osi sú očíslované zľava doprava od 0 do 255 a na zvislej osi zdola hore od 0 do 242, pričom súradnice bodov používané v programe sa zhodujú so súradnicami grafického formátu obrazovky (pozri obr. 1.8).

Príkaz

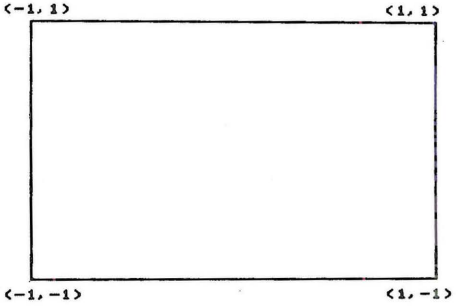
```
SCALE -1,1,-1,1
```

spôsobí, že jednému bodu grafického formátu obrazovky zodpovedá približne hodnota



1.8 Nastavenie mierky príkazom SCALE Ø,255,Ø,242

2/256 = 0.0078
jednotky súradníc na vodorovnej osi a
2/243 = 0.0082
jednotky súradníc na zvislej osi súradnicového systému používaného v programe, pričom body na vodorovnej osi sú očíslované od –1 do 1 zľava doprava a na zvislej osi od –1 do 1 zdola nahor (pozri obr. 1.9).



1.9 Nastavenie mierky príkazom SCALE –1,1,–1,1

Vidíme teda, že príkaz SCALE vykonáva transformáciu medzi nami zadanou mierkou zobrazenia a grafickým formátom obrazovky podľa vzorca

$$\frac{x_2 - x_1}{256}$$

pre vodorovné súradnice a

$$\frac{y_2 - y_1}{243}$$

pre zvislé súradnice.

Príkaz SCALE treba použiť pred prvým použitím ostatných grafických príkazov bodového režimu.

b) Nastavenie pera príkazom MOVE

V rámci mierky zadanej príkazom SCALE a určujúcej súradnice používané v programe umiestnime pomyselné „pero" do konkrétneho bodu obrazovky príkazom

```
MOVE x,y
```

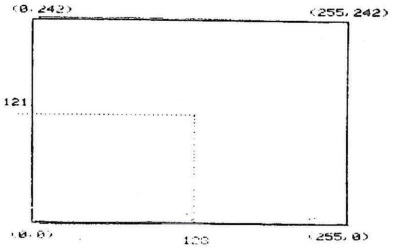
kde *x* je vodorovná súradnica požadovaného bodu v nami zvolenej mierke, *y* je zvislá súradnica požadovaného bodu v nami zvolenej mierke. Za *x* a *y* môžeme dosadiť aritmetické výrazy.

Príklad 1.58

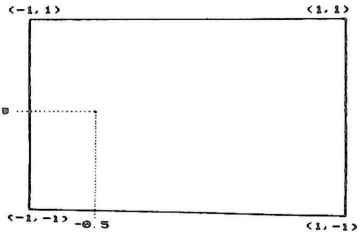
V úseku programu

```
10 REM PRIKAZ MOVE
20 SCALE 0,255,0,242
30 MOVE 128,121
```

nastaví príkaz v riadku 30 „pero" do stredu obrazovky (pozri obr. 1.10).



1.10 Nastavenie „pera" príkazom MOVE 128,121



1.11 Nastavenie „pera" príkazom MOVE –0.5,0

V úseku programu

```
10 REM PRIKAZ MOVE
20 SCALE -1,1,-1,1
30 MOVE -0.5,0
```

nastaví príkaz v riadku 30 „pero" do stredu ľavej polovice obrazovky (pozri obr. 1.11).

Príkaz MOVE nezobrazuje polohu nastavenia pera.

c) Zväčšovanie znakov príkazom LABEL

Príkaz LABEL slúži na zobrazovanie zväčšených znakov. Používa sa v tvare

LABEL k_1,k_2;zoznam

kde k_1 je zväčšenie základného formátu znakov v smere vodorovných súradníc, k_2 je zväčšenie základného formátu znakov v smere zvislých súradníc a *zoznam* je zoznam vystupujúcich znakových alebo číselných údajov, pre ktorý platia rovnaké pravidlá ako pre zoznam príkazu PRINT.

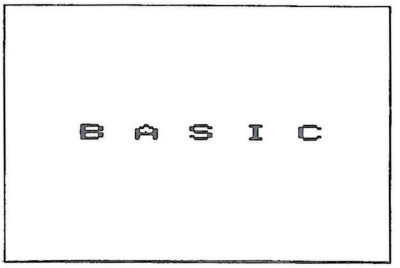
Ako sme už uviedli, základný formát zobrazenia jedného znaku (číslica, písmeno, znamienko) je 6 × 8 bodov, teda jeden znak štandardne zaberá obdĺžniček šírky 6 bodov a výšky 8 bodov. Príkazom LABEL zobrazujeme znaky vo formáte, ktorý je celočíselným násobkom tohto základného formátu, pričom môžeme programovať rôzne zväčšenia formátu vo vodorovnom a zvislom smere. Miesto, kde sa majú na obrazovke príslušné znaky vypísať, určujeme nastavením „pera" príkazom MOVE v súradniciach zadaných príkazom SCALE.

Príklad 1.59

Príkazmi

```
10 REM PRIKAZ LABEL
20 GCLEAR
30 SCALE 0,255,0,242
40 MOVE 50,115
50 LABEL 3,2;"B A S I C"
```

vypíšeme do stredu obrazovky zväčšený text B A S I C (pozri obr. 1.12). Každý znak zaberá plochu 18 × 16 bodov (trojnásobné zväčšenie vo vodorovnom smere a dvojnásobné zväčšenie vo zvislom smere). Zmenou parametrov v príkaze LABEL môžeme rôzne meniť šírku a výšku zobrazovaných znakov. Napríklad príkazom



1.12 Výpis zväčšeného textu príkazom LABEL

```
50 LABEL 3,8;"B A S I C"
```

vypíšeme text, v ktorom sú znaky trojnásobne zväčšené do šírky a osemnásobne do výšky, teda zaberajú 18 × 64 bodov.

Pri spracovaní príkazu LABEL sa zobrazovanie zväčšených znakov uskutočňuje zdola nahor a zľava doprava. Po ukončení zobrazenia zostane „pero" nastavené na pravej strane posledného zobrazovaného znaku.

Príklad 1.60

Pomocou príkazov MOVE a LABEL môžeme riadiť polohu výpisu na obrazovke aj v programoch, ktoré nie sú určené na kreslenie na obrazovke. Ak chceme zmeniť štandardný spôsob výpisov príkazom PRINT (rolovanie zdola nahor) na výpisy zhora nadol, môžeme postupovať napr. takto:

```
100 GCLEAR
110 SCALE 0,255,0,242
120 FOR I=1 TO 10
130 MOVE 0,240-I*10
140 LABEL 1,1;T$(I)
150 NEXT I
```

V tomto prípade sa obsah reťazcového poľa T$ vypíše na obrazovku (v štandardnom formáte) zhora nadol.

d) Kreslenie príkazom PLOT

Príkazom PLOT zobrazíme bod na obrazovke a spojíme ho úsečkou s predchádzajúcou polohou nastavenia „pera". Používame ho v tvare

```
PLOT x,y,k
```

kde x je vodorovná súradnica bodu, ktorý chceme zobraziť, y je zvislá súradnica bodu, ktorý chceme zobraziť, a k je parameter spôsobu zobrazenia: ak $k = 1$, zobrazí sa iba bod (x, y), ak $k \neq 1$ alebo nie je uvedené, zobrazí sa bod (x, y) spolu s úsečkou, spájajúcou ho s posledným bodom, na ktorom bolo nastavené „pero" pred vykonaním príkazu PLOT.

Súradnice x a y v príkaze PLOT udávame v mierke zadanej príkazom SCALE.

Príklad 1.61

Príkazmi

```
200 GCLEAR
210 SCALE 0,255,0,242
220 MOVE 10,10
230 PLOT 100,10
```

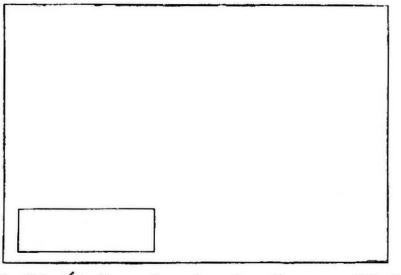
nakreslíme na obrazovke úsečku, ktorá spája body (10, 10) a (100, 10), teda vodorovnú úsečku v ľavej dolnej časti obrazovky.

Príklad 1.62

Úsek programu

```
200 GCLEAR
210 SCALE 0,255,0,242
220 MOVE 10,10
230 PLOT 100,10
240 PLOT 100,50
250 PLOT 10,50
260 PLOT 10,10
```

nakreslí obdĺžnik v ľavej dolnej časti obrazovky (pozri obr. 1.13).



1.13 Obdĺžnik nakreslený príkazom PLOT

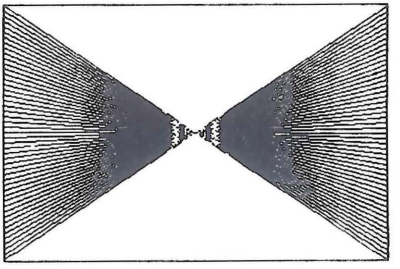
SKÚSTE TO S BASICOM

Príklad 1.63

Pomerne jednoduchým použitím príkazu PLOT môžeme kresliť efektné obrázky. Napr. program

```
10 REM OBRAZOK
20 GCLEAR
30 SCALE 0,255,0,242
40 FOR Y=0 TO 242 STEP 5
50 MOVE 255,Y
60 PLOT 1,242-Y
70 NEXT Y
```

nakreslí motív uvedený na obr. 1.14.



1.14 Kreslenie príkazom PLOT

V druhej časti knihy ukážeme, ako príkaz PLOT (hoci je určený na kreslenie úsečiek) možno použiť na kreslenie kružnice, elipsy a ďalších geometrických útvarov.

e) Príkaz AXES a kreslenie grafov funkcií

Príkaz AXES slúži na nakreslenie zvislej a vodorovnej súradnicovej osi. Používa sa v tvare

```
AXES x,y
```

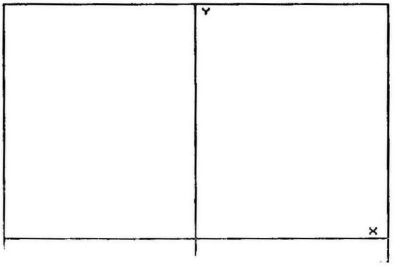
kde x je vodorovná súradnica a y je zvislá súradnica bodu, v ktorom sa majú pretínať súradnicové osi. Súradnice x a y sa zadávajú v mierke definovanej príkazom SCALE. Pri spracovaní príkazu AXES sa najprv vykreslí vodorovná súradnicová os (zľava doprava) a potom zvislá os (zhora nadol). Po ukončení kreslenia zostane „pero" nastavené na dolnom konci zvislej súradnicovej osi, čo treba mať na zreteli pri zadávaní ďalších grafických príkazov.

Príklad 1.64

Úsek programu

```
10 REM PRIKAZ AXES
20 GCLEAR
30 SCALE -10,10,-10,100
40 AXES 0,0
50 MOVE 9,1
60 LABEL 1,1;"X"
70 MOVE 0.3,95
80 LABEL 1,1;"Y"
```

nakreslí súradnicové osi s priesečníkom v bode (0, 0), pričom príkazmi MOVE a LABEL označí vodorovnú os ako X a zvislú os ako Y (pozri obr. 1.15). Všimnime si, že mierka je definovaná od −10 do 10 pre vodorovné súradnice a od −10 do 100 pre zvislé súradnice.



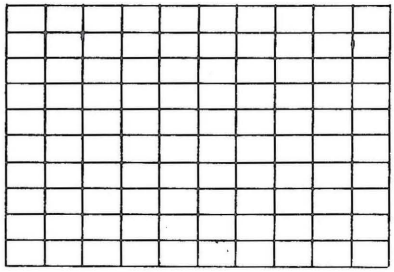
1.15 Kreslenie súradnicových osí príkazom AXES

Príklad 1.65

Príkaz AXES môžeme použiť aj na iné účely ako na kreslenie súradnicových osí. Napríklad úsek programu

```
10 REM TABULKA
20 GCLEAR
30 SCALE 0,10,0,10
40 FOR I=0 TO 10
50 AXES I,I
60 NEXT I
```

nakreslí tabuľku, ktorá má 10 riadkov a 10 stĺpcov, teda sústavu vodorovných a zvislých čiar, ktoré sa pretínajú v bodoch (0, 0), (1, 1), (2, 2) atď. (pozri obr. 1.16).



1.16 Kreslenie tabuľky príkazom AXES

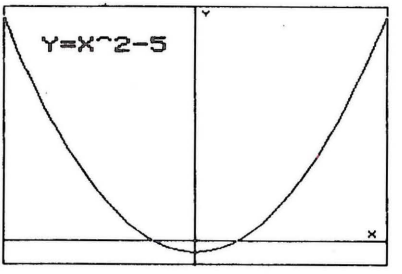
Príkaz AXES používame najmä v súvislosti s kreslením grafov funkcií príkazom PLOT.

PROGRAM 1.10

Výpis 1.10 obsahuje program, ktorý nakreslí graf paraboly $y = x^2 - 5$ v intervale $-10 \leqq x \leqq 10$ s krokom zmeny premennej x 0.5. Zároveň

```
10 REM PROGRAM 1.10 - PARABOLA, 20. 4. 88
20 GCLEAR
30 SCALE -10,10,-10,100
40 AXES 0,0
50 MOVE 9,1
60 LABEL 1,1;"X"
70 MOVE 0.3,95
80 LABEL 1,1;"Y"
100 REM GRAF FUNKCIE Y = X^2 - 5
110 MOVE -10,95
120 FOR X=-10 TO 10 STEP 0.5
130 Y=X^2-5
140 PLOT X,Y
150 NEXT X
160 MOVE -8,80
170 LABEL 2,2;"Y=X^2-5"
180 END
```

nakreslí súradnicové osi s počiatkom v bode (0, 0) a pomocou príkazu LABEL vypíše v ľavej hornej časti obrazovky zobrazovanú funkciu (pozri obr. 1.17). Vhodným definovaním mierky príkazom SCALE zabezpečíme, aby sa zobrazovaná časť grafu funkcie zmestila na obrazovku.



1.17 Výstup programu PARABOLA

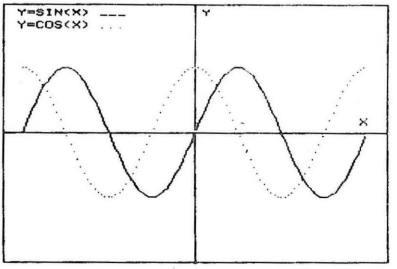
PROGRAM 1.11

Vo výpise 1.11 uvádzame program, ktorý nakreslí grafy funkcií $y = \sin(x)$ a $y = \cos(x)$ v intervale od -2π do 2π (pozri obr. 1.18).

```
10 REM PROGRAM 1.11 - SINUSOVKA, 21. 4. 88
20 GCLEAR
30 SCALE -7,7,-2,2
40 AXES 0,0
50 MOVE 6,0.1
60 LABEL 1,1,"X"
70 MOVE 0.2,1.8
80 LABEL 1,1,"Y"
90 PI=3.14159
100 REM GRAF FUNKCIE SINUS
110 MOVE -2*PI,0
120 FOR X=-2*PI TO 2*PI STEP PI/50
130 PLOT X,SIN(X)
140 NEXT X
150 REM GRAF FUNKCIE COSINUS
160 FOR X=-2*PI TO 2*PI STEP PI/25
170 PLOT X,COS(X),1
180 NEXT X
200 MOVE -6.5,1.8
210 LABEL 1,1,"Y=SIN(X) ___"
220 MOVE -6.5,1.6
230 LABEL 1,1,"Y=COS(X) ..."
240 END
```

Graf funkcie sínus je vykreslený súvisle a graf funkcie kosínus bodkovane. Na „bodkované kreslenie" sme využili príkaz PLOT s tretím parametrom rovným 1, teda

17Ø PLOT X,COS(X),1



1.18 Výstup programu SÍNUSOVKA

f) Kreslenie príkazom FILL

Príkaz FILL slúži na kreslenie obdĺžnikových plošných obrázkov. Používa sa v tvare

FILL x,y;maska

kde x je počet bodov základného formátu obrazovky v smere vodorovnej osi, y je počet bodov základného formátu obrazovky v smere zvislej osi a *maska* je celé kladné číslo, ktoré určuje spôsob kreslenia prostredníctvom tzv. *bitovej masky.*

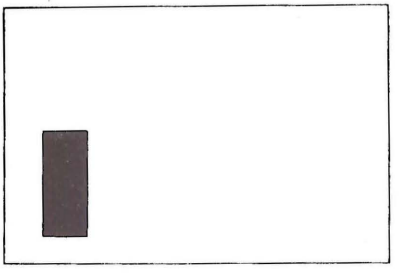
Použitie príkazu FILL a bitovej masky vysvetlíme na príkladoch.

Príklad 1.66

Úsek programu

```
1Ø REM PRIKAZ FILL
2Ø GCLEAR
3Ø SCALE Ø,1Ø,Ø,1Ø
4Ø MOVE 1,1
5Ø FILL 3Ø,1ØØ;1
```

vykreslí v ľavej dolnej časti obrazovky plochu obdĺžnika šírky 30 bodov a výšky 100 bodov základného formátu obrazovky (pozri obr. 1.19). Pozor: v príkaze MOVE, ktorým nastavíme „pero", používame súradnice v mierke zadanej príkazom SCALE, zatiaľ čo v príkaze FILL bez ohľadu na mierku v príkaze SCALE uvádzame rozsah kreslenia vždy vo formáte 256 × 243 bodov.



1.19 Kreslenie príkazom FILL 3Ø,1ØØ;1

Kreslenie príkazom FILL sa začína od bodu zadaného príkazom MOVE a pokračuje v smere zľava doprava a zdola nahor.

Vysvetlíme teraz, ako v príkaze FILL funguje bitová maska. Celé kladné desiatkové číslo od 0 do 63 môžeme v dvojkovej sústave vyjadriť pomocou 6 bitov (dvojkových číslic 0 a 1). Napríklad

$21 = 1.2^0 + 0.2^1 + 1.2^2 + 0.2^3 + 1.2^4 + 0.2^5$

Dvojkové vyjadrenie desiatkového čísla pomocou číslic 0 a 1, zapísané v poradí od najnižšej k najvyššej mocnine dvojky, nazývame bitovou maskou. Bitová maska, definovaná posledným parametrom

príkazu FILL, riadi spôsob zobrazovania takto:

Príkazom MOVE sa nastaví pero do určitého bodu obrazovky. Obdĺžnik šírky x bodov a výšky y bodov základného formátu sa môže príkazom FILL vykresliť 6-krát vedľa seba, pričom ak príslušná číslica v bitovej maske je 1, obdĺžnik sa vykreslí, zatiaľ čo pri číslici 0 sa na obrazovke vynechá nevykreslený priestor rovnakej veľkosti ako vykresľovaný obdĺžnik.

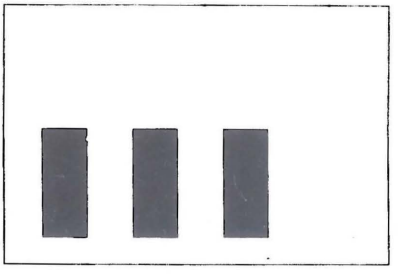
V príklade 1.66 sme v príkaze FILL použili v poslednom parametri desiatkové číslo 1, ktorého bitová maska je 100000. Preto sa zadaný obdĺžnik vykreslil iba raz od bodu určeného príkazom MOVE (na prvom mieste bitovej masky je 1, na ostatných miestach sú 0).

Príklad 1.67

Úsek programu

```
100 REM BITOVA MASKA
110 GCLEAR
120 SCALE 0,10,0,10
130 MOVE 1,1
140 FILL 30,100;21
```

vykreslí na obrazovke 3 obdĺžniky (pozri obr. 1.20) v súlade s bitovou maskou 101010 čísla 21.



1.20 Kreslenie príkazom FILL 30,100;21

Štandardným spôsobom kreslenia na PMD 85 a Didaktiku Alfa je negácia predchádzajúceho stavu. Kreslí sa „biele na čiernom", teda zobrazovaný bod sa „rozsvieti" na čiernom podklade obrazovky. Ak dvakrát kreslíme do toho istého bodu, pri druhom vykreslení sa tento bod „vymaže". Príkaz FILL môžeme využiť na tzv. inverziu obrazovky.

Príklad 1.68

Úsek programu

```
100 REM INVERZIA OBRAZOVKY
110 SCALE 0,255,0,242
120 MOVE 0,0
130 FILL 255,242;1
```

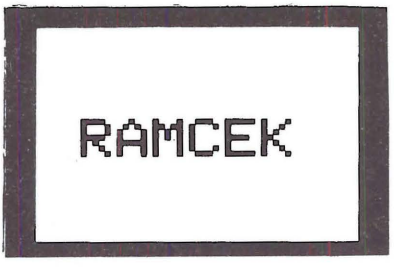
vykreslí obdĺžnik rozsahu celej pracovnej časti obrazovky a na základe kreslenia negáciou predchádzajúceho stavu spôsobí opačné zobrazenie (všetky „svietiace" body „zhasnú" a „nesvietiace" body sa „rozsvietia").

Príklad 1.69

Inverziu obrazovky môžeme využiť napríklad na vytvorenie rámčeka príkazom FILL. Postupujeme takto:

```
10 REM RAMCEK
20 GCLEAR
30 SCALE 0,255,0,242
40 MOVE 0,0
50 FILL 255,242;1
60 MOVE 20,20
70 FILL 215,202;1
80 MOVE 50,100
90 LABEL 4,5;"RAMCEK"
```

Úsek programu má tento účinok: Najprv sa vykreslí celá pracovná časť obrazovky. Druhý príkaz FILL potom vykreslí iba plochu na obrazovke od bodu (20, 20) v šírke 215 bodov a výške 202 bodov, teda vymaže vnútrajšok pracovnej časti obrazovky. Vznikne tak rámček v šírke 20 bodov na okrajoch obrazovky, do ktorého môžeme napríklad zapísať nejaký text, nakresliť graf alebo obrázok (pozri obr. 1.21).



1.21 Kreslenie rámčeka príkazom FILL

PROGRAM 1.12

Vo výpise 1.12 uvádzame program, ktorý nakreslí súradnicové osi a vyznačí na nich mierku (pozri obr. 1.22). Všimnime si, ako v tomto prípade funguje príkaz

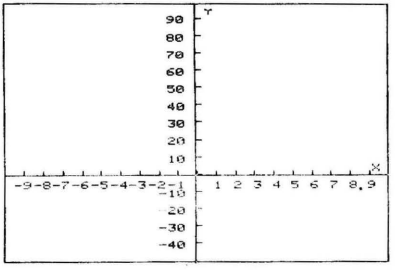
2ØØ FILL 1,4;1

```
10 REM PROGRAM 1.12 - SURADNICOVE OSI, 20. 4. 88
20 GCLEAR
30 SCALE -10,10,-50,100
130 AXES 0,0
140 MOVE 9.2,2
150 LABEL 1,1;"X"
160 MOVE 0.5,92
170 LABEL 1,1;"Y"
180 FOR I=-9 TO 9
190 MOVE I,0
200 FILL 1,4;1
210 IF I=0 THEN 240
220 MOVE I-0.5,-8
230 LABEL 1,1;I
240 NEXT I
250 FOR J=-40 TO 90 STEP 10
260 MOVE 0,J
270 FILL 4,1;1
280 IF J=0 THEN 310
290 MOVE -2,J-3
300 LABEL 1,1;J
310 NEXT J
320 END
```

Kreslí „obdĺžnik" šírky 1 bod a výšky 4 body, teda zvislú úsečku, ktorá vyznačuje mierku na osi x. Podobne príkaz

27Ø FILL 4,1;1

kreslí vodorovnú úsečku, ktorá vyznačuje mierku na osi y. Na vypísanie číselných hodnôt mierky používame príkaz LABEL.



1.22 Popisovanie súradnicových osí príkazom FILL

1.5.4 PRÍKAZY NA KRESLENIE V BAJTOVOM REŽIME

V bajtovom grafickom režime je pracovná oblasť obrazovky rozdelená na 48 bajtov (šestíc bodov) vo vodorovnom smere a na 243

riadkov v zvislom smere. Každý z 243 riadkov si môžeme predstaviť ako postupnosť 48 šestíc bodov. Zatiaľ čo v bodovom režime sa každý motív vykresľoval bod po bode, v bajtovom režime sa súčasne vykresľujú celé šestice bodov, čo podstatne zvyšuje rýchlosť kreslenia. Na nastavenie „pera" používame v bajtovom režime príkaz BMOVE, na kreslenie používame príkaz BPLOT.

a) Nastavenie „pera" príkazom BMOVE

Príkaz BMOVE slúži na nastavenie miesta na obrazovke, v ktorom sa má začať vykresľovať obrázok zadaný v príkaze BPLOT. Používa sa v tvare

```
BMOVE x,y
```

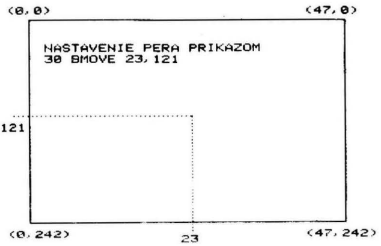
kde x je číslo od 0 do 47, ktoré určuje poradové číslo šestice bodov (bajtu) v riadku a y je celé číslo od 0 do 242, ktoré určuje poradové číslo riadka. Pozície bajtov, zadávané parametrom x, sú číslované zľava doprava, pozície riadkov zhora nadol (na rozdiel od príkazu MOVE bodového režimu, v ktorom sa riadky číslujú zdola nahor). Formát obrazovky je pevne zadaný a nemožno ho meniť (príkaz SCALE nemá vplyv na kreslenie v bajtovom režime).

Príklad 1.70

Príkaz

```
30 BMOVE 23,121
```

nastaví „pero" na začiatok 24. bajtu v 122. riadku obrazovky (pozri obr. 1.23).



1.23 Nastavenie pera príkazom BMOVE 23,121

b) Kreslenie príkazom BPLOT

Príkaz BPLOT slúži na vykresľovanie obrázka po šesticiach bo-

dov, pričom obrázok je definovaný reťazcom. Používame ho v tvare

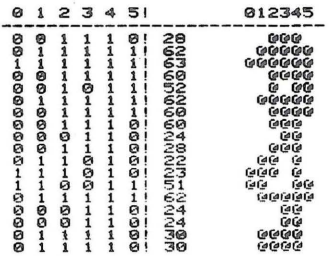
```
BPLOT reťazec,r
```

kde *reťazec* je reťazcová konštanta alebo premenná, ktorá definuje zobrazovaný motív, *r* je hodnota, ktorá určuje, koľko bajtov zobrazovaného reťazca sa môže zobraziť vedľa seba.

Podstatné je naučiť sa správne tvoriť reťazec, ktorý definuje zobrazovaný motív. Každý znak, ktorý sa v pamäti počítača zapíše do jedného bajtu, má nejakú číselnú hodnotu v ASCII kóde (od 0 do 255 desiatkovo). Túto hodnotu môžeme vyjadriť binárne pomocou 8 bitov (dvojkových číslic 0 a 1). Z binárneho vyjadrenia zoberieme iba prvých 6 bitov najnižších rádov, ktoré nazveme bitovou maskou (podobne ako v príkaze FILL). Bitová maska zodpovedá šestici zobrazovaných bodov, pričom ak na príslušnom mieste bitovej masky je číslica 1, potom zodpovedajúci bod „svieti" a v opačnom prípade (na príslušnom mieste bitovej masky je číslica 0) zodpovedajúci bod „nesvieti". Keď si pomocou bitovej masky pripravíme motív, ktorý chceme zobraziť, musíme vypočítať príslušné desiatkové hodnoty každej bitovej masky a tieto previesť na znaky a vložiť do reťazca. Nemusíme pritom poznať ASCII kód, stačí, keď použijeme funkciu CHR$, ktorá prevedie desiatkové číslo na zodpovedajúci znak.

Príklad 1.71

Na obr. 1.24 uvádzame jednoduchý motív figúrky. V pravej časti obrázka vidíme schematické znázornenie figúrky, v ľavej časti tabuľku čísel.



BITOVE VYJADRENIE MOTIVU FIGURKY

1.24 Motív figúrky

Naša figúrka zaberá 18 riadkov a v každom z nich 6 bodov grafického formátu obrazovky. Priestor, v ktorom je zobrazená, si teda môžeme predstaviť ako obdĺžnik 6 × 18 bodov. Každý z bodov tohto obdĺžnika buď „svieti", alebo „nesvieti" a „svietiace" body vytvárajú príslušný obrázok.

V tabuľke v ľavej časti obrázka sme na mieste bodov, ktoré majú „svietiť", zapísali jednotky a na ostatných miestach nuly. Stĺpce tabuľky sme označili 0, 1, 2, 3, 4, a 5. Každý riadok tabuľky má 6 prvkov a zodpovedá jednému bajtu, ktorý definuje príslušný motív. Stĺpce tabuľky vyjadrujú (zľava doprava) nultý, prvý, druhý až piaty bit príslušnej bitovej masky (spolu 6 bitov). Pomocou mocnín dvojky vynásobených dvojkovými číslicami 0 a 1 našej tabuľky vypočítame desiatkovú hodnotu každej bitovej masky.

Pripomenieme, že

$2^0 = 1, 2^1 = 2, 2^2 = 4, 2^3 = 8, 2^4 = 16, 2^5 = 32$

Napríklad hodnota prvého bajtu (pre prvý riadok tabuľky) je

$0.2^0 + 0.2^1 + 1.2^2 + 1.2^3 + 0.2^4 + 0.2^5 = 28$

(desiatkové hodnoty jednotlivých bajtov sú uvedené v poslednom stĺpci tabuľky).

Po výpočte desiatkových hodnôt treba tieto číselné hodnoty pomocou funkcie CHR$ previesť na znaky, a tým dostaneme reťazec, na základe ktorého príkazom BPLOT nakreslíme motív figúrky. Napríklad nasledujúci úsek programu nakreslí tento motív do stredu obrazovky (pozri obr. 1.25):

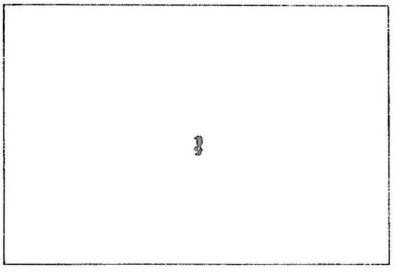
```
1Ø REM FIGURKA
2Ø GCLEAR
3Ø BMOVE 23, 121
4Ø FOR I=1 TO 18
5Ø READ K : T$=T$+CHR$(K)
6Ø NEXT I
7Ø BPLOT T$,1
1ØØ DATA 28,62,63,6Ø,52,62,6Ø,6Ø,24
11Ø DATA 28,22,23,51,62,24,24,3Ø,3Ø
```

Desiatkové čísla, z ktorých vytvárame reťazec T$ s motívom figúrky, sme uložili v príkazoch DATA. Príkazom

```
3Ø BMOVE 23,121
```

sme nastavili „pero" na začiatok 24. bajtu v 122. riadku obrazovky. Keďže šírka zobrazovaného motívu je 6 bodov, teda jeden bajt, uviedli sme v príkaze BPLOT za názvom reťazca ako druhý parameter šírku 1.

Analogickým spôsobom môžeme pripraviť a zobraziť akýkoľvek obrázok (v rozsahu grafického formátu obrazovky, t. j. 243 riadkov po 48 bajtov).



1.25 Zobrazenie motívu figúrky

PROGRAM 1.13

Na obr. 1.26 sme znázornili tabuľku na prípravu reťazca s motívom šípky. Naša šípka má šírku 18 bodov a výšku 7 bodov a smeruje

```
 0 1 2 3 4 5 0 1 2 3 4 5 0 1 2 3 4 5!
------------------------------------

 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0!   0   0   4
 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0!   0   0   8
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1!   0   0  16
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1!  63  63  63
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1!   0   0  16
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0!   0   0   8
 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0!   0   0   4
```

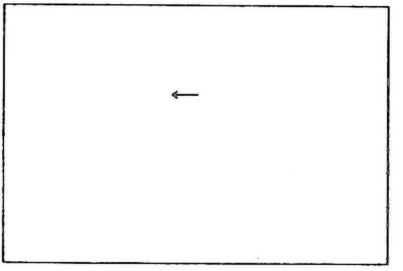
BITOVE VYJADRENIE MOTIVU SIPKY

1.26 Motív šípky

zľava doprava. Reťazec s motívom šípky teda obsahuje 7 riadkov po 3 bajty. V ľavej časti tabuľky uvádzame bitové vyjadrenie motívu, v pravej časti trojice desiatkových hodnôt pre jednotlivé bajty.

```
10 PROGRAM 1.13 - SIPKA, 10. 5. 88
20 GCLEAR
30 FOR I=1 TO 21
40 READ K
50 T$=T$+CHR$(K)
60 NEXT I
70 DISP "VLOZ RIADOK (10 AZ 230): " : INPUT R
80 DISP "VLOZ STLPEC (2 AZ 46): " : INPUT S
90 MOVE S,R
100 BPLOT T$,3
200 SCALE 0,255,0,242
210 MOVE 0,0
220 PLOT 255,0
230 PLOT 255,242
240 PLOT 0,242
250 PLOT 0,0
500 DATA 0,0,4
510 DATA 0,0,8
520 DATA 0,0,16
530 DATA 63,63,63
540 DATA 0,0,16
550 DATA 0,0,8
560 DATA 0,0,4
```

Program z výpisu 1.13 nakreslí šípku v zadanom mieste obrazovky (umiestnenie šípky volíme počas vykonávania programu). V príkaze BPLOT sme v tomto prípade ako druhý parameter použili šírku zobrazenia 3. Jednu zo zvolených polôh zobrazenia šípky uvádzame na obr. 1.27.



1.27 Zobrazenie motívu šípky

1.5.5 NIEKOĽKO UŽITOČNÝCH POZNATKOV

Ako sme už uviedli, kreslenie na PMD 85 a Didaktiku Alfa sa realizuje ako negácia predchádzajúceho stavu. Znamená to, že ak dvakrát kreslíme do toho istého bodu, pri druhom vykreslení sa tento bod „vymaže".

Príklad 1.72

Na určité miesto obrazovky vypíšeme príkazom LABEL text

KROK C. 1

a po určitom čase chceme na to isté mieto vypísať text

VYSLEDKY KROKU C. 1

Môžeme postupovať napr. takto:

```
100 SCALE 0,255,0,242
110 MOVE 10,220
120 LABEL 1,2;"KROK C. 1"

300 MOVE 10,220
310 LABEL 1,2;"KROK C. 1"
320 MOVE 10,220
330 LABEL 1,2;"VYSLEDKY KROKU C. 1"
```

Príkazmi v riadkoch 110 a 120 najprv vypíšeme prvý text. Opakovaným výpisom tohto textu v rovnakom mieste príkazmi v riadkoch 300 a 310 sa potom text vymaže a príkazmi v riadkoch 320 a 330 na to isté miesto vypíšeme nový text. Priamym prepísaním už vypísaného textu by sme na obrazovke získali nečitateľné znaky (podobne ako keď prepíšeme už napísaný text iným textom na písacom stroji).

Princíp „premazávania“ opätovným vykreslením môžeme niekedy s výhodou využiť pri práci s príkazmi PLOT a BPLOT na postupnú zmenu už vykreslených obrázkov alebo na programovanie jednoduchého pohybu po obrazovke.

PROGRAM 1.14

Výpis 1.14 obsahuje program na pohyb vykresleného obrázka po obrazovke (pre jednoduchosť pohybujeme obdĺžnikom zľava doprava).

```
10 REM PROGRAM 1.14 - POHYB, 11. 5. 88
20 DIM X(4),Y(4)
30 SCALE 0,255,0,242
40 GCLEAR
50 MOVE 50,160
60 LABEL 2,2;"POHYB  ==>"
70 MOVE 0,0
80 PLOT 255,0
90 PLOT 255,242
100 PLOT 0,242
110 PLOT 0,0
150 REM UDAJE PRE OBDLZNIK
160 FOR I=0 TO 3
170 READ X(I),Y(I)
180 NEXT I
190 X(4)=X(0) : Y(4)=Y(0)
200 REM POHYB OBDLZNIKA
210 FOR L=1 TO 8
220 J0=2 : IF L=8 THEN J0=1
230 REM NAKRESLENIE A VYMAZANIE OBDLZNIKA
240 FOR J=1 TO J0
250 MOVE X(0),Y(0)
260 FOR K=1 TO 4
270 PLOT X(K),Y(K)
280 NEXT K
290 IF J=1 THEN PAUSE 5
300 NEXT J
310 REM ZMENA UDAJOV O POLOHE OBDLZNIKA
320 FOR M=0 TO 4

330 X(M)=X(M)+20
340 NEXT M
350 NEXT L
360 END
400 REM UDAJE O VYCHODISKOVEJ POLOHE
410 DATA 20,20,70,20,70,100,20,100
```

Program funguje tak, že po vykreslení obdĺžnika a krátkej pauze opätovne vykreslí obdĺžnik v rovnakom mieste, a tým vymaže jeho starú polohu. Potom „posunie" obdĺžnik o 20 bodov základného formátu smerom vpravo, zobrazí jeho novú polohu, znovu ju vymaže a rovnakým spôsobom pokračuje ďalej. Pri svojom pohybe zľava doprava obdĺžnik postupne nadobúda 8 polôh.

Kreslenie „negáciou" nám poslúžilo na imitáciu pohybu. Negácia predchádzajúceho stavu, t. j. vymazávanie pri opätovnom kreslení na to isté miesto, však môže niekedy znehodnotiť kreslený obrázok. Možno tomu zabrániť použitím príkazu

```
POKE -15878,176
```

ktorý zmení režim kreslenia a zabezpečí, že už zobrazené body sa

pri prekreslení nebudú vymazávať. K pôvodnému režimu kreslenia negáciou sa potom môžeme vrátiť príkazom

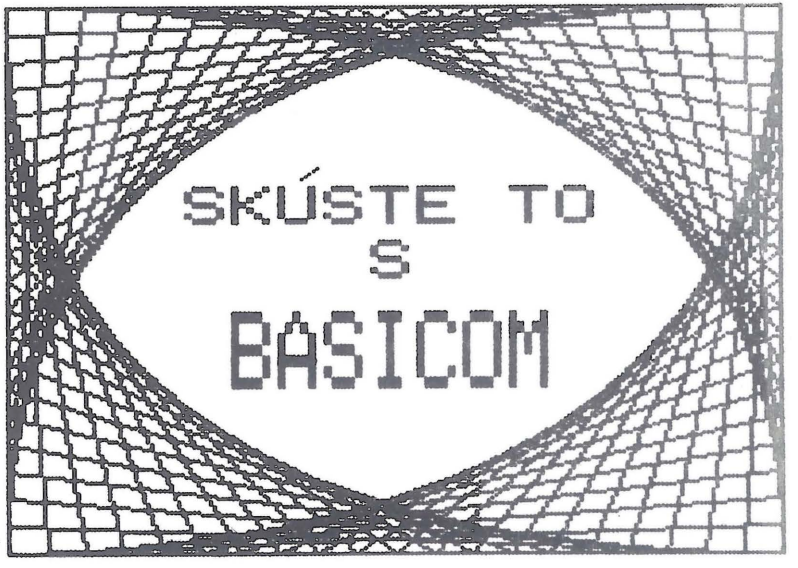
```
POKE -15878,168
```

PROGRAM 1.15

Program z výpisu 1.15 kreslí sieť, ktorá vytvára rámček s názvom tejto knihy. Použili sme v ňom príkaz na zrušenie kreslenia negáciou

```
10 REM PROGRAM 1.15 - HLAVICKA, 11. 5. 88
20 GCLEAR
25 POKE -15878,176
30 SCALE 0,254,0,243
40 MOVE 25,30
50 FOR I=0 TO 200 STEP 10
60 PLOT 225,30+I
70 PLOT 225-I,230
80 PLOT 25,230-I
90 PLOT 25+I,30
100 NEXT I
105 MOVE 100,165
106 LABEL 1,1;"/"
110 MOVE 70,150
120 LABEL 2,2;"SKUSTE TO"
130 MOVE 70,130
140 LABEL 2,2;"     S"
150 MOVE 70,90
160 LABEL 2,4;" BASICOM"
170 POKE -15878,168
```

(aby sa nedeformovala vykresľovaná sieť v bodoch, v ktorých sa pretínajú jednotlivé krivky). Na obr. 1.28 uvádzame výstup programu na obrazovku.



1.28 Výstup programu 1.15

Ukážeme si ešte, ako na PMD 85 a Didaktiku Alfa programovať svetelné efekty (úrovne jasu a blikanie).

Počítač PMD 85 umožňuje zobrazovať texty a obrázky v dvoch jasových úrovniach: tzv. polojasom a plným jasom. Plný jas je normálny režim zobrazovania, ktorý sa automaticky nastaví po zapnutí počítača. Na zobrazenie objektu s menšou intenzitou (polojasom) treba tento režim zmeniť príkazom

```
POKE -16326,Ø
```

Na zvýraznenie niektorých častí obrázkov alebo textov môžeme využiť aj efekt blikania (zobrazovaný objekt sa v krátkych intervaloch striedavo zobrazuje a mizne). Režim blikania objektu zobrazeného polojasom nastavíme príkazom

```
POKE -16326,128
```

a blikanie objektu zobrazeného plným jasom nastavíme príkazom

```
POKE -16326,192
```

Príslušný režim musíme nastaviť vždy pred prvým výstupným príkazom zobrazenia objektu, na ktorom sa má svetelný efekt prejaviť. K normálnemu režimu (zobrazovanie plným jasom bez blikania) sa vrátime príkazom

```
POKE -16326,64
```

Trocha iný účinok majú uvedené príkazy na počítači Didaktik Alfa. V pravej časti klávesnice Didaktiku Alfa sú dva špeciálne klávesy: jeden so znakom „slniečka" a druhý s vyšrafovaným kruhom. Účinok jednotlivých príkazov POKE závisí na Diďaktiku od toho, či pred vykonaním príkazu bolo stlačené „slniečko", alebo „vyšrafovaný kruh". Nasledujúca tabuľka uvádza jednotlivé účinky:

	vyšrafovaný kruh	slniečko
POKE -16326,Ø	polojas	polojas
POKE -16326,64	normálny jas	normálny jas
POKE -16326,128	zvýšený jas	zvýšený jas
POKE -16326,192	vysoký jas	blikanie

Po zapnutí počítača a privolaní interpreta basicu sa automaticky nastaví režim „vyšrafovaného kruhu" a normálneho jasu zobrazenia.

Poznámka 1.9

Na počítači PMD 85–2 máme navyše k dispozícii funkciu INK, ktorá riadi intenzitu a blikanie v príkazoch PRINT (pozri poznámku 1.6). Pre grafické príkazy možno okrem toho na PMD 85–2 používať príkaz

PEN výraz

kde *výraz* je aritmetický výraz, ktorého hodnota určuje režim zobrazovania pre grafické príkazy PLOT, AXES, LABEL a FILL. Výraz môže mať tieto hodnoty:

1	kreslenie polojasom,
2	kreslenie blikaním,
4	inverzné kreslenie (čierne na bielom),
8	kreslenie negáciou (kreslením sa neguje predchádzajúci stav),
16	kreslenie „mazaním" (maže sa obsah obrazovky).

Po zapnutí počítača sa nastavuje automaticky PEN 8, teda kreslenie plným jasom, bez blikania, biele na čiernom a režim negácie.

Kombinovanie rôznych režimov zobrazovania zabezpečujeme súčtami príslušných hodnôt. Ak nie je v súčte ani 8, ani 16, kreslí sa „nastavovaním" (t. j. body na obrazovke, v ktorých kreslíme, sa zobrazujú bez ohľadu na predchádzajúci stav). Ak je v súčte 8 aj 16, berie sa len 16. Ak nie je v súčte 4, kreslí sa biele na čiernom, ak je v súčte 4, kreslí sa inverzne (t. j. čierne na bielom). Napr. príkaz PEN 9 zabezpečí kreslenie polojasom a negáciou.

1.6 PROGRAMOVANIE VÝSTUPOV NA TLAČIAREŇ

K osobným počítačom PMD 85 a Didaktik Alfa možno pripojiť tlačiarne D–100 z PĽR a PRT–80 z MĽR. V tejto kapitole uvádzame všeobecné príkazy jazyka basic G na prácu s tlačiarňou (ktoré nezávisia od typu používanej tlačiarne) a ďalej niektoré špecifické riadiace znaky na regulovanie formátov tlače (ktoré závisia od vlastností konkrétnej tlačiarne).

1.6.1 PRÍKAZY VÝSTUPOV NA TLAČIAREŇ

Po zapojení tlačiarne do siete a pripojení k počítaču príslušným konektorom uvedieme tlačiareň do chodu vypínačom (tlačiareň D–100 ho má na ľavej strane a je zapnutý, ak je v polohe ON, tlačiareň PRT–80 má vypínač v zadnej stene a musí byť v polohe 1). Ak je tlačiareň pripravená na prácu, svieti signálne svetlo READY.

Aby sme mohli tlačiť výpisy programov alebo priame výstupy z programov, musíme najprv výstup na tlačiareň otvoriť príkazom CONTROL. Výpisy programov tlačíme povelom LIST a výstupy z programov programujeme príkazom OUTPUT.

a) Otvorenie výstupu na tlačiareň

Otvorenie výstupu programujeme príkazom

```
CONTROL 4,3;16Ø,13
```

Tento príkaz vkladáme buď v priamom režime, alebo v programovom režime. Za ním spravidla vkladáme príkaz

```
OUTPUT 4Ø3;
```

ktorým si overíme funkčnú schopnosť tlačiarne (jeho účinkom sa „vytlačí" prázdny riadok).

Príklad 1.73

Na začiatok programu, v ktorom požadujeme výstupy na tlačiareň, vložíme inicializačné príkazy

```
2Ø CONTROL 4,3;16Ø,13
3Ø OUTPUT 4Ø3;
```

b) Výpis programu

Na výpis programu uloženého vo vnútornej pamäti používame povel

```
LIST #4Ø3;čr
```

kde *čr* je číslo riadka, od ktorého požadujeme vytlačenie výpisu programu. Tento povel vkladáme v priamom režime a pred jeho prvým použitím musíme vložiť príkaz CONTROL. Ak v príkaze LIST #4Ø3; neuvedieme číslo riadka, vytlačí sa celý program od začiatku.

Príklad 1.74

Predpokladajme, že chceme vytlačiť program, ktorý sa nachádza vo vnútornej pamäti. Najprv vložíme príkaz

```
CONTROL 4,3;16Ø,13
```

a odošleme ho stlačením EOL. V dialógovom riadku sa po jeho úspešnom vykonaní objaví správa OK. Potom vložíme povel

```
LIST #4Ø3;
```

a odošleme ho stlačením EOL. Po ukončení tlače sa v dialógovom riadku objaví správa OK a môžeme pokračovať v ďalšej práci.

Tlačiareň D–100 má 2 KB pamäte (vyše 2 000 znakov), do ktorej sa po častiach prenáša vystupujúci program, takže správa OK

o ukončení prenosu sa môže zjaviť ešte pred ukončením tlače. V tom prípade môžeme v práci s počítačom pokračovať aj vtedy, ak tlačiareň ešte pracuje.

Ak chceme tlač ukončiť pred úplným vytlačením celého programu, stlačíme kláves STOP.

c) Programovanie výstupov na tlačiareň

Podobne ako výstupy na obrazovku (ktoré programujeme príkazom PRINT) môžeme programovať aj výstupy na tlačiareň. Na tieto účely používame príkaz

```
OUTPUT 4Ø3;zoznam
```

kde *zoznam* je zoznam vystupujúcich údajov, pre ktorý platia rovnaké pravidlá ako pre zoznam príkazu PRINT.

Príklad 1.75

Predpokladajme, že v programe chceme tlačiť nejaké výsledky výpočtov, uložené v dvojrozmernom poli A. Môžeme postupovať napr. takto:

```
1Ø REM PROGRAM S TLACOU VYSLEDKOV
2Ø CONTROL 4,3;16Ø,13
3Ø OUTPUT 4Ø3;
4Ø DIM A(2Ø,5)
   .
   .
   .
5ØØ REM TLAC VYSLEDKOV
51Ø OUTPUT 4Ø3;"VYSLEDKY:"
52Ø OUTPUT 4Ø3;"_________"
53Ø OUTPUT 4Ø3;
54Ø FOR I=1 TO 2Ø
55Ø FOR J=1 TO 5
56Ø OUTPUT 4Ø3;A(I,J);
57Ø NEXT J
58Ø OUTPUT 4Ø3;
59Ø NEXT J
```

Príkazom v riadku 20 otvárame výstup na tlačiareň. Úsek programu od riadku 500 vytlačí najprv na začiatku riadka text

VYSLEDKY:

a vynechá riadok. Ďalej tlačí maticu A po riadkoch (bodkočiarka

v zozname príkazu OUTPUT má rovnaký význam ako v príkaze PRINT, t. j. zabezpečuje vytlačenie ďalšieho vystupujúceho údaja do rovnakého riadka). Príkaz OUTPUT v riadku 580 zabezpečí prechod k novému riadku tlače.

V zozname príkazu OUTPUT môžeme využívať funkcie TAB a SPC na riadenie polohy tlače v riadku.

Príklad 1.76

Ak chceme v úseku programu z príkladu 1.75 zabezpečiť, aby stĺpce vystupujúcej matice boli vytlačené od rovnakých pozícií v riadku, môžeme postupovať napr. takto:

```
540 FOR I=1 TO 20
550 FOR J=1 TO 5
560 OUTPUT 403:TAB((J-1)*10);A(I,J);
570 NEXT J
580 OUTPUT 403;
590 NEXT I
```

Funkcia TAB v riadku 560 nastaví pred vytlačením každého prvku riadka matice tlačiareň do nultej (prvý prvok), desiatej (druhý prvok), dvadsiatej (tretí prvok) atď. pozície.

Normálny rozsah tlače v riadku je 80 znakov. Počet znakov tlačených v riadku možno zmeniť príkazom

```
POKE 1927,pz
```

kde *pz* je počet znakov, ktoré chceme vytlačiť v jednom riadku. Napr.

```
POKE 1927,50
```

zužuje rozsah tlače na 50 znakov v riadku.

Okrem vystupujúcich konštánt, premenných, ucelených aritmetických výrazov a funkcií TAB a SPC môžu byť súčasťou zoznamu v príkaze OUTPUT aj tzv. *riadiace znaky*, pomocou ktorých regulujeme spôsob tlače. Niektoré z riadiacich znakov sú spoločné pre rôzne tlačiarne, iné závisia od vlastností konkrétnej tlačiarne.

Medzi všeobecne použiteľné riadiace znaky patria nasledovné:

CHR$(10) vynechanie riadka,
CHR$(12) skok na novú stranu,
CHR$(13) skok na začiatok nasledujúceho riadka.

Príklad 1.77

Ak chceme v úseku programu z príkladu 1.75 zabezpečiť, aby sa záhlavie vystupujúcej matice vytlačilo na začiatku novej strany, zmeníme riadok 510 takto:

```
510 OUTPUT 403;CHR$(12);"VYSLEDKY"
```

Vynechanie riadka po podčiarknutí záhlavia možno zabezpečiť zmenou riadka 520 takto:

```
520 OUTPUT 403;"--------":CHR$(10);
```

Ušetríme tým riadok 530.

1.6.2 RIADENIE FORMÁTU TLAČE

Rôzne tlačiarne poskytujú rôzne možnosti riadenia formátov tlače. Tieto formáty regulujeme pomocou riadiacich znakov uvádzaných v zozname príkazu OUTPUT. Oboznámime sa so základnými formátmi tlače a ich riadením na tlačiarňach D–100 a PRT–80.

a) Formáty tlače a riadiace znaky pre tlačiareň D–100

Pri normálnom formáte tlače sa do jedného riadka vytlačí 80 znakov (vrátane medzier). Okrem toho možno pri práci s tlačiarňou D–100 využívať ďalšie formáty tlače, ktorými možno spestriť grafickú úpravu tlačených textov.

1. *Znaky dvojnásobnej šírky:* každý znak má rovnakú výšku ako pri normálnom formáte a dvojnásobnú šírku. V tomto prípade možno v riadku vytlačiť najviac 40 znakov.
2. *Znaky dvojnásobnej výšky:* každý znak má dvojnásobnú výšku a rovnakú šírku ako pri normálnom formáte; do riadka možno vytlačiť najviac 80 znakov a na stránku sa zmestí v porovnaní s normálnym formátom polovičný počet riadkov (spravidla 36, normálny formát 72).
3. *Znaky dvojnásobnej výšky a šírky:* každý znak má dvojnásobnú výšku a šírku v porovnaní s normálnym formátom; do riadka možno vytlačiť maximálne 40 znakov, na stranu 36 riadkov.
4. *Zúžené znaky:* do jedného riadka možno vytlačiť maximálne 132 zúžených znakov.
5. *Zdvojená tlač:* každý riadok sa vytlačí dvakrát, takže výsledná tlač je výraznejšia.
6. *Tlač s fázovým posunom:* každý znak sa vytlačí dvakrát, pričom pri jeho druhom vytlačení dôjde k malému posunu, takže výsledná tlač je výraznejšia v dôsledku nepatrného rozšírenia znakov.

Každý z týchto formátov sa nastavuje jedným alebo dvoma riadiacimi znakmi uvádzanými pomocou funkcie CHR$ v zozname príslušného príkazu OUTPUT. Pôsobnosť zadaného formátu sa zruší vložením riadiacich znakov ukončenia formátu, účinkom ktorých tlačiareň prejde k normálnemu formátu. Riadiace znaky pre tlačiareň D–100 uvádzame v tabuľke 1.2.

Tabuľka 1.2
Riadiace znaky formátov tlače pre tlačiareň D–100

formát	riadiace znaky nastavenia	riadiace znaky ukončenia
dvojnásobná šírka	CHR$(14);	CHR$(2Ø);
dvojnásobná výška	CHR$(27)+CHR$(91);	CHR$(27)+CHR$(51);
dvojnásobná výška, šírka	CHR$(27)+CHR$(92);	CHR$(27)+CHR$(51);
zúžené znaky	CHR$(15);	CHR$(18);
zdvojená tlač	CHR$(27)+CHR$(69);	CHR$(27)+CHR$(7Ø);
fázový posun	CHR$(27)+CHR$(71);	CHR$(27)+CHR$(72);

Príklad 1.78

Predpokladajme, že v úseku programu z príkladu 1.75 chceme zabezpečiť, aby sa záhlavie vystupujúcej matice vytlačilo na začiatku novej strany od tretieho riadka znakmi dvojnásobnej výšky a šírky (samotnú maticu tlačíme normálnym formátom). Program treba zmeniť takto:

```
5ØØ REM TLAC VYSLEDKOV
51Ø OUTPUT 4Ø3;CHR$(12);CHR$(1Ø);
52Ø OUTPUT 4Ø3;CHR$(27)+CHR$(92);"VYSLEDKY:"
53Ø OUTPUT 4Ø3;"--------";CHR$(27)+CHR$(51);
535 OUTPUT 4Ø3;CHR$(1Ø);
```

Niektoré z formátov tlače možno kombinovať. Napríklad zdvojenú tlač možno kombinovať so všetkými formátmi okrem formátu tlače s fázovým posunom.

Príklad 1.79

Predpokladajme, že v úseku programu z príkladu 1.75 chceme okrem úprav vykonaných v príklade 1.78 zabezpečiť zdvojenú (t. j. výraznejšiu) tlač záhlavia aj vystupujúcej matice. Príslušný úsek programu upravíme takto:

```
500 REM TLAC VYSLEDKOV
510 OUTPUT 403;CHR$(12);CHR$(10);
515 OUTPUT 403;CHR$(27)+CHR$(69);
520 OUTPUT 403;CHR$(27)+CHR$(92);"VYSLEDKY:"
530 OUTPUT 403:"--------";CHR$(27)+CHR$(51);
535 OUTPUT 403;CHR$(10);
540 FOR I=1 TO 20
550 FOR J=1 TO 5
560 OUTPUT 403;TAB((J-1)*10);A(I,J);
570 NEXT J
580 OUTPUT 403;
590 NEXT J
600 OUTPUT 403;CHR$(27)+CHR$(70);
```

V príkaze 515 sme nastavili formát zdvojenej tlače, ktorý zostane v platnosti až do vykonania príkazu 600, v ktorom ho zrušíme príslušnými riadiacimi znakmi.

PROGRAM 1.16

Výpis 1.16 obsahuje program na vytlačenie pozvánky na stretnutie krúžku mladých programátorov. Program je určený pre tlačiareň

```
10 REM PROGRAM 1.16 - POZVANKA, 11. 5. 88
100 CONTROL 4,3;160,13 ! OUTPUT 403;
200 REM TLAC POZVANOK
210 DISP "KOLKO POZVANOK CHCETE VYTLACIT? "
220 INPUT LP
240 FOR I=1 TO LP
250 IF I-INT(I/2)*2)0 THEN OUTPUT 403;CHR$(12);
260 OUTPUT 403;
270 OUTPUT 403;"KRUZOK MLADYCH PROGRAMATOROV PRI PIONIERSKEJ"
280 OUTPUT 403;"SKUPINE"
290 OUTPUT 403;
300 OUTPUT 403;
310 OUTPUT 403;"VAS POZYVA NA OSME STRETNUTIE NA TEMU:"
320 OUTPUT 403;
330 OUTPUT 403;CHR$(27)+CHR$(91);
340 OUTPUT 403;"      KRESLENIE NA OBRAZOVKE NA PMD 85"
350 OUTPUT 403;CHR$(27)+CHR$(51);
360 OUTPUT 403;
370 OUTPUT 403;
380 OUTPUT 403;"KTORE SA USKUTOCNI V PONDELOK 31. OKTOBRA 1988"
390 OUTPUT 403;"O 14,30 V DOME PIONIEROV. "
400 OUTPUT 403;
410 OUTPUT 403;
420 OUTPUT 403;"V BRATISLAVE 25.10.1988"
430 OUTPUT 403;
440 OUTPUT 403;
450 OUTPUT 403;
460 OUTPUT 403;"                                VEDUCI KRUZKU"
470 IF I-INT(I/2)*2=0 THEN 490
480 FOR J=1 TO 10 ! OUTPUT 403; ! NEXT J
490 NEXT I
500 END
```

D–100 (využíva jej špecifické formáty tlače). Po odštartovaní programu si môže používateľ zadať počet vytlačených pozvánok. Na jeden list tabelačného papiera sa tlačia vždy dve pozvánky. Ukážku vytlačenej pozvánky uvádzame na obr. 1.29.

```
KRUZOK MLADYCH PROGRAMATOROV PRI PIONIERSKEJ
SKUPINE

VAS POZYVA NA OSME STRETNUTIE NA TEMU:

     KRESLENIE NA OBRAZOVKE NA PMD 85

KTORE SA USKUTOCNI V PONDELOK 31. OKTOBRA 1988
O 14,30 V DOME PIONIEROV.

V BRATISLAVE 25.10.1988

                                VEDUCI KRUZKU
```

1.29 Výstup programu 1.16 na tlačiarni D–100

b) Formáty tlače a riadiace znaky pre tlačiareň PRT–80

Tlačiareň PRT–80 poskytuje používateľovi bohatšiu ponuku možností riadenia formátov tlače ako tlačiareň D–100. Ďalej uvádzame prehľad niektorých zaujímavých formátov tlače a možností ich kombinovania.

1. *Normálny formát:* znaky typu PICA, maximálne 80 znakov v riadku, 72 riadkov na stranu tabelačného papiera (12 palcov, 6 riadkov na palec, palec je zhruba 2,54 cm), súvislá tlač bez ohľadu na prechod na novú stranu.

2. *Zúžené znaky typu PICA:* do jedného riadka možno vytlačiť maximálne 132 znakov, výška je rovnaká ako v normálnom formáte.

3. *Zhustené znaky typu ELITE:* do jedného riadka možno vytlačiť maximálne 106 znakov, výška znakov je rovnaká ako v normálnom formáte.

4. *Znaky dvojnásobnej šírky:* každý znak má rovnakú výšku ako pri normálnom formáte a dvojnásobnú šírku. V dvojnásobnej šírke možno tlačiť znaky všetkých troch vyššie uvedených formátov; pre dvojnásobnú šírku normálneho formátu maximálne 40 znakov v riadku, pre dvojnásobnú šírku zúžených znakov PICA maximálne 66 znakov v riadku, pre dvojnásobnú šírku zhustených znakov ELITE maximálne 53 znakov v riadku.

5. *Horizontálne zvýraznené znaky:* ekvivalentné tlači s fázovým posunom na D–100. Každý riadok sa tlačí dvakrát, pričom pri druhej tlači sa všetky znaky posunú o 1/10 svojej šírky vpravo. Horizontálne možno zvýrazňovať iba znaky v normálnom formáte a v normálnom formáte dvojnásobnej šírky.

6. *Vertikálne zvýraznené znaky:* každý riadok sa tlačí dvakrát, pričom pri druhej tlači sa celý riadok posunie o 0,2 mm smerom dole. Vertikálne možno zvýrazňovať znaky všetkých formátov.

7. *Tlač s podčiarknutím:* tlačený text sa automaticky podčiarkne.

8. *Nastavenie riadkovania:* zmena normálneho režimu riadkovania na režim n/120, kde n je počet palcov na 120 riadkov. Dĺžka štandardného listu tabelačného papiera je 12 palcov, teda napr. pre n = 24 dostaneme riadkovanie 60 riadkov na stranu (5 riadkov na palec).

9. *Nastavenie stránkovania:* zadanie počtu riadkov na stranu.

10. *Nastavenie ľavého okraja:* zadanie pozície, od ktorej sa začína tlač v riadku.

11. *Nastavenie pravého okraja:* zadanie pozície, v ktorej sa končí tlač v riadku.

12. *Preskočenie perforácie na konci stránky.*

Na rozdiel od tlačiarne D–100 nie je na PRT–80 k dispozícii formát znakov dvojnásobnej výšky.

Riadiace znaky pre jednotlivé formáty uvádzame v tabuľke 1.3. Ak nie sú uvedené riadiace znaky ukončenia, znamená to, že príslušný formát stráca platnosť nastavením riadiacich znakov nového formátu.

Príklad 1.80

Prepíšeme úsek programu z príkladu 1.79, ktorý predpokladá použitie tlačiarne D–100, aby plnil analogické funkcie pri výstupe na tlačiareň PRT–80. Keďže nemáme k dispozícii znaky dvojnásobnej výšky, vytlačíme záhlavie normálnymi znakmi PICA dvojnásobnej šírky s horizontálnym zvýraznením a samotnú maticu normálnymi znakmi s horizontálnym zvýraznením.

```
500 REM TLAC VYSLEDKOV
510 OUTPUT 403;CHR$(12);CHR$(10);
520 OUTPUT 403;CHR$(27);"!";CHR$(40);"VYSLEDKY:"
530 OUTPUT 403;"---------"
535 OUTPUT 403;CHR$(10);CHR$(27);"!";CHR$(8);
540 FOR I=1 TO 20
550 FOR J=1 TO 5
560 OUTPUT 403;TAB((J-1)*10);A(I,J);
570 NEXT J
580 OUTPUT 403;
590 NEXT I
600 OUTPUT 403;CHR$(27);"!";CHR$(0);
```

V riadku 600 nastavujeme normálny formát tlače, čím sa ukončí formát horizontálne zvýraznených znakov.

Vzhľadom na to, že všetky znaky majú v rôznych formátoch rovnakú výšku, pri využívaní tlačiarne PRT–80 možno voliť rôzne formáty v tom istom riadku.

1.7 PRÁCA S OSOBNÝMI POČÍTAČMI ATARI 800/XL A 130/XE

V domácnostiach aj v krúžkoch mladých programátorov sa stretávame so značným počtom 8-bitových osobných počítačov Atari,

Tabuľka 1.3 Riadiace znaky formátov tlače pre tlačiareň PRT-80

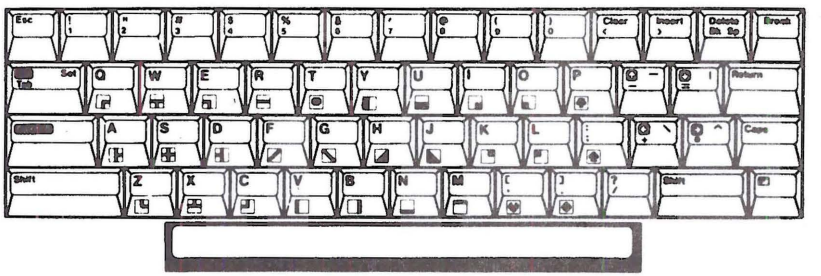
formát	riadiace znaky nastavenia riadiace znaky ukončenia poznámka
normálny PICA	CHR$(27);"!";CHR$(Ø); nastavuje sa automaticky po zapnutí tlačiarne
zúžené znaky PICA	CHR$(27);"!";CHR$(4);
zhustené znaky ELITE	CHR$(27);"!";CHR$(1);
dvojnásobná šírka	
normál PICA	CHR$(27);"!";CHR$(32);
zúžené PICA	CHR$(27);"!";CHR$(36);
zhustené ELITE	CHR$(27);"!";CHR$(33);
horizontálne zvýraznenie	
normál PICA	CHR$(27);"!";CHR$(8);
dvojnásobná šírka	CHR$(27);"!";CHR$(4Ø);
vertikálne zvýraznenie	
normál PICA	CHR$(27);"!";CHR$(16);
zúžené PICA	CHR$(27);"!";CHR$(2Ø);
zhustené ELITE	CHR$(27);"!";CHR$(17);
podčiarknutie	CHR$(27);"-";CHR$(1); CHR$(27);"-";CHR$(2); treba ukončiť
nastavenie riadkovania	CHR$(27);"J";CHR$(n); kde 0 ≦ n ≦ 255
nastavenie stránkovania	CHR$(27);"C";CHR$(n); kde 1 ≦ n ≦ 127, n je počet riadkov na stranu
ľavý okraj	CHR$(27);"I";CHR$(n); začiatok tlače od n-tej pozície zľava
pravý okraj	CHR$(27);"Q";CHR$(n); koniec tlače v n-tej pozícii riadka
preskočenie perforácie	CHR$(27);"N";CHR$(n); preskočenie na novú stranu n riadkov pred ukončením strany CHR$(27);"O"; treba ukončiť

ktoré sa k nám dovážajú prostredníctvom podniku zahraničného obchodu TUZEX. Ide o spoľahlivé a výkonné počítače s dokonalou farebnou grafikou a nástrojmi na spracovanie zvukov, ktoré sú (na rozdiel od iných u nás dostupných osobných počítačov) k dispozícii so všetkými základnými prídavnými zariadeniami: kazetovým záznamníkom programov Atari XC12, maticovou tlačiarňou Atari 1029, jednotkou pamäte na pružných diskoch Atari 1050 a ovládačom hier (joystick). Počítač Atari 800/XL má vnútornú pamäť RAM 64 KB a Atari 130/XE má vnútornú pamäť 128 KB.

V tejto kapitole uvádzame stručný prehľad základných poznatkov o práci s počítačmi Atari v basicu, pričom sa zameriavame predovšetkým na rozdiely medzi dialektom basicu pre PMD 85 a Didaktik Alfa (basicom G) a Atari basicom.

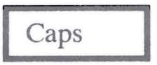
1.7.1 KLÁVESNICA A OBRAZOVKA

Klávesnica počítača Atari 130/XE (pozri obr. 1.30) má 57 klávesov. Ich usporiadanie je rovnaké ako na počítači Atari 800/XL. V ľavej a pravej časti klávesnice sú tzv. ovládacie klávesy, stredná časť obsahuje alfanumerické klávesy a klávesy so špeciálnymi znamienkami. Väčšina alfanumerických klávesov umožňuje vkladať 3 znaky: veľké písmeno, malé písmeno a grafický znak. Po zapnutí počítača písmenové klávesy vkladajú veľké písmená. Klávesy s horným a dolným znakom vkladajú pri stlačení dolný znak, horný znak vložíme súčasným stlačením príslušného dvojznakového klávesu a klávesu



1.30 Klávesnica počítača Atari 130/XE

Stlačením klávesu



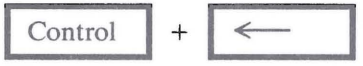
v pravej časti klávesnice nastavíme režim malých a veľkých písmen. Stlačením písmenového klávesu potom vkladáme malé písmeno a súčasným stlačením klávesu *Shift* a písmenového klávesu vkladáme veľké písmeno. V tomto režime zostáva bez zmeny práca s klávesmi obsahujúcimi horný a dolný znak (*Shift* + dvojznakový kláves generuje horný znak). Režim veľkých a malých písmen zrušíme opätovným stlačením klávesu *Caps*.

Na prednej strane väčšiny alfanumerických klávesov sú vyznačené grafické znaky (napr. obdĺžničky, štvorčeky a i.). Tieto znaky vkladáme súčasným stlačením klávesu

na ľavej strane klávesnice a príslušného klávesu. Spolu máme k dispozícii 29 klávesov s rôznymi grafickými znakmi, pomocou ktorých môžeme vytvárať jednoduché obrázky.

Ak držíme stlačený kláves, vkladanie znaku sa opakuje, kým kláves neuvoľníme.

Kláves *Control* spolu s ďalšími ovládacími klávesmi sa používa na riadenie pohybu kurzora po obrazovke a editovanie vkladaného textu. Štyri klávesy v pravej časti klávesnice obsahujú znaky „šípka vľavo", „šípka vpravo", „šípka hore" a „šípka dole". Súčasným stlačením klávesov



(šípka vľavo)

posunieme kurzor o jednu pozíciu vľavo atď. Kurzor môžeme týmto spôsobom (stláčaním Control a príslušnej šípky) nastavovať po celej ploche obrazovky. Kurzor sa zobrazuje ako malý biely štvorček. Pri pohybe po obrazovke kurzor nepremazáva znaky, cez ktoré sa posúva. Stlačením klávesov

vymažeme obrazovku.

Na editovanie programov a vkladaných textov sa využívajú spolu s klávesom *Control* klávesy *Insert, Delete, BkSp* a *Shift*. Súčasným stlačením klávesov

Control + Insert

sa na miesto, na ktorom je nastavený kurzor, vloží medzera pre jeden znak (celý text, počínajúc znakom, na ktorý je nastavený kurzor, sa posunie o jednu pozíciu doprava). Stlačením klávesov

Shift + Insert

vložíme nový prázdny riadok nad riadkom, na ktorý je nastavený kurzor.

Súčasným stlačením klávesov

Control + Delete

sa vymaže znak, na ktorom je nastavený kurzor, a celý text vpravo od kurzora sa posunie o jednu pozíciu doľava. Ak stlačíme iba

Delete

posunie sa kurzor o jednu pozíciu doľava a vymaže znak (vloží namiesto neho medzeru). Stlačením klávesov

Shift + Delete

sa vymaže celý riadok, na ktorom je nastavený kurzor.

Na tabelovanie slúži kláves

Tab

Po jeho prvom stlačení sa kurzor posunie o 5 pozícií doprava, pri každom ďalšom stlačení o 8 pozícií.

Stlačením klávesu ◩ v pravom dolnom rohu klávesnice (tzv. *Inverse video*) dosiahneme inverzné zobrazenie. Po jeho stlačení sa všetky vkladané znaky zobrazujú inverzne (pri čiernobielej obrazovke „čierne na bielom", zatiaľ čo normálne zobrazenie je „biele na čiernom"). K normálnemu zobrazovaniu sa vrátime opätovným stlačením klávesu ◩

Stlačením klávesu

Break

v pravom hornom rohu klávesnice ukončíme práve vykonávanú činnosť (beh programu, výpis na obrazovku). Klávesom

Return

v pravej časti klávesnice odosielame do pamäte počítača vkladané povely, príkazy a údaje (podobne ako EOL na počítačoch PMD 85).

V pravej časti panelu počítača nad vlastnou klávesnicou (na Atari 800/XL napravo od klávesnice) máme ešte 5 väčších tlačidiel: *Help, Start, Select, Option* a *Reset*. Prvé štyri sa používajú ako funkčné klávesy, ktorých účinok možno naprogramovať. Tlačidlom

Reset

znovunaštartujeme interpret basicu (napr. pri strate kontroly nad činnosťou počítača).

Obrazovka je v základnom alfanumerickom formáte rozdelená do 24 riadkov po 40 znakov. Riadky adresujeme od 0 do 23 zhora dole, stĺpce od 0 do 39 zľava doprava. Atari basic umožňuje pracovať s farebnou obrazovkou v 16 rôznych formátoch a využívať 16 základných farieb, z ktorých možno vytvoriť rôznymi odtieňmi ďalších 112 farieb. Základný alfanumerický formát nastavuje automaticky tmavomodrú farbu pozadia, svetlomodrú farbu popredia (textu) a čierne orámovanie obrazovky. Pri práci s čiernobielou obrazovkou máme čierne pozadie a bielu farbu popredia (zobrazovanie textu). Príkazmi basicu možno farby pozadia, popredia a okrajov obrazovky meniť.

1.7.2 POVELY INTERPRETA JAZYKA ATARI BASIC

Po zapnutí počítača sa automaticky privolá interpret Atari basicu a na prvom riadku obrazovky sa zjaví správa

READY

pričom kurzor sa nastaví na začiatok nasledujúceho riadka. Počítač je pripravený na prijímanie a vykonávanie povelov a príkazov.

Povely NEW, LIST, RUN a CONT sa používajú rovnako alebo podobne ako na PMD 85 a Didaktik Alfa. Povelom

NEW

vymažeme pamäť počítača pred začiatkom vkladania nového programu. Povel

LIST

vypisuje na obrazovku program uložený vo vnútornej pamäti. Časť programu vypíšeme povelom

```
LIST čr1,čr2
```

kde $čr_1$ je číslo prvého programového riadka a $čr_2$ je číslo posledného programového riadka úseku programu, ktorý chceme vypísať. Povel

```
LIST čr
```

vypíše iba riadok číslo *čr*. Povelom

```
CONT
```

obnovíme beh programu prerušeného stlačením klávesu Break.

V porovnaní s PMD 85 a Didaktikom Alfa chýba povel AUTO na automatické číslovanie vkladaných programových riadkov a povel LLIST na výpis do dialógového riadka. Atari basic nepracuje s dialógovým riadkom a programy možno editovať po celej ploche obrazovky.

Väčšie rozdiely sú v práci s programovými súbormi na kazetách. Program zapíšeme z vnútornej pamäte na kazetu povelom

```
CSAVE
```

Na kazetovom záznamníku najprv nastavíme kazetu za posledný záznam (orientujeme sa pomocou počítadla otáčok) stlačením tlačidla F.FWD (pohyb dopredu) alebo REWIND (pohyb dozadu). Pohyb pásky zastavíme stlačením tlačidla STOP. Vložíme príkaz CSAVE a stlačíme kláves *Return*. Na záznamníku stlačíme tlačidlá PLAY a RECORD a znovu stlačíme kláves *Return*. Tým sa automaticky zapne motor záznamníka a začne sa zápis programu na kazetu. Počas zápisu svieti červené kontrolné svetlo SAVE v pravej časti horného panelu záznamníka. Po ukončení zápisu sa na obrazovke zjaví správa

READY

a záznamník sa zastaví. Nezabudnite si pred zápisom na kazetu poznačiť polohu programového súboru na kazete podľa údaja na počítadle otáčok (na rozdiel od PMD 85 a Didaktiku Alfa Atari basic nevie hľadať na kazete podľa poradového čísla).

Z kazety prepíšeme program do vnútornej pamäte povelom

```
CLOAD
```

Najprv nastavíme záznamník na začiatok súboru, vložíme povel CLOAD a stlačíme kláves *Return*. Po zvukovom signále („pípnutí") stlačíme na záznamníku tlačidlo PLAY a na klávesnici ešte raz *Return*. Po ukončení čítania sa na obrazovke zjaví správa

READY

a záznamník sa zastaví.

Interpret Atari basicu umožňuje spájať program vo vnútornej pamäti s programovým súborom na kazete. Ak programový súbor zapíšeme na kazetu povelom CSAVE, potom pri jeho načítaní do vnútornej pamäte povelom CLOAD sa program, ktorý sa nachádza vo vnútornej pamäti, vymaže. Ak však program zapíšeme na kazetu povelom

```
LIST "C:"
```

môžeme ho z kazety do vnútornej pamäte prepísať povelom

```
ENTER "C:"
```

a v tomto prípade sa program z kazety pripojí k programu, ktorý je už vo vnútornej pamäti. Vstup z kazety prebieha podľa rovnakých pravidiel ako vstup z klávesnice (riadky programu z kazety sa zaradia do programu vo vnútornej pamäti podľa poradových čísel, pričom riadky programu z kazety prepíšu rovnako očíslované riadky vo vnútornej pamäti).

Na zápis programov na kazetu a čítanie programov z kazety môžeme použiť aj povely

```
SAVE "C:"
```

a

```
LOAD "C:"
```

ktoré fungujú rovnako ako povely CSAVE a CLOAD.

Ak máme pripojenú tlačiareň, vypíšeme program z vnútornej pamäte na tlačiareň povelom

```
LIST "P:"
```

V prípade, že máme pripojenú disketovú jednotku a pracujeme pod riadením operačného systému DOS 2.5, zapíšeme program na disketu povelom

```
SAVE "D1:názov.BAS"
```

kde *názov* je maximálne 8-znakový názov, ktorý pridelíme programovému súboru, a BAS je štandardná prípona súboru s programom v basicu. Z diskety do vnútornej pamäte načítame program povelom

```
LOAD "D1:názov.BAS"
```

kde *názov* je názov, pod ktorým bol programový súbor zapísaný na disketu.

Na rozdiel od PMD 85 a Didaktiku Alfa nemá interpret Atari basicu povel na kontrolu správnosti záznamov na kazete, analogický povelu CHECK. Záznam na kazetu je však na počítačoch Atari podstatne spoľahlivejší.

1.7.3 SPOLOČNÝ ZÁKLAD

Väčšina poznatkov z kapitoly 1.3 o základných prvkoch jazyka basic zostáva v platnosti aj pre Atari basic.

Programové riadky sa číslujú od 0 do 32767 (ako na počítači PMD 85–2). Maximálna dĺžka programového riadka je 114 znakov (necelé tri riadky na obrazovke). Mená premenných a polí sa tvoria z veľkých písmen a číslic, pričom prvým znakom mena musí byť písmeno. Dĺžka mena premennej alebo poľa je ohraničená iba dĺžkou programového riadka. Mená reťazcových premenných sa končia znakom $ (dolár).

Atari basic umožňuje väčšiu presnosť výpočtov. Môžeme pracovať s číslami, ktorých absolútna hodnota je v rozsahu od 10^{-99} do 10^{99} s presnosťou na 9 miest. Reťazcová premenná môže obsahovať najviac 32767 znakov. Môžeme pracovať s jedno- alebo dvojrozmernými číselnými poľami. Zvláštnosti práce s reťazcami vysvetľujeme v odseku 1.7.4.

Zostávajú zachované všetky pravidlá pre prácu s aritmetickými a logickými výrazmi. V Atari basicu sú k dispozícii tieto štandardné funkcie: ABS, SQR, INT, SGN, LOG, CLOG (dekadický logaritmus), RND, EXP, SIN, COS, TAN, ATN, ASC, CHR$, LEN, STR$, VAL, PEEK, USR a FRE. Používajú sa rovnakým spôsobom ako na PMD 85 a Didaktiku Alfa.

Bez podstatných zmien zostávajú aj štandardné príkazy. Príkazy PRINT, DATA, READ, RESTORE, DIM (pre číselné polia),

GOTO, IF THEN, ON GOTO, FOR TO STEP, NEXT, INPUT, LET, GOSUB, RETURN, ON GOSUB, STOP, REM, POKE a END sa v Atari basicu používajú rovnako alebo takmer rovnako ako v basicu G. V zozname príkazu INPUT môžeme používať iba jednoduché premenné. V zozname príkazu PRINT nemôžeme používať funkcie TAB a SPC. Náhradu nám poskytuje príkaz

```
POSITION j,i
```

kde *j* je číslo stĺpca a *i* je číslo riadka obrazovky. Tento príkaz nastavuje polohu výpisu na obrazovke. Príkaz GCLEAR môžeme nahradiť príkazom

```
PRINT CHR$(125);
```

a príkaz BEEP príkazom

```
PRINT CHR$(253);
```

Príkaz CLEAR (vynulovanie premenných) nahradíme príkazom

```
CLR
```

Príkaz ON ERR, ktorý je k dispozícii na PMD 85–2, nahradíme v Atari basicu príkazom

```
TRAP čr
```

kde *čr* je číslo programového riadka, ktorému sa má odovzdať riadenie v prípade výskytu chyby.

Pri práci s interpretom Atari basicu máme na programy a údaje k dispozícii 37 KB vnútornej pamäte (na PMD 85 a Didaktiku Alfa iba 19 KB).

Ďalej podrobnejšie vysvetlíme tie prvky, v ktorých sa Atari basic výraznejšie odlišuje od basicu G: príkazy a funkcie na prácu s reťazcami, farebnou grafikou a spracovanie zvukov.

1.7.4 PRÁCA S REŤAZCAMI

Každú reťazcovú premennú musíme v Atari basicu deklarovať príkazom DIM, v ktorom uvedieme maximálny počet znakov reťazca.

Príklad 1.81

Príkazy

```
100 DIM Z$(500),P$(300)
110 DIM Q0$(1)
```

deklarujú tri reťazcové premenné: Z$ s maximálnou dĺžkou 500 znakov, P$ s maximálnou dĺžkou 300 znakov a Q0$ s dĺžkou 1 znak.

Príklad 1.82

Reťazcovú premennú vymedzerujeme takto:

```
100 DIM T$(200)
110 T$(1)=" " : T$(200)=" "
120 T$(2)=T$
```

Po vykonaní týchto príkazov sa reťazec T$ naplní 200 medzerami.

Uvedením názvu reťazca a začiatočnej a koncovej pozície nejakej časti reťazca môžeme z reťazcov vyberať podreťazce (nepotrebujeme teda funkcie LEFT$, RIGHT$ a MID$, používané v basicu G).

Príklad 1.83

V úseku programu

```
100 DIM M$(10) : M$="DOBRY DEN "
110 PRINT M$(1,6)
120 PRINT M$(7,10)
```

najprv naplníme reťazec M$ textom

DOBRY DEN

a potom z reťazca vyberieme a vypíšeme na obrazovku jednotlivé slová: príkazom

```
110 PRINT M$(1,6)
```

slovo DOBRY a príkazom

```
120 PRINT M$(7,10)
```

slovo DEN.

Ukážeme ďalej, ako programovať vstup reťazcov príkazom INPUT a nahradiť pritom možnosť práce s poľami.

Príklad 1.84

Predpokladajme, že chceme v reťazcovej premennej Z$ vytvoriť zoznam žiakov vstupom z klávesnice. Každý záznam (priezvisko, meno) môže obsahovať najviac 20 znakov a zoznam najviac 25 záznamov. Potrebujeme na to reťazec rozsahu 25 × 20 t. j. 500 znakov. Môžeme postupovať napr. takto:

```
10 REM Naplnenie zoznamu
```

```
20 DIM Z$(500)
30 Z$(1)=" " : Z$(500)=" "
40 Z$(2)=Z$
50 DIM MENO$(20)
60 DIM MEDZ$(20)
70 MEDZ$(1)=" " : MEDZ$(20)=" "
80 MEDZ$(2)=MEDZ$ : I=0
100 REM Vkladanie zaznamov z klavesnice
110 I=I+1 : MENO$(1,20)=MEDZ$
120 INPUT MENO$
130 ID=(I-1)*20+1 : IH=I*20
140 IF MENO$(1,1)="*" THEN 200
150 Z$(ID,IH)=MENO$
160 IF I<25 THEN 110
200 PRINT "Koniec vkladania"
```

V programe najprv deklarujeme reťazcovú premennú Z$ rozsahu 500 znakov a vymedzerujeme ju (riadky 20, 30, 40). Ďalej deklarujeme reťazcovú premennú MENO$ rozsahu 20 znakov na vstup jedného záznamu a reťazcovú premennú MEDZ$ rozsahu 20 znakov, ktorú vymedzerujeme (túto premennú používame na priebežné medzerovanie premennej MENO$).

V úseku programu od riadka 100 do riadka 160 najprv vymedzerujeme reťazcovú premennú MENO$ (riadok 110) a potom príkaz INPUT očakáva vloženie priezviska a mena z klávesnice (riadok 120). V prípade, že náš zoznam obsahuje menej ako 25 záznamov a chceme ukončiť vkladanie, vložíme namiesto priezviska znak * (hviezdičku) a riadok 140 odovzdá riadenie riadku 200. Ak vložíme priezvisko a meno, riadok 130 určí začiatočnú a koncovú polohu príslušného záznamu v reťazci Z$ (zapisuje do úsekov po 20 znakov) a riadok 150 uloží záznam na patričné miesto. Po ukončení vkladania sa vypíše správa

Koniec vkladania

(riadok 200). Na Atari môžeme v reťazcoch pri zápise textov používať malé písmená.

Ukážeme teraz, ako zoznam vypísať na obrazovku a na tlačiareň a ako ho zapísať na kazetu.

Pri výstupe na obrazovku môžeme regulovať polohu výstupu príkazom POSITION. Ak tento príkaz nepoužijeme, výpis sa začína v prvom prázdnom riadku obrazovky. Ak bola pred prvým príka-

zom výstupu obrazovka prázdna, zobrazujú sa výstupy postupne zhora dole, po naplnení obrazovky výstupy „rolujú" zdola hore (ako na PMD 85–2).

Príklad 1.85

Výstup zoznamu na obrazovku naprogramujeme takto:

```
300 REM Vypis zoznamu na obrazovku
310 FOR I=1 TO 25
320 POSITION 10,I-1
330 ID=(I-1)*20+1 : IH=I*20
340 PRINT Z$(ID,IH)
350 NEXT I
```

Úsek programu vypisuje zoznam z reťazcovej premennej Z$ (po jednom zázname v každom riadku) zhora dole (od prvého riadka obrazovky), pričom výstup sa začína vždy od 10. stĺpca každého riadka. V prípade, že zoznam obsahuje menej ako 25 záznamov, posledné riadky budú prázdne.

Na výstup na tlačiareň používame príkaz

```
LPRINT zoznam
```

kde *zoznam* je zoznam vystupujúcich konštánt, premenných alebo zložitejších výrazov, pre ktorý platia rovnaké pravidlá ako pre zoznam príkazu PRINT.

Príklad 1.86

Zoznam môžeme vypísať na tlačiareň napr. takto:

```
500 REM Vypis zoznamu na tlaciaren
510 PRINT "Zapni tlaciaren a nastav papier"
520 FOR I=1 TO 25
530 ID=(I-1)*20+1 : IH=I*20
540 LPRINT Z$(ID,IH)
550 NEXT I
560 PRINT "Koniec tlace zoznamu"
```

Uvedieme ešte spôsob programovania dotazu s očakávanou odpoveďou A (áno) alebo N (nie).

Príklad 1.87

Rozhodovanie (napr. o tom, či chceme vytlačiť zoznam na tlačiarni) môžeme naprogramovať takto:

```
360 DIM Q0$(1) : Q0$(1)=" "
370 PRINT "Chcete vystup na tlaciaren (A, N): ";
380 INPUT Q0$
390 IF Q0$(1)="A" OR Q0$(1)="a" THEN 500
400 IF Q0$(1)="N" OR Q0$(1)="n" THEN 600
410 PRINT "Nerozumiem" : GOTO 370
    .
    .
    .
500 REM Vypis zoznamu na tlaciaren
    .
    .
    .
600 END
```

Všimnime si, že aj reťazcovú premennú, do ktorej vkladáme jediný znak, musíme deklarovať príkazom DIM (premenná Q0$ v riadku 360).

1.7.5 PRÁCA SO SÚBORMI

Atari basic poskytuje možnosť uchovávať údajové súbory na kazetách alebo disketách. Ukážeme, ako súbor zapísať na kazetu (disketu), resp. ako ho prečítať z kazety (diskety) do vnútornej pamäte.

Na programovanie práce so súbormi používame príkazy OPEN, PRINT #n, INPUT #n a CLOSE.

Príkaz OPEN otvára súbory a zariadenia pre vstup a výstup. Používame ho v tvare

```
OPEN #n,c,0,špec
```

kde *n* je číslo súboru, pomocou ktorého sa na súbor odvolávame v príkazoch PRINT #n a INPUT #n, *c* je označenie činnosti, ktorú budeme so súborom vykonávať (čítanie, zápis, čítanie aj zápis, doplňovanie), *špec* je tzv. špecifikácia súboru, s ktorým chceme pracovať. V programe môžeme súčasne otvoriť najviac 7 súborov, teda číslo súboru *n* môže byť z intervalu od 1 do 7. Činnosť definujeme číselným parametrom takto:

c = 4	vstup (iba čítanie),
c = 8	výstup (iba zápis),
c = 9	dopĺňanie (pripisovanie nových záznamov na koniec súboru),

c = 12 aktualizácia (čítanie aj zápis).

Posledné dve činnosti možno vykonávať iba so súbormi na disketách.

Špecifikácia definuje zariadenie, na ktorom sa súbor nachádza, a názov súboru. Zapisuje sa ako reťazcová konštanta v tvare

"zar:meno.príp"

kde *zar:* je názov zariadenia, *meno* je maximálne 8-znakový názov súboru a *príp* je maximálne trojznaková prípona. Pri práci s kazetovým záznamníkom a disketovou jednotkou používame tieto názvy zariadení:

C: kazetový záznamník,
D1: disketová jednotka.

Meno a príponu priraďujeme súboru pri jeho vytvorení. Prípona nie je povinná, používa sa na bližšiu špecifikáciu typu súboru. Napr. prípona BAS označuje, že ide o programový súbor s programom v basicu, prípona DAT označuje, že ide o údajový súbor, prípona TXT označuje textový súbor a pod.

Príklad 1.88

Príkaz

```
100 OPEN #1,8,0,"C:ZOZNAM.DAT"
```

otvára súbor s číslom 1 pre výstup na kazetu, pričom súbor bude uložený pod názvom ZOZNAM.DAT. Príkaz

```
200 OPEN #2,4,0,"C:ZOZNAM.DAT"
```

otvára súbor s číslom 2 pre vstup z kazety. Príkaz

```
300 OPEN #3,12,0,"D1:ZOZNAM.DAT"
```

otvára súbor s číslom 3 na diskete pre aktualizáciu; predpokladá sa, že ide o už existujúci súbor ZOZNAM.DAT.

Príkaz

```
CLOSE #n
```

uzatvára súbor otvorený príkazom OPEN pod číslom *n*. Uvoľňuje tak číslo súboru *n* pre otvorenie nového súboru.

Do súboru otvoreného pre výstup zapisujeme príkazom

```
PRINT #n,zoznam
```

kde *n* je číslo súboru, použité v príslušnom príkaze OPEN, a *zoznam* je zoznam údajov, ktoré chceme zapísať do súboru.

Zo súboru otvoreného pre vstup čítame príkazom

```
INPUT #n,zoznam
```

kde *n* je číslo súboru, použité v príslušnom príkaze OPEN, a *zoznam* je zoznam premenných, ktorým chceme priradiť údaje zo súboru.

Príklad 1.89

Zoznam, ktorý sme vložili úsekom programu z príkladu 1.84, zapíšeme na kazetu ako súbor ZOZNAM.DAT takto:

```
300 REM Zapis suboru na kazetu
310 PRINT CHR$(125);
320 PRINT
330 PRINT "ZAPIS ZOZNAMU NA KAZETU"
340 PRINT
350 PRINT "    Vykonajte tieto pripravy:"
360 PRINT
370 PRINT "    1. VLOZTE KAZETU"
380 PRINT "    2. PRETOCTE KAZETU NA ZACIATOK"
390 PRINT "    3. PO ODZNENI ZVUKOVEHO SIGNALU"
400 PRINT "       STLACTE 'RECORD' A 'PLAY'"
410 PRINT "    4. STLACTE Return"
420 POSITION 5,19
430 PRINT "Ste pripraveny (A, N): ";
440 INPUT Q0$
450 IF Q0$(1)="A" OR Q0$(1)="a" THEN 480
460 IF Q0$(1)="N" OR Q0$(1)="n" THEN 420
470 GOTO 420
480 OPEN #1,8,0,"C:ZOZNAM.DAT"
490 FOR I=1 TO 25
500 ID=(I-1)*20+1 : IH=I*20
510 PRINT #1,Z$(ID,IH)
520 NEXT I
530 CLOSE #1
```

Predpokladáme, že súbor ZOZNAM.DAT bude prvým súborom na kazete, preto najprv nastavíme kazetu na začiatok. Po vykonaní príkazu OPEN sa ozve zvukový signál. Na záznamníku stlačíme tlačidlá RECORD a PLAY a na klávesnici stlačíme *Return*. Zoznam uložený v poli Z$ sa zapíše na kazetu a po ukončení zápisu sa kazetový záznamník zastaví.

František Turnovec

PROGRAM 1.17

Program z výpisu 1.17 umožňuje vykonávať tieto činnosti: vložiť zoznam z klávesnice, zapísať ho na kazetu, načítať zoznam z kazety, prezerať si zoznam na obrazovke a vytlačiť ho na tlačiarni.

```
10 REM Program 1.17 - ZOZNAM, 2.5.88
20 REM Atari basic
30 DIM Z$(500)
40 Z$(1)=" ":Z$(500)=" ":Z$(2)=Z$
50 DIM VETA$(30)
60 DIM MEDZ$(20),MENO$(20)
70 MEDZ$(1)=" ":MEDZ$(20)=" "
80 MEDZ$(2)=MEDZ$
90 DIM Q0$(1):Q0$(1)=" "
100 REM Ponuka cinnosti
110 PRINT CHR$(125);
120 PRINT "Mozete si zvolit:"
130 PRINT
140 PRINT "   1. Vlozenie zoznamu"
150 PRINT "   2. Zapis zoznamu"
160 PRINT "   3. Precitanie zoznamu"
170 PRINT "   4. Prezeranie zoznamu"
180 PRINT "   5. Vytlacenie zoznamu"
190 PRINT "   6. Ukoncenie prace"
200 POSITION 2,10
210 PRINT "Vlozte poradove cislo cinnosti ";
220 INPUT K:IF K<1 OR K>6 THEN 200
230 ON K GOSUB 1000,2000,3000,4000,5000,10000
240 GOTO 110
1000 REM Vlozenie zoznamu
1010 PRINT CHR$(125);:I=0
1020 PRINT "Mozete vkladat:"
1030 I=I+1:MENO$(1,20)=MEDZ$
1040 POSITION 10,22:PRINT MENO$
1050 POSITION 10,22:INPUT MENO$
1060 ID=(I-1)*20+1:IH=I*20
1070 Z$(ID,IH)=MENO$
1080 IF MENO$(1,1)="*" THEN 1100
1085 POSITION 10,I+1:PRINT I;". ";Z$(ID,IH)
1090 IF I<25 THEN 1030
1100 PRINT CHR$(125);:PRINT "Koniec vkladania"
1110 FOR K=1 TO 500:NEXT K
1120 RETURN
2000 REM Zapis zoznamu na kazetu
2010 PRINT CHR$(125);:PRINT
2020 PRINT "ZAPIS ZOZNAMU NA KAZETU"
2030 PRINT
2040 PRINT "   Vykonajte tieto pripravy:"
2050 PRINT
2060 PRINT "   1. Vlozte kazetu"
2070 PRINT "   2. Pretocte kazetu na zaciatok"
2080 PRINT "   3. Po odoznení zvukoveho signalu"
2090 PRINT "      stlacte RECORD a PLAY"
2100 PRINT "   4. Stlacte klaves Return"
2110 POSITION 5,19
2120 PRINT "Ste pripraveny (A, N): ";
2130 INPUT Q0$
2140 IF Q0$(1)="A" OR Q0$(1)="a" THEN 2170
2150 IF Q0$(1)="N" OR Q0$(1)="n" THEN 2110
2160 PRINT "Nerozumiem...":GOTO 2110
2170 OPEN #1,8,0,"C:ZOZNAM.DAT"
2180 FOR I=1 TO 25
2190 ID=(I-1)*20+1:IH=I*20
2200 PRINT #1,Z$(ID,IH)
2205 IF Z$(ID,ID)="*" THEN 2220
2210 NEXT I
2220 CLOSE #1
```

```
2230 PRINT CHR$(125);:PRINT "Zapis ukonceny"
2240 FOR K=1 TO 500:NEXT K
2250 RETURN
3000 REM Precitanie zoznamu z kazety
3010 PRINT CHR$(125);:PRINT
3020 PRINT "PRECITANIE ZOZNAMU Z KAZETY"
3030 PRINT
3040 PRINT "   Vykonajte tieto pripravy:"
3050 PRINT
3060 PRINT "   1. Vlozte kazetu"
3070 PRINT "   2. Pretocte kazetu na zaciatok"
3080 PRINT "   3. Po odzneni zvukoveho signalu"
3090 PRINT "      stlacte PLAY"
3100 PRINT "   4. Stlacte klaves Return"
3110 POSITION 5,19
3120 PRINT "Ste pripraveny (A, N): ";
3130 INPUT Q0$
3140 IF Q0$(1)="A" OR Q0$(1)="a" THEN 3170
3150 IF Q0$(1)="N" OR Q0$(1)="n" THEN 3110
3160 PRINT "Nerozumiem...":GOTO 3110
3170 OPEN #1,4,0,"C:ZOZNAM.DAT"
3180 FOR I=1 TO 25
3190 ID=(I-1)*20+1:IH=I*20
3200 INPUT #1,VETA$
3210 Z$(ID,IH)=VETA$(11,30)
3220 IF VETA$(11,11)="*" THEN 3240
3230 NEXT I
3240 CLOSE #1
3250 PRINT CHR$(125);:PRINT "Citanie ukoncene"
3260 FOR K=1 TO 500:NEXT K
3270 RETURN
4000 REM Prezeranie zoznamu na obrazovke
4010 PRINT CHR$(125);
4020 FOR I=1 TO 25
4030 ID=(I-1)*20+1:IH=I*20
4040 IF Z$(ID,ID)="*" THEN 4070
4050 PRINT "     ";Z$(ID,IH)
4060 NEXT I
4070 FOR K=1 TO 500:NEXT K
4080 POSITION 2,23
4090 PRINT "Navrat k ponuke stlacenim Return";
4100 INPUT Q0$
4110 RETURN
5000 REM Vypis zoznamu na tlaciaren
5010 PRINT CHR$(125);
5020 PRINT "VYPIS ZOZNAMU NA TLACIAREN"
5030 PRINT
5040 PRINT "   Zapnite tlaciaren a nastavte papier"
5050 POSITION 5,19
5060 PRINT "Ste pripraveny (A, N): ";
5070 INPUT Q0$
5080 IF Q0$(1)="A" OR Q0$(1)="a" THEN 5110
5090 IF Q0$(1)="N" OR Q0$(1)="n" THEN 5050
5100 PRINT "Nerozumiem...":GOTO 5050
5110 LPRINT "     ZOZNAM ZIAKOV":LPRINT
5120 FOR I=1 TO 25
5130 ID=(I-1)*20+1:IH=I*20
5140 IF Z$(ID,ID)="*" THEN 5170
5150 LPRINT "     ";I;". ";Z$(ID,IH)
5160 NEXT I
5170 PRINT CHR$(125);:PRINT "Tlac ukoncena"
5180 FOR K=1 TO 500:NEXT K
5190 RETURN
10000 END
```

Podobným spôsobom pracujeme so súbormi na disketách. Rozdiel spočíva iba v inom označení zariadenia v príkaze OPEN a v jednoduchšej manipulácii s disketou (stačí ju iba vložiť do otvoru

disketového mechanizmu a zatvoriť dvierka). Zápis na disketu a čítanie z diskety prebiehajú oveľa rýchlejšie ako práca s kazetou.

1.7.6 FAREBNÁ GRAFIKA

Podstatné rozdiely medzi Atari basicom a basicom G sú v spôsobe programovania grafických aplikácií. Atari basic má 16 grafických režimov, ktoré definujú rôzne formáty obrazovky pre grafické príkazy a umožňujú využívať 256 farebných odtieňov. Grafické režimy umožňujú rozdeliť obrazovku na grafickú a textovú časť (tzv. okienko).

V tomto odseku sa oboznámime iba s najdôležitejšími poznatkami o grafických príkazoch Atari basicu: GRAPHICS, COLOR, SETCOLOR, PLOT a DRAWTO.

a) Príkaz GRAPHICS

Príkazom GRAPHICS zadávame požadovaný zobrazovací režim. Používame ho v tvare

```
GRAPHICS rež+dod
```

kde *rež* je číslo zobrazovacieho režimu obrazovky (od 0 do 15) a *dod* je konštanta, ktorú môžeme pripočítať k číslu režimu zobrazenia a definovať tak spôsob zobrazovania.

Jednotlivé zobrazovacie režimy sa líšia rozlišovacou schopnosťou a počtom použiteľných farieb.

Režim 0 sa nastavuje automaticky. Ak sa pri práci s počítačom nachádzame v inom režime, prejdeme k režimu 0 príkazom

```
GRAPHICS 0
```

Režim 0 definuje alfanumerický formát obrazovky, teda zobrazovanie znakov do 24 riadkov (po 40 znakov v riadku). Obrázky v ňom môžeme vytvárať iba pomocou grafických znakov. Štandardné farby v tomto režime sú: pozadie – tmavomodrá, popredie (farba „pera") – svetlomodrá, okraje obrazovky – čierna.

Režim 1 sa zadáva príkazom

```
GRAPHICS 1
```

Ide o alfanumerický režim, ktorý delí obrazovku na dve časti: *texto-*

vé okienko, ktoré zaberá 4 posledné riadky obrazovky, a *grafické pole obrazovky* v hornej časti obrazovky na zobrazovanie zväčšených znakov, ktoré zaberá 20 riadkov základného formátu obrazovky. V režime 1 sa príkazom PRINT vypisuje do textového okienka normálnym formátom (40 znakov v riadku) a príkazom

```
PRINT #6;zoznam
```

do grafického poľa, pričom znaky v grafickom poli majú dvojnásobnú šírku v porovnaní so základným formátom. V grafickom poli môžeme teda zobraziť 20 riadkov po 20 znakov.

V režime 1 môžeme používať 5 farieb: odlišné farby pre pozadie, veľké písmená, malé písmená, inverzné veľké písmená, inverzné malé písmená. Štandardné farby v režime 1 sú

pozadie	čierna
veľké písmená	oranžová
malé písmená	svetlozelená
inverzné veľké písmená	tmavomodrá
inverzné malé písmená	červená

Režim 2 sa líši od režimu 1 iba tým, že v grafickom poli obrazovky sa zobrazujú znaky dvojnásobnej šírky a výšky, môžeme v ňom teda zobraziť iba 10 riadkov po 20 znakov. Štandardné farby sú rovnaké ako v režime 1. Režim 2 nastavíme príkazom

```
GRAPHICS 2
```

Ak chceme v režimoch 1 a 2, resp. v ďalších režimoch pokryť grafickým poľom celú obrazovku (teda zrušiť textové okienko), použijeme príkaz GRAPHICS v tvare

```
GRAPHICS rež+16
```

Ak chceme, aby sa pri prechode k príslušnému režimu nevymazal doterajší obsah obrazovky, použijeme

```
GRAPHICS rež+32
```

Ak použijeme

```
GRAPHICS rež+16+32
```

zabezpečíme obidva spôsoby zobrazenia (zrušenie textového okienka aj zachovanie doterajšieho obsahu obrazovky).

Príklad 1.90

Príkaz

100 GRAPHICS 1+16

nastavuje grafický režim 1 bez textového okienka. Rovnaký účinok má príkaz

100 GRAPHICS 17

Príkazom

200 GRAPHICS 34

nastavíme grafický režim 2 s textovým okienkom a zachovaním obsahu obrazovky.

Vlastné grafické príkazy môžeme používať až od režimu 3.

Režim 3 formátuje grafické pole obrazovky na 20 riadkov po 40 stĺpcov, teda jednotkou zobrazenia je obdĺžnik veľkosti jedného znaku základného formátu v režime 0. V tomto režime možno pracovať s paletou 4 farieb.

Režim 4 formátuje grafické pole obrazovky na 40 riadkov po 80 stĺpcov, teda základnou jednotkou zobrazenia je bod v mierke 80 × 40. V tomto režime možno pracovať iba s paletou 2 farieb.

Režim 5 sa od režimu 4 odlišuje iba tým, že umožňuje pracovať s paletou 4 farieb.

Režim 6 formátuje grafické pole obrazovky na 80 riadkov po 160 stĺpcov. Možno v ňom pracovať s paletou 2 farieb (farba pozadia a farba pera).

Režim 7 sa od režimu 6 odlišuje iba tým, že umožňuje prácu s paletou 4 farieb (farba pozadia a 3 farby pera).

Režim 8 umožňuje „najjemnejšiu" grafiku. Formátuje grafické pole obrazovky na 160 riadkov po 320 stĺpcov. Umožňuje kresliť iba jednou farbou: štandardne svetlomodrou na tmavomodrom pozadí s čiernymi okrajmi.

Na počítači Atari 130/XE (800/XL) je k dispozícii ešte ďalších 7 grafických režimov s rôznymi formátmi grafického poľa obrazovky a s väčším počtom farieb palety (až 16).

V každom z režimov 3 až 8 možno (podobne ako v režimoch 1 a 2) rozhodnúť o zrušení textového okienka a zachovaní obsahu obrazovky pri prechode k inému režimu.

b) Príkaz SETCOLOR

V Atari basicu môžeme používať 16 základných farieb:

0 šedá

1 zlatožltá

2 oranžová
3 červenooranžová
4 ružová
5 purpurová
6 fialová
7 tmavomodrá
8 svetlomodrá
9 belasá
10 tyrkysová
11 modrozelená
12 zelená
13 žltozelená
14 svetlozelená (oranžovozelená)
15 svetlooranžová

Farby sme uviedli s ich číselnými kódmi, pomocou ktorých sa na ne odvolávame v príkazoch SETCOLOR. Z týchto základných farieb môžeme vytvoriť spolu 256 farieb zadaním intenzity jasu od 0 do 14 (len párne čísla, s výnimkou módov GTIA, ktoré používajú párne i nepárne čísla až do 15). Čím vyššia je hodnota intenzity jasu, tým svetlejšia a jasnejšia je farba. Napríklad s využitím rôznych intenzít jasu môžeme z farby 0 (šedá) vytvoriť všetky odtiene od bielej po čiernu farbu.

Na riadenie farebnosti zobrazenia sa používajú tzv. *farbové registre*, ktoré definujú paletu farieb. Keďže v režimoch 0 až 8 možno používať paletu s maximálne 5 farbami, máme 5 farbových registrov, ktoré označujeme číslami 0, 1, 2, 3 a 4.

Príkaz

```
SETCOLOR reg,f,j
```

kde *reg* je číslo farbového registra, *f* je číslo farby a *j* je intenzita jasu, sa používa na nastavenie farby a intenzity jasu v príslušnom farbovom registri. Farbové registre majú rôzny význam v rôznych zobrazovacích režimoch.

Význam farbových registrov v režime 0:
register 0 nepoužíva sa
register 1 farba popredia (pera)
register 2 farba pozadia
register 3 nepoužíva sa
register 4 farba okrajov obrazovky

Význam farbových registrov v režimoch 1 a 2:
register 0 farba pera pre veľké písmená
register 1 farba pera pre malé písmená
register 2 farba pera pre inverzné veľké písmená
register 3 farba pera pre inverzné malé písmená
register 4 farba pozadia

Príklad 1.91
Príkazom
100 SETCOLOR 2,3,4
zmeníme v režime 0 štandardnú tmavomodrú farbu pozadia na červenooranžovú s malou intenzitou jasu (zmenili sme farbový register 2, ktorý v režime 0 nastavuje farbu pozadia, na farbu číslo 3 s intenzitou jasu 4).
Príkazmi
200 GRAPHICS 1
210 SETCOLOR 4,15,5
nastavíme režim 1 a zmeníme register 4, ktorý v režimoch 1 a 2 určuje farbu pozadia, na farbu číslo 15 s intenzitou jasu 5.

Význam farbových registrov v režimoch 3, 5, 7:
register 0 farba pera
register 1 farba pera
register 2 farba pera
register 3 nepoužíva sa
register 4 farba okrajov a pozadia

Význam farbových registrov v režimoch 4 a 6:
register 0 farba pera
register 1 nepoužíva sa
register 2 nepoužíva sa
register 3 nepoužíva sa
register 4 farba okrajov a pozadia

Význam farbových registrov v režime 8:
register 0 nepoužíva sa
register 1 farba pera
register 2 farba pozadia
register 3 nepoužíva sa
register 4 farba okrajov

V grafických režimoch (od režimu 3 vyššie) sa príkaz SETCOLOR používa v súčinnosti s grafickým príkazom COLOR.

c) Príkaz COLOR

Príkaz slúži na nastavenie farby pera (kreslenia) pre grafické príkazy PLOT a DRAWTO. Používa sa v tvare

```
COLOR reg
```

kde *reg* je číslo farbového registra, ktorý sa ma použiť pri kreslení bezprostredne nasledujúcimi grafickými príkazmi. Farby registrov sa nastavujú príkazom SETCOLOR.

Príklad 1.92

Príkazy

```
500 GRAPHICS 7
510 COLOR 0
```

nastavia grafický režim 7 a zvolia pre kreslenie bezprostredne nasledujúcimi grafickými príkazmi farbu z registra 0 (štandardné farby v registroch režimu 7 sú: register 0 – oranžová farba pera, register 1 – zelená farba pera, register 2 – svetlomodrá farba pera, register 4 – čierna farba pozadia a okrajov), bude sa teda kresliť oranžovou na čiernom pozadí.

d) Príkaz PLOT

Príkaz sa používa v tvare

```
PLOT x,y
```

kde *x* je súradnica stĺpca a *y* je súradnica riadka. Tieto súradnice určujú bod, ktorý sa má zobraziť na obrazovke.

Na rozdiel od príkazu PLOT na PMD 85 príkaz PLOT v Atari basicu slúži iba na nastavenie a vykreslenie bodu na obrazovke. Keď chceme nastaviť bod bez jeho vykreslenia, musíme zmeniť farbu pera tak, aby bola zhodná s farbou pozadia.

Príklad 1.93

Úsek programu

```
100 GRAPHICS 3
110 COLOR 1
120 PLOT 0,0
```

nastaví grafický režim 3, zvolí farbu kreslenia z registra 1 (štandardne zelená) a nakreslí v ľavom hornom rohu obrazovky zelený bod na čiernom pozadí.

e) Príkaz DRAWTO

Príkaz sa používa v tvare

```
DRAWTO x,y
```

kde *x* je súradnica stĺpca a *y* je súradnica riadka. Jeho účinkom sa bod určený súradnicami (x,y) spojí úsečkou s posledným bodom nastavenia pera.

Príklad 1.94

Úsek programu

```
100 GRAPHICS 8
110 PLOT 10,10
120 DRAWTO 50,10
130 DRAWTO 50,40
140 DRAWTO 10,40
150 DRAWTO 10,10
```

nastaví grafický režim 8 a nakreslí v ľavom hornom rohu obrazovky svetlomodrý obdĺžnik na tmavom pozadí.

PROGRAM 1.18

Výpis 1.18 obsahuje program, v ktorom sme použili grafické príkazy Atari basicu. Program kreslí v rôznych grafických režimoch zaujímavé priestorové obrazce. Grafický režim kreslenia si zadáva používateľ odpoveďou na dotaz

Zvol graficky rezim (3, 5, 7):

```
10 REM Program 1.18 - PRIESTOR, 25. 4. 88 (Atari)
20 REM Atari basic
25 PRINT CHR$(125);:PRINT "Zvol graficky rezim (3, 5, 7): ";
30 L=0:H=0
40 INPUT MOD
45 IF MOD<>3 AND MOD<>5 AND MOD<>7 THEN 15
50 IF MOD=3 THEN P=39:D=19
60 IF MOD=5 THEN P=79:D=39
70 IF MOD=7 THEN P=159:D=79
80 GRAPHICS MOD
90 PRINT "Graficky rezim ";MOD
100 FOR K=1 TO 1000
110 COLOR 2
120 TRAP 300
130 PRINT CHR$(253)
140 REM Kresba
150 PLOT L,H
160 COLOR 1
170 DRAWTO P,H
180 COLOR 2
190 DRAWTO P,D
200 COLOR 1
210 DRAWTO L,D
220 COLOR 3
230 DRAWTO L,H
240 REM Pauza
250 FOR POCKAJ=1 TO 500:NEXT POCKAJ
260 REM Nove rozmery
270 L=L+2:H=H+2:P=P-2:D=D-2
280 NEXT K
300 GRAPHICS 0
310 END
```

Všimnime si riadky 240 a 250. Keďže v Atari basicu nemáme k dispozícii príkaz PAUSE, prerušujeme program pomocou cyklu FOR NEXT. V riadku 120 sme použili príkaz TRAP na odskok v prípade výskytu chyby (beh programu sa ukončí, keď meniace sa rozmery kreslenia vyjdú za rámec prípustného formátu grafického poľa obrazovky).

1.7.7 SPRACOVANIE ZVUKOV

V Atari basicu môžeme programovať rôzne zvukové efekty a melódie. Na programovanie zvukov používame príkaz

SOUND a,b,c,d

kde *a* je číslo hlasu (0 až 3), *b* je výška tónu (0 až 255), *c* je skreslenie zvuku (0 až 14) a *d* je hlasitosť (0 až 15).

Môžeme programovať zároveň 4 zvuky (4 hlasy). Po vykonaní príkazu SOUND sa ozve príslušný zvuk a tento zvuk trvá až do vykonania ďalšieho príkazu SOUND s rovnakým prvým parametrom (s rovnakým hlasom).

Zvuk príslušného hlasu vypneme príkazom

SOUND a,Ø,Ø,Ø

Príklad 1.95

```
SOUND 1,5Ø,Ø,8
```

odštartuje zvuk podobný výbuchu. Zvukový efekt ukončíme vložením príkazu

```
SOUND 1,Ø,Ø,Ø
```

(obidva príkazy môžete vložiť v priamom režime).

Tretí parameter príkazu SOUND určuje čistotu zvuku. Čistý tón (bez skreslenia) nastavíme parametrom 10, napr.

```
SOUND Ø,243,1Ø,8
```

nastaví tón „nízke C“ a

```
SOUND Ø,29,1Ø,8
```

nastaví tón „vysoké C“.

Posledný parameter príkazu SOUND určuje hlasitosť zvuku. Môže byť z intervalu 0 až 15, vo väčšine prípadov je vhodná hodnota 8.

Príklad 1.96

Nasledujúci úsek programu vytvára zvukový efekt sirény:

```
1Ø FOR TON=5Ø TO 1ØØ
2Ø SOUND Ø,TON,1Ø,8
3Ø NEXT TON
4Ø FOR TON=1ØØ TO 5Ø STEP -1
5Ø SOUND Ø,TON,1Ø,8
6Ø NEXT TON
7Ø GOTO 1Ø
```

Sirénu môžete ukončiť stlačením tlačidla *Reset*.

Pomocou príkazu SOUND môžeme programovať hudobné motívy. Máme k dispozícii necelé 4 oktávy. V tabuľke 1.4 uvádzame číselné kódy jednotlivých tónov v týchto oktávach.

Tabuľka 1.4
Kódy tónov pre príkaz SOUND

tón	1. oktáva	2. oktáva	3. oktáva	4. oktáva
C	243	121	60	29
Cis	230	114	57	28
D	217	108	53	26
Dis	204	102	50	
E	193	96	47	
F	182	91	45	
Fis	173	85	42	
G	162	81	40	
Gis	153	76	37	
A	144	72	35	
Ais	136	68	33	
H	128	64	31	

František Turnovec

PROGRAM 1.19

Výpis 1.19 obsahuje program na demonštráciu prvých troch oktáv. Po vložení čísla oktávy odpoveďou na dotaz

Vloz oktavu (1 az 3):

sa ozve stupnica príslušnej oktávy. Kódy tónov sú uvedené v zozname konštánt (príkazy DATA) a na začiatku vykonávania programu sa načítajú do poľa TON príkazom READ.

```
10 REM Program 1.19 - OKTAVY, 3.5.88
20 REM Atari basic
30 PRINT CHR$(125);
40 REM Nacitanie tonov
50 DIM TON(36)
60 FOR INDEX=1 TO 36
70 READ A:TON(INDEX)=A
80 NEXT INDEX
100 REM Ukazka tonov
110 PRINT "Vloz oktavu (1 az 3): ";
120 INPUT O
130 IF INT(O)<1 OR INT(O)>3 THEN 110
150 FOR T=(O-1)*12+1 TO O*12
160 SOUND 0,TON(T),10,8
170 FOR I=1 TO 50:NEXT I
180 SOUND 0,0,0,0
190 NEXT T
400 END
500 REM Hodnoty tonov
510 DATA 243,230,217,204,193,182,173,162,153,144,136,128
520 DATA 121,114,108,102,96,91,85,81,76,72,68,64
530 DATA 60,57,53,50,47,45,42,40,37,35,33,31
```

Program 1.20

Program z výpisu 1.20 ilustruje možnosti spojenia grafiky a zvukových efektov.

```
10 REM Program 1.20 - KOZMICKA LOD, 14. 5. 88
20 REM Atari basic
30 DIM Q0$(1),A$(14)
50 GOSUB 100
60 GOSUB 400
70 GOTO 60
100 REM uvodna obrazovka
110 GRAPHICS 2
120 PRINT #6;"        HRA"
130 PRINT #6
140 PRINT #6;"   KOZMICKA LOD"
150 PRINT "Riadite let kozmickej lode v oblasti"
160 PRINT "zvyseneho vyskytu meteoritov. Let mo-"
170 PRINT "zete riadit joystickom alebo tlacit-"
180 PRINT "kami Start a Select.";
190 FOR I=1 TO 2500:NEXT I
200 PRINT CHR$(125);
210 PRINT "Mate joystick? (a, n): ";
220 OPEN #1,4,0,"K":GET #1,R:CLOSE #1:Q0$(1)=CHR$(R)
230 IF Q0$="A" OR Q0$="a" THEN KR=1:GOTO 330
240 PRINT Q0$
250 IF Q0$="N" OR Q0$="n" THEN KR=2:GOTO 270
260 GOTO 210
270 PRINT "Budete teda pouzivat tlacitka:"
300 PRINT "   Select   uhybny manever vpravo"
310 PRINT "   Start    uhybny manever vlavo";
320 FOR I=1 TO 2500:NEXT I
330 PRINT CHR$(125);
340 PRINT "Pripravte sa, zaciname...";
350 FOR I=1 TO 1000:NEXT I
360 RETURN
400 REM podprogram hry
410 GRAPHICS 0:POKE 710,0
420 POKE 752,1:SOUND 0,150,10,12
430 FOR I=0 TO 15:POSITION 12,23
440 ? "|  KOZMICKA LOD  |"
450 NEXT I:B=0
460 A$="  Zrazka po    "
470 P=PEEK(106)-20:POKE 54279,P
480 M=P*256:POKE 559,46
490 POKE 53277,3:POKE 53256,0:POKE 704,198
500 FOR Q=M+512 TO M+640:POKE Q,0:NEXT Q
510 X=100:Y=30
520 POKE M+512+Y,7:POKE M+513+Y,7
530 POKE M+514+Y,2
540 POKE 53278,0
550 ON KR GOTO 560,570
560 X=X+4*(STICK(0)=7)-4*(STICK(0)=11):GOTO 580
570 X=X+4*(PEEK(53279)+6=11)-4*(PEEK(53279)+1=7)
580 POKE 53248,X:B=B+100
590 POSITION 12,23:? "|                    |"
600 POSITION INT(RND(0)*16)+13,23
610 ? "*"
615 IF PEEK(53252)<>4 THEN 540
620 REM zrazka
630 FOR CAS=1 TO 4
650 FOR TON=50 TO 100
660 SOUND 0,TON,10,8
670 NEXT TON
680 FOR TON=100 TO 50 STEP -1
690 SOUND 0,TON,10,8
700 NEXT TON
710 NEXT CAS
720 ? A$;B;" km letu!!!      "
```

```
730 POSITION 3,23
740 ? "Stlacte  akykolvek klaves"
750 POKE 53248,0
760 OPEN #1,4,0,"K":GET #1,R:CLOSE #1
770 RETURN
```

Kozmická loď letí oblasťou zvýšeného výskytu meteoritov a hráč, ktorý ovláda jej pohyb, musí riadiť let kozmickej lode tak, aby nedošlo k zrážke s meteoritom. Úspešnosť v hre sa meria počtom kilometrov letu bez nehody. Letové manévre (pohyb vľavo, vpravo) riadite buď joystickom (jeho naklonením v príslušnom smere), alebo funkčnými tlačidlami *Start* (pohyb vľavo) a *Select* (pohyb vpravo). Spôsob riadenia (v závislosti od vybavenia vášho počítača) oznámite programu v úvodnom dialógu pred začiatkom hry. Zrážka je sprevádzaná zvukom sirény. Po zrážke obnovíte hru stlačením ľubovoľného klávesu. Hru môžete kedykoľvek ukončiť stlačením tlačidla *Reset*, ktoré tiež vypne zvukový signál.

1.7.8 ŠTARTOVANIE PROGRAMOV V STROJOVOM KÓDE

Väčšina obľúbených hier a iných programov (napr. interpret jazyka Turbo basic, textové procesory a pod.) sa rozširuje na kazetách v podobe programových súborov v strojovom kóde. Takéto programy nemožno čítať do pamäte povelom CLOAD a štartovať povelom RUN pod riadením interpreta Atari basicu.

Program v strojovom kóde, uložený na kazete, štartujeme takto:

a) Do kazetového záznamníka vložíme kazetu so strojovým programom a nastavíme ju na začiatok programového súboru.

b) Vložíme príkaz BYE, ktorý ukončí činnosť interpreta Atari basicu a naštartuje tzv. SELF TEST (program, ktorý overuje funkčnú schopnosť jednotlivých prvkov počítača).

c) Stlačíme tlačidlo *Reset*, držíme ho a stlačíme súčasne tlačidlá *Option* a *Start* v pravej hornej časti panelu počítača.

d) Uvoľníme *Reset* (pričom naďalej držíme tlačidlá *Option* a *Start*) a počkáme na zvukový signál.

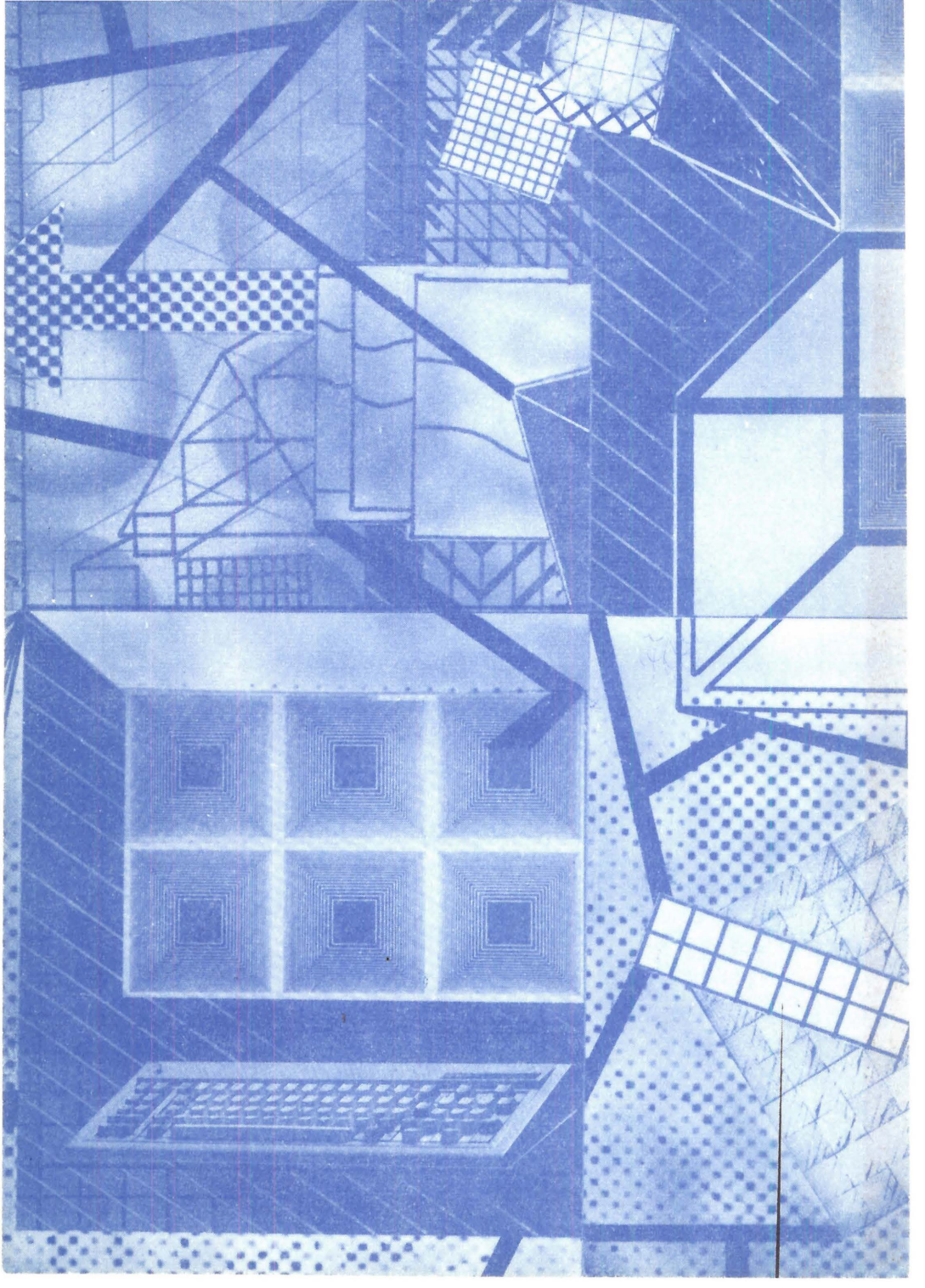
e) Po odznení zvukového signálu uvoľníme tlačidlá *Option* a *Start*.

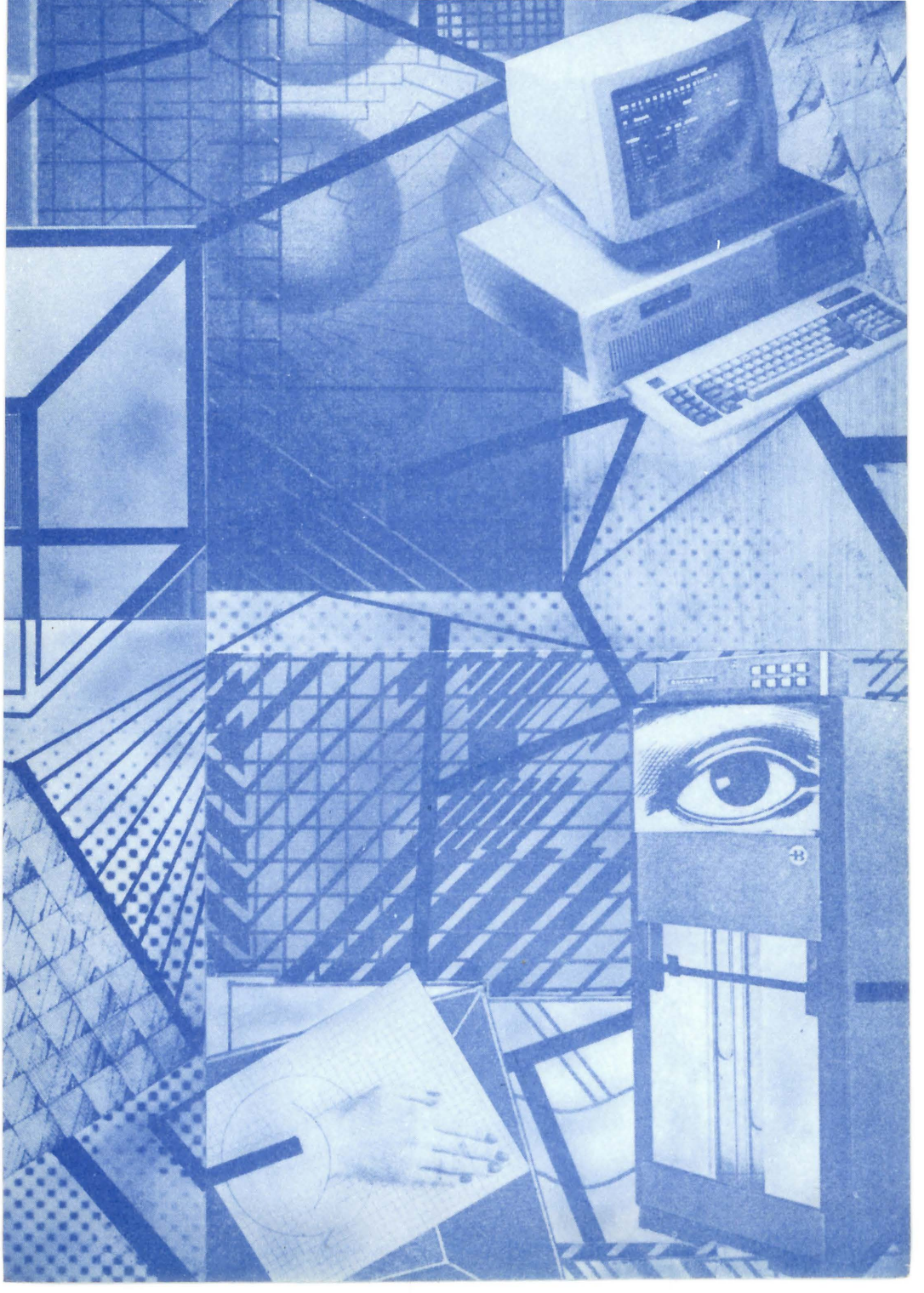
f) Na kazetovom záznamníku stlačíme tlačidlo PLAY.

g) Stlačíme *Return* na klávesnici.

Program sa začne nahrávať do vnútornej pamäte počítača. Po ukončení nahrávania sa záznamník zastaví, program sa automaticky odštartuje a ohlási sa prvými správami na obrazovke. V niektorých prípadoch sa programový súbor skladá z dvoch častí. Po načítaní

prvej časti sa záznamník zastaví. V tomto prípade treba stlačiť kláves *Return*, ktorý zabezpečí načítanie druhej časti programu a jeho odštartovanie.





2. TRIDSAŤ PROGRAMOV NIELEN PRE ZÁBAVU

V prvej časti knihy sme sa usilovali vysvetliť základné pravidlá programovania v basicu G a Atari basicu. Keďže programovanie je predovšetkým zručnosť, použili sme pri vysvetľovaní veľa príkladov. Dvadsať ucelených programov, ktoré sme vám už ponúkli, doplníme v tejto časti o ďalších tridsať programov z rôznych oblastí využitia osobných počítačov. Uvedené programy sú určené predovšetkým na rozvoj samostatných programátorských návykov podľa zásady: lepší jeden príklad ako desať poučiek. Veríme, že niektoré z nich vás aj pobavia, najmä však, že vás budú inšpirovať k vlastným pokusom o samostatné nové programy.

Prvá skupina programov je z oblasti grafiky. Sú venované kresle-

niu základných geometrických obrazcov, rôznych typov grafov a kresleniu obrázkov.

Druhá skupina obsahuje počítačové hry. Keďže počítače by sa nemali stať hracími automatmi, usilovali sme sa nájsť také hry, pri ktorých treba nielen stláčať klávesy, ale aj rozmýšľať.

Tretia skupina programov je venovaná praktickým aplikáciám počítačov pri výpočtoch, spracovaní textových údajov a výučbe.

Programy neprepisujte mechanicky, usilujte sa každý príkaz a každý úsek programu pochopiť. Vracajte sa pritom k jednotlivým kapitolám prvej časti knihy. Programy neobsahujú žiadne prvky, ktoré neboli vysvetlené v predchádzajúcom texte. Výpisy všetkých programov tejto časti knihy sú priamo z počítača, takže by nemali obsahovať chyby. Ak však predsa narazíte na chybu, pripomeňte si starý programátorský aforizmus: každá posledná chyba v programe je v skutočnosti predposlednou.

2.1 KRESLÍME NA OBRAZOVKE

Na obrazovke počítača možno pomocou grafických príkazov kresliť základné geometrické obrazce, grafy a diagramy, ako aj obrázky, ktoré vychádzajú z vašich vlastných výtvarných nápadov. V praxi sa počítačová grafika používa na automatizáciu projektových prác – počítač dokáže rýchlo a spoľahlivo vytvárať technické výkresy a spájať kreslenie s potrebnými výpočtami.

2.1.1 KRUŽNICA A ELIPSA

Funkcie sínus a kosínus, ktoré možno mnohí z vás v matematike príliš neobľubujú, nám pri kreslení na počítači poskytujú neoceniteľné služby. Bez nich by sme napríklad na obrazovke ťažko nakreslili kružnicu.

Pri kreslení kružnice využívame možnosť výpočtu súradníc kružnice podľa vzorcov

$x = r.\cos(a) + x_0$

$y = r.\sin(a) + y_0$

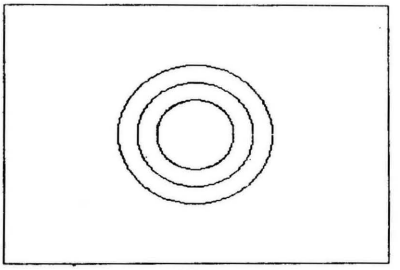
kde (x, y) sú súradnice bodov kružnice so stredom (x_0, y_0) a polomerom r, pričom uhol a sa mení od 0 do 360 stupňov (t. j. od 0 do 2π).

PROGRAM 2.1

Program z výpisu 2.1 zobrazuje kružnice so zadaným stredom a polomerom. Polohu kružnice na obrazovke zadávame (odpoveďou na dotaz počítača) v mierke 0–100 pre vodorovné a 0–100 pre zvislé súradnice. Program umožňuje zobraziť viac kružníc. Napr. na obr.

```
10 REM PROGRAM 2.1 - KRUZNICE, 12. 5. 88
20 REM BASIC G (PMD 85, DIDAKTIK ALFA)
30 GCLEAR : SCALE 0,100,0,100
40 DISP "VLOZ STRED KRUZNICE (X,Y): "
50 INPUT X,Y
60 DISP "VLOZ POLOMER KRUZNICE R: "
70 INPUT R
100 REM ZOBRAZENIE KRUZNICE
110 MOVE X+R,Y
120 PI=3.14159
130 FOR I=0 TO 2*PI STEP PI/100
140 PLOT R*COS(I)+X,R*SIN(I)+Y
150 NEXT I
160 REM DOTAZ O POKRACOVANI
170 DISP "CHCES DALSIU KRUZNICU? (A,N): "
180 Q0$=CHR$(USR(-31583))
190 IF Q0$="A" THEN 40
200 IF Q0$="N" THEN 220
210 GOTO 170
220 END
```

2.1 uvádzame zobrazenie troch sústredných kružníc. Program je určený pre počítače PMD 85 a Didaktik Alfa.



2.1 Kružnice nakreslené programom 2.1

PROGRAM 2.2

Program z výpisu 2.2 kreslí farebné kružnice na počítačoch Atari XL a XE. Polohu kružnice na obrazovke zadávame stredom a polomerom vo formáte grafického režimu 8 (320 stĺpcov a 160 riadkov). Program kreslí farebne, ale zobrazí kružnicu aj na čiernobielej obrazovke (farby sa v tomto prípade zobrazujú ako rôzne odtiene šedej farby). Použili sme príkaz TRAP, ktorý umožňuje spracovať prípad

```
10 REM Program 2.2 - FAREBNE KRUZNICE, 25. 4. 88
20 REM Atari basic
30 DIM Q$(1)
40 GRAPHICS 8:COLOR 1
50 PRINT "Vloz stred kruznice (x,y): ";:INPUT X,Y
60 PRINT "Vloz polomer kruznice r: ";:INPUT R
70 TRAP 300
100 REM Zobrazenie kruznice
110 DEG
120 PLOT X+R,Y
130 FOR I=0 TO 360
140 DRAWTO X+R*COS(I),Y+R*SIN(I)
150 NEXT I
200 REM Dotaz o pokracovani
210 PRINT "Chces dalsiu kruznicu (A, N): ";:INPUT Q$
220 IF Q$(1)="A" THEN 50
230 IF Q$(1)="N" THEN 400
240 PRINT "Nerozumiem..."
250 GOTO 210
300 REM Spracovanie chyby
310 PRINT "Kruznica sa nezmesti na obrazovku!!!"
320 PRINT "Format obrazovky je 360x160"
330 PRINT "Zopakuj!"
340 GOTO 50
400 GRAPHICS 0
410 END
```

nesprávne zadanej polohy kružnice. Texty sa zobrazujú v okienku (posledné 4 riadky textového formátu obrazovky).

Pomocou funkcií SIN a COS môžeme kresliť aj elipsy. Súradnice elipsy vypočítame podľa vzorcov

$x = p.\cos(a) + x_0$

$y = q.\cos(a) + y_0$

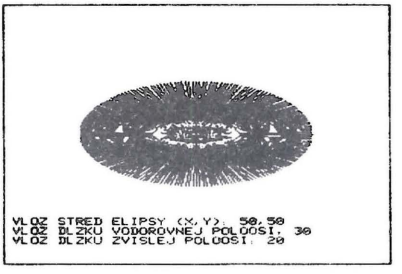
kde (x, y) sú súradnice bodov elipsy, p je dĺžka vodorovnej polosi, q je dĺžka zvislej polosi, (x_0, y_0) je poloha stredu elipsy a uhol a sa mení od 0 do 2π.

PROGRAM 2.3

Program z výpisu 2.3 zobrazí elipsu so zadaným stredom a dĺžkami vodorovnej a zvislej polosi v mierke 0–100 pre vodorovné a zvislé

```
10 REM PROGRAM 2.3 - ELIPSY, 25. 4. 88
20 REM BASIC G (PMD 85, DIDAKTIK ALFA)
30 GCLEAR : SCALE 0,100,0,100
40 PRINT "   VLOZ STRED ELIPSY (X,Y): ";
50 INPUT X,Y
60 PRINT "   VLOZ DLZKU VODOROVNEJ POLOOSI: ";
70 INPUT P
80 PRINT "   VLOZ DLZKU ZVISLEJ POLOOSI: ";
90 INPUT Q
100 GOSUB 200
110 END
200 REM PODPROGRAM KRESLENIA ELIPSY
210 PI=3.14159
220 FOR I=0 TO 2*PI STEP PI/100
230 MOVE X,Y
240 PLOT P*COS(I)+X,Q*SIN(I)+Y
250 NEXT I
260 RETURN
```

súradnice. Elipsu kreslíme v tomto prípade tak, že spájame jej stred s bodmi na jej obvode (pozri obr. 2.2). Ak by sme chceli nakresliť



2.2 Elipsa nakreslená programom 2.3

iba obvod elipsy, vynecháme riadok 230 a riadok 240 upravíme takto:

240 PLOT P*COS(I)+X,Q*SIN(I)+Y,1

Program je určený pre PMD 85 a Didaktik Alfa. Pre Atari si ho upravte sami výmenou programových riadkov 20 a 30 a premenovaním MOVE na PLOT a PLOT na DRAWTO.

2.1.2 PRAVIDELNÉ N-UHOLNÍKY

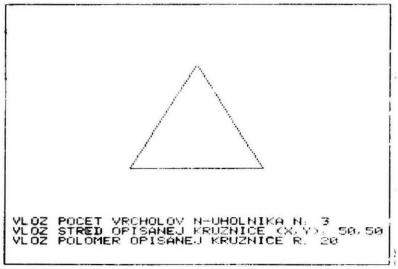
Pravidelný n-uholník môžeme zadať počtom vrcholov a polomerom opísanej kružnice. Na kreslenie nám znovu poslúžia funkcie SIN a COS.

PROGRAM 2.4

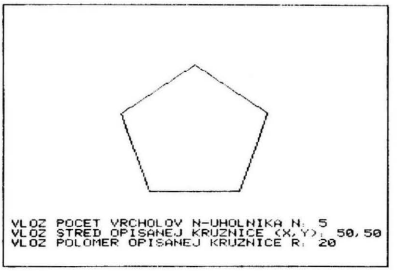
Program z výpisu 2.4 kreslí pravidelné n-uholníky, ktorých tvar, poloha a veľkosť sú zadané počtom vrcholov a stredom a polomerom opísanej kružnice (v mierke 0–100 pre vodorovné a zvislé súradnice). Na obr. 2.3 uvádzame, ako náš program zobrazí rovnostranný trojuholník. Ten istý program nakreslil obrázky 2.4 a 2.5.

```
10 REM PROGRAM 2.4 - N-UHOLNIKY, 26. 4. 88
20 REM BASIC G (PMD 85, DIDAKTIK ALFA)
30 GCLEAR : SCALE 0,100,0,100
40 PRINT "VLOZ POCET VRCHOLOV N-UHOLNIKA N: ";
50 INPUT N
60 PRINT "VLOZ STRED OPISANEJ KRUZNICE (X,Y): ";
70 INPUT X,Y
80 PRINT "VLOZ POLOMER OPISANEJ KRUZNICE R: ";
90 INPUT R
100 REM ZOBRAZENIE N-UHOLNIKA
110 MOVE X,Y+R
120 PI=3.14159
130 FOR I=1 TO N
140 J=PI/2+I*2*PI/N
150 PLOT R*COS(J)+X,R*SIN(J)+Y
160 NEXT I
170 END
```

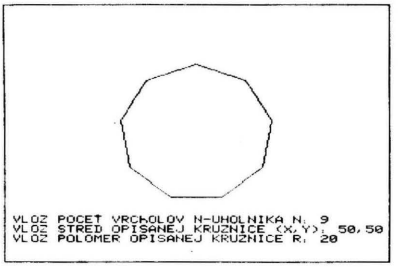
Z obrázkov vidíme, že čím väčší je počet vrcholov, tým viac sa zobrazený n-uholník ponáša na kružnicu. V skutočnosti kružnicu kreslíme ako n-uholník s veľkým počtom vrcholov (v programe 2.1 to bol vlastne n-uholník so 100 vrcholmi). V tom je príčina, že zobrazenia kružníc sú trochu „kostrbaté“. Na obrazovke, ktorá je konečnou tabuľkou zobrazovaných bodov, nie sme vstave nakresliť dokonalý oblúk a oblúky kreslíme príkazom na zobrazovanie úsečiek.



2.3 Trojuholník nakreslený programom 2.4



2.4 Päťuholník nakreslený programom 2.4



2.5 Deväťuholník nakreslený programom 2.4

Keď budete program na kreslenie pravidelných n-uholníkov prepisovať do Atari basicu, nezabudnite, že číslovanie riadkov obrazovky je na Atari opačné ako v bodovom grafickom režime na PMD 85: riadky sa číslujú zhora dole. Mechanické prepísanie programu 2.4 preto bude zobrazovať n-uholníky otočené o 180 stupňov (napr. základňa trojuholníka bude hore).

2.1.3 ŠPIRÁLY A HVIEZDY

Bez funkcií SIN a COS sa nezaobídeme ani pri kreslení špirály. Súradnice špirály vypočítame podľa vzorcov

$x = r.\cos(a) + x_0$

$y = r.\sin(a) + y_0$

kde (x, y) sú súradnice bodov špirály, (x_0, y_0) sú súradnice stredu špirály, uhol a sa mení od 0 do $k.360$ stupňov, kde k je číslo, ktoré udáva počet otáčok špirály, a r sa postupne zväčšuje od 0 po zadanú hodnotu (udáva vzdialenosť bodu špirály od stredu).

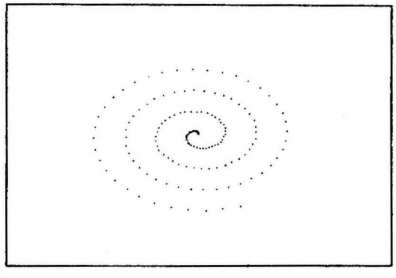
František Turnovec

PROGRAM 2.5

Program z výpisu 2.5 nakreslí špirálu so 4 otáčkami v grafic režime 8 (pozri napr. obr. 2.6). V programe v Atari basicu použili niekoľko nových prvkov: v riadkoch 70–100 kontroluj správnosť zadanie stredu špirály, v riadkoch 280–300 pri kont odpovedí „áno" alebo „nie" pripúšťame veľké aj malé písmc a v riadku 400 sa pred ukončením programu vraciame do základn ho režimu 0.

```
10 REM Program 2.5 - SPIRALA, 26.4.88
20 REM Atari basic
30 DIM Q$(1):Q$(1)=" "
40 GRAPHICS 8:SETCOLOR 1,2,15:COLOR 1
50 PRINT "Zadaj stred spiraly (A,B): ";
60 INPUT X,Y
70 IF X>0 AND X<320 THEN 110
80 IF Y>0 AND Y<160 THEN 110
90 PRINT "Zadal si bod mimo obrazovky"
100 PRINT "Zopakuj!":GOTO 50
110 TRAP 350
120 R=0:DR=0.3:PI=3.14159
200 REM Zobrazenie spiraly
210 PLOT X,Y
220 FOR I=1 TO 24 STEP PI/20
230 R=R+DR
240 PLOT R*COS(I)+X,R*SIN(I)+Y
250 NEXT I
260 PRINT "Chces dalsiu spiralu (A, N): ";
270 INPUT Q$
280 IF Q$(1)="A" OR Q$(1)="a" THEN 50
290 IF Q$(1)="N" OR Q$(1)="n" THEN 400
300 PRINT "Nerozumiem..."
310 GOTO 260
350 REM Spracovanie chyby
360 PRINT "Spirala sa nezmesti na obrazovku!!!"
370 PRINT "Format obrazovky je 320x160"
380 PRINT "Zopakuj!":GOTO 50
400 GRAPHICS 0
410 END
```

Program ľahko samostatne upravíte pre PMD 85 a Didaktik Alfa: nezabudnite nahradiť riadok 40 príkazom SCALE s vhodne zvolenou mierkou, riadok 30 nie je potrebný, na jeho miesto vložte



2.6 Špirála nakreslená programom 2.5

GCLEAR. Na PMD 85–2 namiesto TRAP môžete použiť príkaz ON ERR.

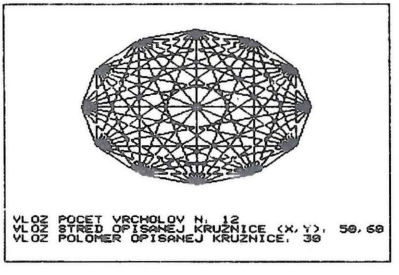
Pomocou vzájomného spájania vrcholov n-uholníkov môžeme kresliť efektné obrázky. tzv. hviezdy.

PROGRAM 2.6

Program z výpisu 2.6 vytvorí pravidelný n-uholník a spojí navzájom všetky jeho vrcholy (každý s každým). Na obr. 2.7 uvádzame

```
10 REM PROGRAM 2.6 - HVIEZDY, 26. 4. 88
20 REM BASIC G (PMD 85, DIDAKTIK ALFA)
30 DIM P(50),Q(50)
40 GCLEAR : SCALE 0,100,0,100
50 PRINT "   VLOZ POCET VRCHOLOV N: ";
60 INPUT N
70 PRINT "   VLOZ STRED OPISANEJ KRUZNICE (X,Y): ";
80 INPUT X,Y
90 PRINT "   VLOZ POLOMER OPISANEJ KRUZNICE: ";
100 INPUT R
110 PI=3.14159
120 FOR I=1 TO N
130 J=PI/2+I*2*PI/N
140 P(I)=R*COS(J)+X
150 Q(I)=R*SIN(J)+Y
160 NEXT I
170 GOSUB 200
180 END
200 REM ZOBRAZENIE HVIEZDY

210 FOR I=1 TO N-1
220 FOR J=I+1 TO N
230 MOVE P(I),Q(I)
240 PLOT P(J),Q(J)
250 NEXT J
260 NEXT I
270 RETURN
```

hviezdu, ktorá vznikne pre n = 12. Nakreslili sme ju na PMD 85. Pre Atari stačí nahradiť riadky 30 a 40 príkazmi GRAPHICS 8 a COLOR 1, MOVE v riadku 230 príkazom PLOT a PLOT v riadku 240 príkazom DRAWTO.



2.7 Hviezda nakreslená programom 2.6

František Turnovec

2.1.4 KOCKA A VALEC

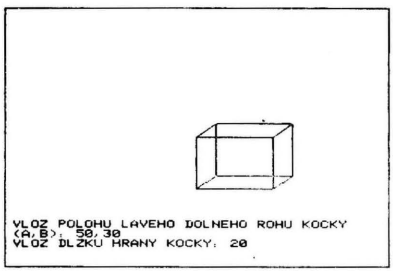
Podobne ako na papieri môžeme aj na obrazovke kresliť obrázky trojrozmerných objektov.

PROGRAM 2.7

Program v basicu G, ktorý uvádzame vo výpise 2.7, zobrazí kocku so zadanou dĺžkou hrany v zadanom mieste obrazovky. Polohu kocky zadávame odpoveďou na dotaz počítača súradnicami jej ľavého dolného rohu v mierke 0 – 100 pre vodorovné a zvislé súradnice.

```
10 REM PROGRAM 2.7 - KOCKA, 28. 4. 88
20 REM BASIC G (PMD 85, DIDAKTIK ALFA)
30 GCLEAR : SCALE 0,100,0,100
40 PRINT "   VLOZ POLOHU LAVEHO DOLNEHO ROHU KOCKY"
45 PRINT "   (A,B): ";
50 INPUT A,B
60 PRINT "   VLOZ DLZKU HRANY KOCKY: ";
70 INPUT X
80 Y=INT(X/4)
100 MOVE A,B
110 PLOT A+X,B
120 PLOT A+X,B+X
130 PLOT A,B+X
140 PLOT A,B
150 PLOT A+Y,B+Y
160 PLOT A+Y+X,B+Y
170 PLOT A+X,B
180 MOVE A+Y+X,B+Y
190 PLOT A+Y+X,B+Y+X
200 PLOT A+Y,B+Y+X
210 PLOT A+Y,B+Y
220 MOVE A,B+X
230 PLOT A+Y,B+X+Y
240 MOVE A+X,B+X
250 PLOT A+X+Y,B+X+Y
260 END
```

V tej istej mierke zadávame dĺžku hrany kocky. Na obr. 2.8 uvádzame jednu z možných kociek, ktoré vieme nakresliť naším programom. Prepíšte program do Atari basicu v grafickom režime 8.



2.8 Kocka nakreslená programom 2.7

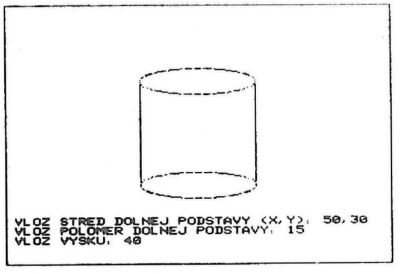
Keďže vieme nakresliť elipsu, sme vstave kresliť aj zložitejšie trojrozmerné obrázky, napr. rotačný valec a rotačný kužeľ.

PROGRAM 2.8

Program v basicu G z výpisu 2.8 nakreslí rotačný valec so zadaným stredom, polomerom podstavy a výškou. Dolnú a hornú podstavu znázorňujeme elipsou, v ktorej vodorovná polos sa rovná polomeru podstavy a zvislá polos sa rovná tretine polomeru podstavy (pozri valec na obr. 2.9).

```
10 REM PROGRAM 2.8 - ROTACNY VALEC, 27. 4. 88
20 REM BASIC G (PMD 85, DIDAKTIK ALFA)
30 GCLEAR : SCALE 0,100,0,100
40 PRINT "   VLOZ STRED DOLNEJ PODSTAVY (X,Y): ";
50 INPUT X,Y
60 PRINT "   VLOZ POLOMER DOLNEJ PODSTAVY: ";
70 INPUT R
80 PRINT "   VLOZ VYSKU: ";
90 INPUT V
100 A=R : B=R/3
110 GOSUB 200
120 MOVE X-R,Y
130 PLOT X-R,Y+V
140 MOVE X+R,Y
150 PLOT X+R,Y+V
160 Y=Y+V
170 GOSUB 200
180 END
200 REM PODPROGRAM KRESLENIA ELIPSY
210 PI=3.14159
220 FOR I=0 TO 2*PI STEP PI/100
230 PLOT A*COS(I)+X,B*SIN(I)+Y,1
240 NEXT I
250 RETURN
```

Prepíšte program do Atari basicu v grafickom režime 8 (nezabudnite pritom na opačné číslovanie riadkov).



2.9 Rotačný valec nakreslený programom 2.8

František Turnovec

2.1.5 KRUHOVÝ POHYB

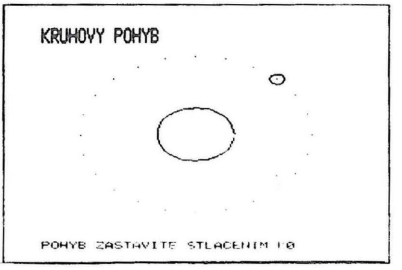
Ako sme už ukázali v kapitole 1.5, kreslenie „negáciou" na PMD 85 a Didaktiku Alfa možno využiť na programovanie pohybu na obrazovke. Postupným kreslením a vymazávaním obrázka a posunom jeho polohy môžeme vytvoriť ilúziu pohybu rýchlym striedaním polôh obrázka na obrazovke. Ukážeme, ako programovať kruhový pohyb.

PROGRAM 2.9

Program v basicu G z výpisu 2.9 riadi zobrazenie pohybu po obežnej dráhe (pozri obr. 2.10). Program najprv vykreslí väčší stredový kruh a potom sa okolo neho proti smeru hodinových ručičiek začne pohybovať menšie kruhové teleso. Pohyb možno kedykoľvek zastaviť stlačením funkčného klávesu KØ alebo stlačením klávesu STOP. Na naprogramovanie kruhového pohybu sme využili už známe vyjadrenie súradníc kružnice pomocou funkcií COS a SIN. Príkazy

```
10 REM PROGRAM 2.9 - KRUHOVY POHYB, 28. 4. 88
20 REM BASIC G (PMD 85, DIDAKTIK ALFA)
30 GCLEAR : SCALE 0,100,0,100
40 PI=3.14159
50 MOVE 10,85
60 LABEL 1,2;"KRUHOVY POHYB"
70 MOVE 10,5
80 LABEL 1,1;"POHYB ZASTAVITE STLACENIM KØ"
90 GOSUB 300
100 FOR I=0 TO 2*PI STEP PI/12
105 IF INKEY=0 THEN 200
110 X=30*COS(I)+50 : Y=30*SIN(I)+50
112 POKE -15878,176
113 PLOT X,Y,1
114 POKE -15878,168
115 MOVE X+2,Y
120 FOR K=0 TO 2*PI STEP PI/5
125 PLOT 2*COS(K)+X,2*SIN(K)+Y
130 NEXT K
135 MOVE X+2,Y
140 FOR K=0 TO 2*PI STEP PI/5
145 PLOT 2*COS(K)+X,2*SIN(K)+Y
150 NEXT K
160 NEXT I
180 GOTO 100
200 END
300 REM VYKRESLENIE KRUHU
305 MOVE 60,50
310 FOR L=0 TO 2*PI STEP PI/20
320 PLOT 10*COS(L)+50,10*SIN(L)+50
330 NEXT L
360 RETURN
```

140 POKE -15878,176
a
160 POKE -15878,168
používame na postupné prepínanie režimu negácie pri vykresľovaní obežnej dráhy (aby sa pri párnych obehoch nemazala už nakreslená obežná dráha).



2.10 Zobrazenie kruhového pohybu programom 2.9

Naznačíme ešte, ako programovať jednoduchý pohyb v Atari basicu. Atari basic nepoužíva kreslenie negáciou predchádzajúceho stavu. Vzhľadom na možnosť používania farieb už nakreslený bod vymažeme tak, že ho znovu vykreslíme a pred znovuvykreslením zmeníme príkazom COLOR farbu pera, aby bola zhodná s farbou pozadia. Napríklad v grafickom režime 8 je farba pera štandardne v registri 1 a farba pozadia v registri 2. Preto pred kreslením použijeme príkaz COLOR 1 a pred vymazávaním príkaz COLOR 2. Pokúste sa na základe týchto poznatkov prepísať program 2.9 do Atari basicu.

2.1.6 GRAFY FUNKCIÍ A DIAGRAMY

Ukážeme, ako zobrazovať rôzne druhy funkčných závislostí.

PROGRAM 2.10

V basicu G môžeme s výhodou využiť príkaz FILL na zvýrazňovanie grafických zobrazení pomocou tzv. „okienok". Program z výpisu 2.10 kreslí grafy paraboly a pomocou okienka zvýrazňuje nezáporný kvadrant súradnicového systému. Na obr. 2.11 a 2.12 uvádzame

príklady zobrazenia dvoch rôznych parabol. Koeficienty *a, b, c* paraboly

$y = ax^2 + bx + c$

vkladáme odpoveďou na úvodné dotazy. Okienko v nezápornej časti súradnicového systému programujeme príkazmi

```
51Ø MOVE Ø,Ø
52Ø FILL 12Ø,14Ø;1
```

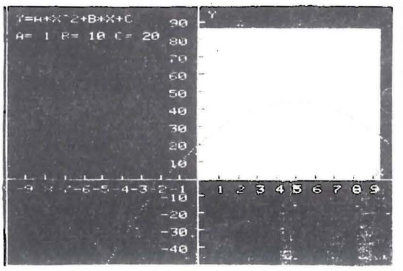
inverziou príslušnej časti obrazovky.

```
10 REM PROGRAM 2.10 - PARABOLY S OKIENKOM, 28. 4. 88
20 REM BASIC G (PMD 85, DIDAKTIK ALFA)
30 PRINT "VLOZ KOEFICIENTY PARABOLY"
40 PRINT "      Y=A*X^2+B*X+C"
50 PRINT
60 PRINT "A = "; : INPUT A
70 PRINT "B = "; : INPUT B
80 PRINT "C = "; : INPUT C
100 REM SURADNICOVE OSI
110 GCLEAR
120 SCALE -10,10,-50,100
130 AXES 0,0
140 MOVE 9.2,2
150 LABEL 1,1;"X"
160 MOVE 0.5,92
170 LABEL 1,1;"Y"
180 FOR I=-9 TO 9
190 IF I=0 THEN 240
200 MOVE I,0
210 FILL 1,4;1
220 MOVE I-0.5,-8
230 LABEL 1,1;I
240 NEXT I
250 FOR J=-40 TO 90 STEP 10
260 IF J=0 THEN 310
270 MOVE 0,J
280 FILL 4,1;1
290 MOVE -2,J-3
300 LABEL 1,1;J
310 NEXT J
400 REM ZOBRAZENIE PARABOLY
410 FOR X=-10 TO 10 STEP 0.1
420 Y=A*X^2+B*X+C
430 IF Y>100 OR Y<-50 THEN 450
440 PLOT X,Y,1
450 NEXT X
460 MOVE -9.5,90
470 LABEL 1,1;"Y=A*X^2+B*X+C"

480 MOVE -9.5,80
490 LABEL 1,1;"A=";STR$(A);" B=";STR$(B);" C=";STR$(C)
500 REM ZOBRAZENIE OKIENKA
510 MOVE 0,0
520 FILL 120,140;1
530 END
```

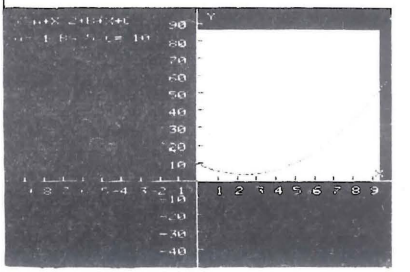
Vyskúšajte si programom PARABOLY, aký vplyv na tvar grafu zobrazovanej kvadratickej funkcie jednej premennej má zmena veľkostí a znamienok jednotlivých koeficientov.

Úprava programu 2.10 pre Atari si vyžiada väčšie zásahy, pretože Atari basic nemá príkaz na kreslenie súradnicových osí a inverziu

obrazovky. Súradnicové osi však ľahko nakreslíme príkazom DRAWTO a zvýrazňovacie okienko vykreslením súvislého priestoru inou farbou, ako je pôvodná farba pozadia.



2.11 Zobrazenie paraboly programom 2.10



2.12 Zobrazenie paraboly programom 2.10

Kvantitatívne závislosti medzi javmi môžu byť vyjadrené aj iným spôsobom ako matematickou funkciou. Na ich zobrazenie využívame rôzne druhy diagramov.

Histogramom nazývame diagram, ktorý porovnáva nejaké číselné hodnoty pomocou rôzne vysokých stĺpcov na vodorovnej osi.

PROGRAM 2.11

Predpokladajme, že chceme graficky znázorniť rozdelenie žiakov v triede do skupín podľa priemerného prospechu:

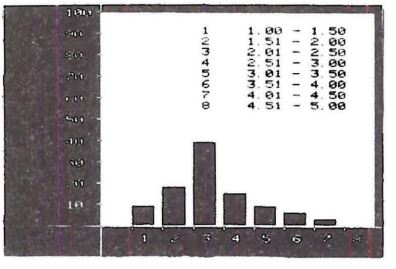
1. 1,00–1,50 3 žiaci
2. 1,51–2,00 6 žiakov
3. 2,01–2,50 13 žiakov
4. 2,51–3,00 5 žiakov
5. 3,01–3,50 3 žiaci
6. 3,51–4,00 2 žiaci
7. 4,01–4,50 1 žiak
8. 4,51–5,00 0 žiakov

Program v basicu G z výpisu 2.11 nakreslí histogram percentuálneho rozdelenia žiakov do prospechových skupín (pozri obr. 2.13).

```
10 REM PROGRAM 2.11 - HISTOGRAM, 29 4. 88
20 REM BASIC G (PMD 85, DIDAKTIK ALFA)
30 GCLEAR : SCALE 0,255,0,242
40 AXES 70,30
```

```
50 FOR I=50 TO 230 STEP 20
60 MOVE 66;I
70 FILL 4,1;1
80 MOVE 40,I-4
90 K=K+10
100 LABEL 1,1;K
110 NEXT I
120 FOR J=90 TO 230 STEP 20
130 MOVE J,26
140 FILL 1,4;1
150 MOVE J,15
170 L=L+1
180 LABEL 1,1;L
190 NEXT J
200 REM PRECITANIE UDAJOV
210 S=0
220 FOR J=1 TO 8
230 READ X(J)
240 S=S+X(J)
250 NEXT J
260 REM VYPOCET PERCENTUALNYCH HODNOT
270 FOR J=1 TO 8
280 P(J)=(X(J)/S)*100
290 NEXT J
300 REM ZOBRAZENIE HISTOGRAMU
310 H=0
320 FOR J=90 TO 230 STEP 20
330 MOVE J,31
340 H=H+1
350 FILL 15,2*P(H);1
360 NEXT J
400 REM POPISANIE STLPCOV
405 JD=1 : JH=1.5
410 FOR J=1 TO 8
420 MOVE 130,220-J*10
430 JD$=LEFT$(STR$(JD+0.0001),5)
440 JH$=LEFT$(STR$(JH+0.0001),5)
450 LABEL 1,1;J;"  ";JD$;" -";JH$
460 JD=JH+0.01
470 JH=JH+0.5
480 NEXT J
500 REM ZOBRAZENIE OKIENKA
510 MOVE 70,30
520 FILL 180,200;1
530 END
600 REM UDAJE
610 DATA 3,6,13,5,3,2,1,0
```

Na vodorovnej osi sú vyznačené prospechové skupiny podľa priemerov (spolu osem skupín), na zvislej osi percentá. Stĺpčeky nášho



2.13 Histogram nakreslený programom 2.11

histogramu vyjadrujú percentuálny podiel jednotlivých prospechových skupín na celkovom počte žiakov v triede. Význam jednotlivých skupín charakterizuje tabuľka v pravej hornej časti diagramu.

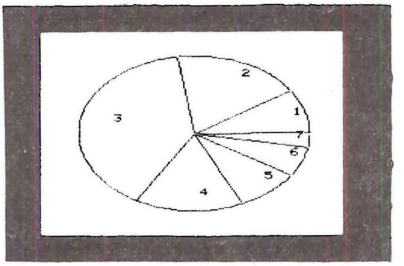
Percentuálnu skladbu môžeme vyjadriť aj tzv. *koláčovým diagramom.* Percentuálny podiel jednotlivých skupín je tu vyjadrený veľkosťou dielov „koláča".

PROGRAM 2.12

Na obr. 2.14 uvádzame „koláčový" diagram percentuálnych podielov jednotlivých prospechových skupín v triede, nakreslený programom z výpisu 2.12. Použili sme rovnaké údaje ako v programe 2.11. Diely „koláča" sú usporiadané proti smeru hodinových ručičiek (čísla jednotlivých dielov zodpovedajú číslam prospechových skupín). Po zodpovedajúcej zmene začiatku programu (náhrada príkazov GCLEAR a SCALE), zmene príkazov MOVE a PLOT na PLOT a DRAWTO a vynechaní príkazov zobrazenia okienka bude

```
10 REM PROGRAM 2.12 - KOLACOVY DIAGRAM, 29 4. 88
20 REM BASIC G (PMD 85, DIDAKTIK ALFA)
30 DIM X(8),P(8)
40 GCLEAR : SCALE 0,100,0,100
50 PI=3.14159
60 X=50 : Y=50 : R=30
70 REM ZOBRAZENIE KRUZNICE
80 MOVE X+R,Y
90 FOR I=0 TO 2*PI STEP PI/100
100 PLOT R*COS(I)+X,R*SIN(I)+Y
110 NEXT I
120 REM PRECITANIE UDAJOV
130 S=0
140 FOR J=1 TO 8
150 READ X(J)
160 S=S+X(J)
170 NEXT J
180 REM VYPOCET PERCENTUALNYCH HODNOT
190 FOR J=1 TO 8
200 P(J)=(X(J)/S)*100
210 NEXT J
220 REM ZOBRAZENIE DIELKOV "KOLACA"
225 A=0
230 FOR J=1 TO 8
235 IF P(J)=0 THEN 310
240 MOVE X,Y
250 A=A+((2*PI)/100)*P(J)
260 PLOT R*COS(A)+X,R*SIN(A)+Y
270 B=0.5*((2*PI)/100)*P(J)
290 MOVE 0.9*R*COS(A-B)+X,0.8*R*SIN(A-B)+Y
300 LABEL 1,1;J
310 NEXT J
320 REM ZOBRAZENIE OKIENKA
330 MOVE 10,10
340 FILL 200,192;1
350 END
400 REM UDAJE
410 DATA 3,6,13,5,3,2,1,0
```

program fungovať aj na počítačoch Atari (pri prepise do Atari basicu použijeme grafický režim 8).



2.14 „Koláčový" diagram nakreslený programom 2.12

2.1.7 PORTRÉTY A KARIKATÚRY

V jazyku basic môžeme programovať aj kreslenie obrázkov. Na ilustráciu vám ponúkame programy, ktoré zobrazujú ľudskú tvár. Vychádzajú z určenia polohy bodov, ktorých spojením dostávame základné črty tváre. Zvolili sme 39 takýchto čŕt: polohu ľavej zreničky, pravej zreničky, polohu a tvar ľavej a pravej dúhovky, dolný okraj ľavej a pravej mihalnice atď. Pre každú z týchto čŕt treba určiť súradnice jedného alebo viacerých bodov, ktorých zobrazením alebo spojením vykreslíme príslušný prvok tváre. Napríklad zreničku zobrazujeme jediným bodom, zatiaľ čo dúhovku postupným spojením piatich bodov.

Číselné súradnice zidealizovanej tváre sme určili podľa obrázka typickej zovšeobecnenej tváre z časopisu *Scientific American* (október 1986).

PROGRAM 2.13

Program z výpisu 2.13 nám na obrazovke nakreslí zidealizovanú ľudskú tvár (pozri obr. 2.15). Číselné súradnice jednotlivých čŕt tváre uvádzame na konci programu v príkazoch DATA. Pred každým príkazom DATA (alebo skupinou príkazov DATA) obsahujúcim súradnice určitej črty uvádzame v komentárovom príkaze REM slovnú charakteristiku príslušnej črty. Údaje v príkazoch DATA sú usporiadané takto: prvé číslo skupiny údajov uvádza počet bodov, ktoré sa majú pri zobrazení danej črty spojiť; ďalej nasledujú dvojice

```
10 REM  PROGRAM 2.13 - PORTRET, 29. 4. 88
20 REM BASIC G (PMD 85, DIDAKTIK ALFA)
30 SCALE 0,255,0,242
50 GCLEAR
60 GOSUB 1000
70 GOSUB 1200
100 END
1000 REM PODPROGRAM ZOBRAZENIA TVARE
1010 READ N
1015 IF N=0 THEN RETURN
1020 READ X,Y
1030 MOVE X,Y
1040 IF N>1 THEN 1070
1050 PLOT X,Y,1
1060 GOTO 1010
1070 FOR I=2 TO N
1080 READ X,Y
1090 PLOT X,Y
1100 NEXT I
1110 GOTO 1010
1200 REM INVERZIA
1210 MOVE 0,0
1220 FILL 255,242,1
1230 RETURN
2000 REM SURADNICE JEDNOTLIVYCH CRT TVARE
2005 REM LAVA ZRENICKA
2010 DATA 1,95,118
2015 REM PRAVA ZRENICKA
2020 DATA 1,150,118
2025 REM LAVA DUHOVKA
2030 DATA 5,94,122,88,119,93,114,100,119,95,122
2035 REM PRAVA DUHOVKA
2040 DATA 5,150,122,144,119,149,114,156,119,150,122
2045 REM DOLNY OKRAJ LAVEJ MIHALNICE
2050 DATA 3,79,116,93,123,107,117
2055 REM DOLNY OKRAJ PRAVEJ MIHALNICE
2060 DATA 3,137,116,150,123,163,116
2065 REM DOLNY OKRAJ LAVEHO OKA
2070 DATA 3,81,116,93,113,107,117
2075 REM DOLNY OKRAJ PRAVEHO OKA
2080 DATA 3,137,116,151,113,161,115
2085 REM LAVA NADOCNA CIARA
2090 DATA 3,78,120,92,126,108,121
2095 REM PRAVA NADOCNA CIARA
2100 DATA 3,136,120,151,126,164,120
2105 REM LAVA PODOCNA CIARA
2110 DATA 3,87,109,95,110,104,113
2115 REM PRAVA PODOCNA CIARA
2120 DATA 3,138,112,147,109,156,109
2125 REM LAVA STRANA NOSU
2130 DATA 6,116,123,116,110,116,98,114,91,116,84,121,81
2135 REM PRAVA STRANA NOSU
2140 DATA 6,126,123,126,110,126,97,128,91,127,84,121,81
2145 REM LAVA NOSOVA DIERKA
2150 DATA 6,110,94,107,90,106,85,108,81,113,84,121,81
2155 REM PRAVA NOSOVA DIERKA
2160 DATA 6,133,94,136,91,137,85,134,81,130,84,123,81
2165 REM HORNY OKRAJ LAVEHO OBOCIA
2170 DATA 6,72,126,73,131,85,136,99,135,110,132,112,127
2175 REM HORNY OKRAJ PRAVEHO OBOCIA
2180 DATA 6,131,127,133,131,146,134,159,135,168,131,171,126
2185 REM DOLNY OKRAJ LAVEHO OBOCIA
2190 DATA 4,72,125,84,131,98,129,112,127
2195 REM DOLNY OKRAJ PRAVEHO OBOCIA
2200 DATA 4,131,127,147,129,160,131,170,126
2205 REM HORNY OBRYS HORNEJ PERY
2210 DATA 7,97,60,109,64,116,67,122,64,128,66,137,64,147,61
2215 REM DOLNY OBRYS HORNEJ PERY
2220 DATA 7,98,60,108,60,116,61,123,60,130,61,138,60,146,61
2225 REM HORNY OBRYS DOLNEJ PERY
2230 DATA 7,98,60,109,59,116,60,123,61,130,59,137,61,146,60
```

```
2235 REM DOLNY OBRYS DOLNEJ PERY
2240 DATA 7,101,59,108,56,115,53,123,52,131,53,139,56,145,60
2245 REM LAVY OKRAJ OBLICEJA
2250 DATA 3,63,122,61,103,64,82
2255 REM PRAVY OKRAJ OBLICEJA
2260 DATA 3,179,123,182,104,178,84
2265 REM LAVE UCHO
2270 DATA 7,59,113,52,119,48,114,50,103,54,89,59,76,64,79
2275 REM PRAVE UCHO
2280 DATA 7,184,114,191,119,194,112,192,103,190,90,184,78,179,79
2285 REM CELUST
2290 DATA 11,64,82,68,64,75,49,89,35,107,23,122,20,140,24
2300 DATA    156,35,167,48,175,64,179,85
2305 REM LINIA CELA
2310 DATA 13,61,119,67,134,74,149,80,159,91,168,106,171,120,170
2320 DATA    134,168,148,167,161,160,170,149,177,137,182,120
2325 REM HORNA LINIA VLASOV
2330 DATA 13,53,59,38,90,36,121,42,162,59,193,89,217,118,219
2340 DATA    146,218,177,199,196,169,205,129,210,95,193,63
2345 REM LAVA LICNA KOST
2350 DATA 3,105,88,99,81,95,73
2355 REM PRAVA LICNA KOST
2360 DATA 3,138,87,145,80,150,72
2365 REM LAVA CELUSTNA KOST
2370 DATA 3,65,85,69,79,72,73
2375 REM PRAVA CELUSTNA KOST
2380 DATA 3,178,85,174,80,171,74
2385 REM LAVÁ ZVISLA CIARA NAD HORNOU PEROU
2390 DATA 2,119,77,119,70
2395 REM PRAVA ZVISLA CIARA NAD HORNOU PEROU
2400 DATA 2,125,77,125,70
2405 REM ZVISLA CIARA NA BRADE
2410 DATA 2,122,31,122,25
2415 REM CIARA POD DOLNOU PEROU
2420 DATA 3,113,45,122,47,133,44
2430 DATA 0
```

čísel, ktoré udávajú súradnice týchto bodov (prvé z dvojice čísel udáva vodorovnú a druhé zvislú súradnicu). Napríklad príkaz

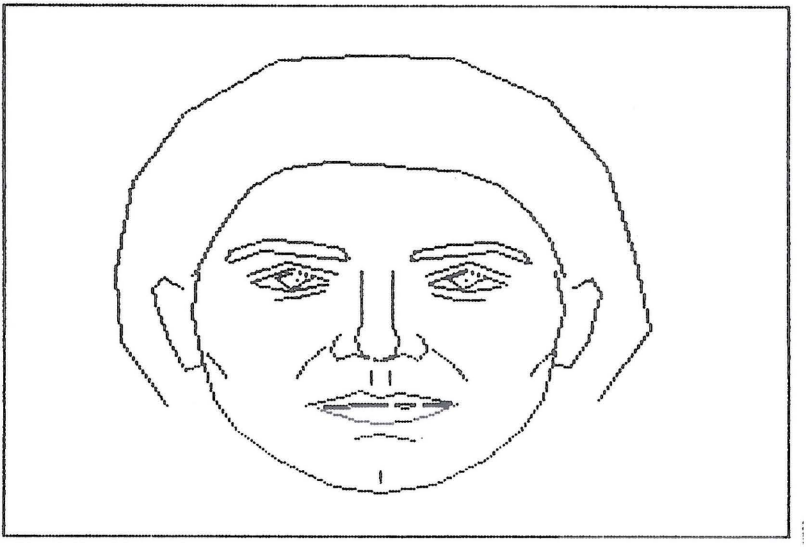
2Ø1Ø DATA 1,95,118

obsahuje súradnice jedného bodu s vodorovnou súradnicou 95 a zvislou súradnicou 118 (zobrazí sa jeden bod – ľavá zrenička). Príkaz

2Ø3Ø DATA 5,94,122,88,119,93,114,1ØØ,119,95,122

obsahuje súradnice piatich bodov: (94, 122), (88, 119), (93, 114), (100, 119) a (95, 122), ktorých postupným spojením v uvedenom poradí zobrazíme ľavú dúhovku.

Program je určený pre počítače PMD 85 a Didaktik Alfa. Pri jeho prepise do Atari basicu treba okrem štandardných zmien (grafický režim 8, zmena príkazov MOVE a PLOT na PLOT a DRAWTO) prepočítať tiež súradnice čŕt tváre, pretože číslovanie riadkov obrazovky je v Atari basicu opačné (zhora dole), teda v našich súradniciach by sa tvár zobrazila naopak (ústa hore, oči dole), a okrem toho grafický režim 8 (aj pri vynechaní textového okienka príkazom GRAPHICS 8+16) poskytuje iba 192 riadkov na rozdiel od 243 riadkov na PMD 85.



2.15 Portrét nakreslený programom 2.13

Starostlivou zmenou súradníc jednotlivých čŕt (najlepšie na štvorčekovom papieri podľa fotografie alebo zjednodušenej kresby) sa môžete pokúsiť o zobrazenie tváre konkrétnej osoby. Okrem trpezlivého merania si to žiada aj veľa pokusov, pretože iba podľa výsledku zobrazenia na obrazovke môžete posúdiť, či sa vám podarilo správne vystihnúť tú-ktorú črtu a celkovú podobu.

Už raz vložené črty tváre môžeme použiť na tvorbu karikatúr konkrétnej zobrazenej osoby. Dosiahneme to bez nového vkladania pozmenených čŕt ich posunmi alebo deformáciami, riadenými programom.

PROGRAM 2.14

V programe z výpisu 2.14 sme použili najjednoduchší spôsob vytvárania variantov karikovanej podoby. Posuny základných charakteristických čŕt môžeme uskutočniť predovšetkým symetrickým zúžením alebo rozšírením všetkých prvkov tváre vo vodorovnom a zvislom smere. Ďalšou možnosťou je nesúmerná náhodná zmena jednotlivých čŕt. Program poskytuje tieto tri možnosti:

1. Zobraziť nedeformovanú tvár v tom tvare, v akom ju zobrazoval program 2.13.
2. Symetricky zúžiť alebo rozšíriť zobrazenie tváre vo vodorovnom a zvislom smere.
3. Náhodne deformovať jednotlivé črty tváre.

```
10 REM  PROGRAM 2.14 - KARIKATURY, 30. 4. 88
20 REM BASIC G (PMD 85, DIDAKTIK ALFA)
30 SCALE 0,255,0,242
50 GCLEAR
60 GOSUB 1750
70 IF L=0 THEN 200
80 ON L GOSUB 300,400,600
100 GCLEAR
140 GOSUB 1000
150 PAUSE 50
160 GOTO 50
200 END
300 REM NEDEFORMOVANE ZOBRAZENIE
310 DEF FNCA(X)=X
320 DEF FNCB(Y)=Y
330 SCALE 0,255,0,242
340 RETURN
400 REM ZUZENIE-ROZSIRENIE TVARE
410 GCLEAR
420 DEF FNCA(X)=A*(X-127)
430 DEF FNCB(Y)=B*(Y-121)
440 PRINT "ZADAJTE VODOROVNE ZUZENIE (ROZSIRENIE)"
450 PRINT "0.5<A<1.0    ZUZENIE"
460 PRINT "1.0<A<1.5    ROZSIRENIE"
470 PRINT
480 PRINT "A =";  : INPUT A
490 PRINT "ZADAJTE ZVISLE ZUZENIE (ROZSIRENIE)"
500 PRINT "0.5<B<1.0    ZUZENIE"
510 PRINT "1.0<B<1.5    ROZSIRENIE"
520 PRINT
530 PRINT "B =";  : INPUT B
540 IF A<0.5 OR A>1.5 THEN A=1.2
550 IF B<0.5 OR B>1.5 THEN B=0.8
560 SCALE -127,127,-121,121
570 RETURN
600 REM NAHODNA DEFORMACIA TVARE
610 GCLEAR
620 DEF FNCA(X)=A*X+(1-A)*INT(255*RND(1)+1)
630 DEF FNCB(Y)=B*Y+(1-B)*INT(242*RND(1)+1)
640 PRINT "ZADAJTE VODOROVNU NAHODNU DEFORMACIU"
650 PRINT "(0.95<A<1.00)
660 PRINT
670 PRINT "A =";  : INPUT A
680 PRINT
690 PRINT "ZADAJTE ZVISLU NAHODNU DEFORMACIU"
700 PRINT "(0.95<B<1.00)
710 PRINT
720 PRINT "B =";  : INPUT B
730 IF A<0.95 OR A>1 THEN A=0.99
740 IF B<0.95 OR B>1 THEN B=0.99
750 SCALE 0,255,0,242
760 RETURN
1000 REM PODPROGRAM ZOBRAZENIA TVARE
1005 RESTORE
1010 READ N
1015 IF N=0 THEN RETURN
1020 READ X,Y
1030 MOVE FNCA(X),FNCB(Y)
1040 IF N>1 THEN 1070
1050 PLOT FNCA(X),FNCB(Y),1
```

```
1060 GOTO 1010
1070 FOR I=2 TO N
1080 READ X,Y
1090 PLOT FNCA(X),FNCB(Y)
1100 NEXT I
1110 GOTO 1010
1750 REM PONUKA MOZNOSTI
1755 SCALE 0,255,0,242
1760 MOVE 10,220
1770 LABEL 1,1;"MATE TIETO MOZNOSTI:"
1780 MOVE 20,200
1790 LABEL 1,1;"1. ZOBRAZIT NEDEFORMOVANU TVAR"
1800 MOVE 20,190
1810 LABEL 1,1;"2. ZUZIT ALEBO ROZSIRIT ZOBRAZENIE"
1820 MOVE 20,180
1830 LABEL 1,1;"3. NAHODNE DEFORMOVAT CRTY TVARE"
1840 MOVE 10,100
1850 LABEL 1,1;"VLOZTE PORADOVE CISLO ZVOLENEJ MOZNOSTI;
1860 MOVE 10,90
1870 LABEL 1,1;"AK CHCETE UKONCIT PRACU S PROGRAMOM,"
1880 MOVE 10,80
1890 LABEL 1,1;"VLOZTE CISLO 0!"
1900 INPUT L
1910 IF L<0 OR L>3 THEN 1900
1920 RETURN
2000 REM SURADNICE JEDNOTLIVYCH CRT TVARE
2005 REM LAVA ZRENICKA
2010 DATA 1,95,118
2015 REM PRAVA ZRENICKA
2020 DATA 1,150,118
2025 REM LAVA DUHOVKA
2030 DATA 5,94,122,88,119,93,114,100,119,95,121
2035 REM PRAVA DUHOVKA
2040 DATA 5,150,122,144,119,149,114,156,119,150,122
2045 REM DOLNY OKRAJ LAVEJ MIHALNICE
2050 DATA 3,79,116,93,123,107,117
2055 REM DOLNY OKRAJ PRAVEJ MIHALNICE
2060 DATA 3,137,116,150,123,163,116
2065 REM DOLNY OKRAJ LAVEHO OKA
2070 DATA 3,81,116,93,113,107,117
2075 REM DOLNY OKRAJ PRAVEHO OKA
2080 DATA 3,137,116,151,113,161,115
2085 REM LAVA NADOCNA CIARA
2090 DATA 3,78,120,92,126,108,121
2095 REM PRAVA NADOCNA CIARA
2100 DATA 3,136,120,151,126,164,120
2105 REM LAVA PODOCNA CIARA
2110 DATA 3,87,109,95,110,104,113
2115 REM PRAVA PODOCNA CIARA
2120 DATA 3,138,112,147,109,156,109
2125 REM LAVA STRANA NOSU
2130 DATA 6,116,123,116,110,116,98,114,91,116,84,121,81
2135 REM PRAVA STRANA NOSU
2140 DATA 6,126,123,126,110,126,97,128,91,127,84,121,81
2145 REM LAVA NOSOVA DIERKA
2150 DATA 6,110,94,107,90,106,85,108,81,113,84,121,81
2155 REM PRAVA NOSOVA DIERKA
2160 DATA 6,133,94,136,91,137,85,134,81,130,84,123,81
2165 REM HORNY OKRAJ LAVEHO OBOCIA
2170 DATA 6,72,126,73,131,85,136,99,135,110,132,112,127
2175 REM HORNY OKRAJ PRAVEHO OBOCIA
2180 DATA 6,131,127,133,131,146,134,159,135,168,131,171,126
2185 REM DOLNY OKRAJ LAVEHO OBOCIA
2190 DATA 4,72,125,84,131,98,129,112,127
2195 REM DOLNY OKRAJ PRAVEHO OBOCIA
2200 DATA 4,131,127,147,129,160,131,170,126
2205 REM HORNY OBRYS HORNEJ PERY
2210 DATA 7,97,60,109,64,116,67,122,64,128,66,137,64,147,61
2215 REM DOLNY OBRYS HORNEJ PERY
2220 DATA 7,98,60,108,60,116,61,123,60,130,61,138,60,146,61
```

```
2225 REM HORNY OBRYS DOLNEJ PERY
2230 DATA 7,98,60,109,59,116,60,123,61,130,59,137,61,146,60
2235 REM DOLNY OBRYS DOLNEJ PERY
2240 DATA 7,101,59,108,56,115,53,123,52,131,53,139,56,145,60
2245 REM LAVY OKRAJ OBLICAJA
2250 DATA 3,63,122,61,103,64,82
2255 REM PRAVY OKRAJ OBLICAJA
2260 DATA 3,179,123,182,104,178,84
2265 REM LAVE UCHO
2270 DATA 7,59,113,52,119,48,114,50,103,54,89,59,76,64,79
2275 REM PRAVE UCHO
2280 DATA 7,184,114,191,119,194,112,192,103,190,90,184,78,179,79
2285 REM CELUST
2290 DATA 11,64,82,68,64,75,49,89,35,107,23,122,20,140,24
2300 DATA    156,35,167,48,175,64,179,85
2305 REM LINIA CELA
2310 DATA 13,61,119,67,134,74,149,80,159,91,168,106,171,120,170
2320 DATA    134,168,148,167,161,160,170,149,177,137,182,120
2325 REM HORNA LINIA VLASOV
2330 DATA 13,53,59,38,90,36,121,42,162,59,193,89,217,118,219
2340 DATA    146,218,177,199,196,169,205,129,210,95,193,63
2345 REM LAVA LICNA KOST
2350 DATA 3,105,88,99,81,95,73
2355 REM PRAVA LICNA KOST
2360 DATA 3,138,87,145,80,150,72
2365 REM LAVA CELUSTNA KOST
2370 DATA 3,65,85,69,79,72,73
2375 REM PRAVA CELUSTNA KOST
2380 DATA 3,178,85,174,80,171,74
2385 REM LAVA ZVISLA CIARA NAD HORNOU PEROU
2390 DATA 2,119,77,119,70
2395 REM PRAVA ZVISLA CIARA NAD HORNOU PEROU
2400 DATA 2,125,77,125,70
2405 REM ZVISLA CIARA NA BRADE
2410 DATA 2,122,31,122,25
2415 REM CIARA POD DOLNOU PEROU
2420 DATA 3,113,45,122,47,133,44
2430 DATA 0
```

Po odštartovaní programu sa na obrazovke zjaví ponuka týchto troch možností. Vložením poradového čísla určíme, s ktorou z nich chceme pracovať (ak chceme ukončiť činnosť programu, vložíme číslo 0). Ak volíme druhú alebo tretiu možnosť, počítač nás vyzve, aby sme vložili hodnoty koeficientov A a B, ktorými riadime stupeň deformácie zobrazovanej tváre.

V prípade symetrického zúženia alebo rozšírenia určuje koeficient A stupeň vodorovného zúženia alebo rozšírenia. Ak $A < 1$, zobrazenie tváre sa zúži v smere od pravého a ľavého okraja obrazovky do stredu; ak $A > 1$, zobrazenie sa v tomto smere rozšíri. Koeficient B určuje stupeň zvislého zúženia alebo rozšírenia: ak $B < 1$, zobrazenie tváre sa zúži v smere od horného a dolného okraja obrazovky do stredu, ak $B > 1$, zobrazenie sa v tomto smere rozšíri.

Stupeň náhodnej deformácie riadime takto: Pre každý bod zobrazovanej tváre, zadaný v príkazoch DATA na základe „jemného" merania, počítač vytvára paralelný náhodný bod (môže ním byť

ľubovoľný bod obrazovky). Výsledná podoba vzniká kombináciou vložených údajov a náhodne generovaných údajov tak, že príslušná vodorovná súradnica bodu deformovaného zobrazenia je kombináciou presnej a náhodnej súradnice

A*X + (1–A)*P

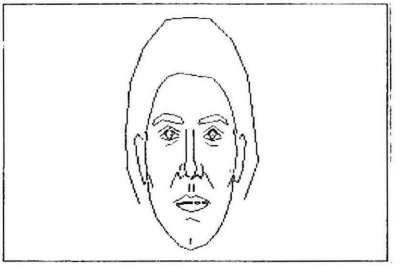
kde *X* je presná súradnica a *P* je náhodne vytvorená súradnica. Príslušná zvislá súradnica bodu deformovaného zobrazenia je kombináciou

B*Y + (1–B)*Q

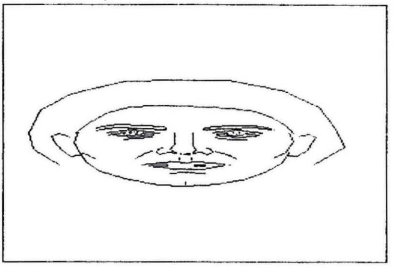
kde *Y* je presná a *Q* je náhodne vytvorená súradnica. Z toho vyplýva, že čím viac sa hodnoty *A* a *B* blížia k 1, tým bližšie je deformované zobrazenie k pôvodnému.

Na tvorbu náhodných súradníc používame funkciu RND (pozri riadky 620 a 630, v ktorých sú definované funkcie FNCA a FNCB na tvorbu súradníc náhodnej deformácie tváre).

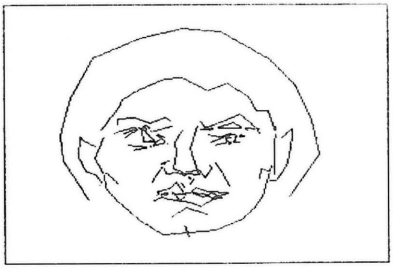
Na obrázkoch 2.16, 2.17 a 2.18 uvádzame ukážky karikatúr vy-



2.16 Karikatúra nakreslená programom 2.14 (deformácia zúžením)



2.17 Karikatúra nakreslená programom 2.14 (deformácia rozšírením)



2.18 Karikatúra nakreslená programom 2.14 (náhodná deformácia)

tvorených naším programom. Prvé dve karikatúry vznikli symetrickým zúžením a rozšírením, ďalšia ilustruje náhodne zdeformovanú tvár. Voľbou rôznych hodnôt A a B v jednotlivých prípadoch získame rôzne karikatúry tej istej tváre.

2.2 JEDENÁSŤ HIER S POČÍTAČOM

Hry s počítačom sú rozšírenou zábavou detí a často aj dospelých. Prostredníctvom hier sa ľahšie učíme „konverzovať" s počítačom a získavame aj užitočné návyky – pohotovosť, schopnosť rýchle sa rozhodovať, sústrediť sa. Vybrali sme pre vás niekoľko programov počítačových hier, o ktorých veríme, že vás zaujmú a že vám zároveň poslúžia aj na prehĺbenie programátorských zručností.

2.2.1 HRA POKLAD

Na obrazovke sa zobrazí tabuľka 10 × 10. V jednom z jej políčok počítač „ukryje" poklad. Našou úlohou je s čo najmenším počtom pokusov tento poklad nájsť. Políčko, v ktorom hľadáme poklad, zadávame vložením čísla riadka a stĺpca tabuľky. V políčku, ktoré sme zvolili, sa zjaví otáznik (ak sme neuhádli polohu pokladu), alebo výkričník (ak sme našli správne políčko s pokladom). V prvom prípade nám počítač oznámi smer ďalšieho hľadania podľa svetových strán a znovu nás vyzve, aby sme vložili číslo riadka a stĺpca, v ktorom očakávame políčko s pokladom. Ak sme v rámci piatich pokusov odhalili ukrytý poklad, počítač nám oznámi náš úspech a ukončí hru. Ak sa nám nepodarilo nájsť poklad ani po piatom pokuse, počítač ponúkne zopakovanie hry. Pri každom novom kole hry sa úkryt pokladu mení.

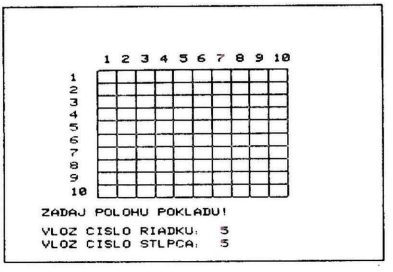
PROGRAM 2.15

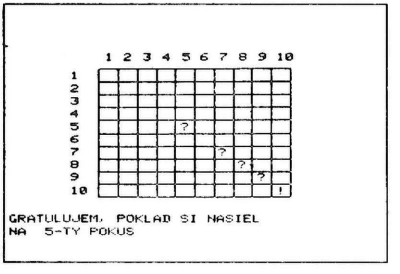
Program, ktorý riadi priebeh hry POKLAD, uvádzame vo výpise 2.15. Všimnite si spôsob zobrazenia tabuľky v riadkoch 1600 až 1770, môže sa vám zísť aj na iné účely. Zobrazované tabuľky spolu s konverzáciou s počítačom uvádzame na obrázkoch 2.19, 2.20 a 2.21.

```
10 REM PROGRAM 2.15 - POKLAD 10. 4. 88
20 REM BASIC G (PMD 85, DIDAKTIK ALFA)
50 GOSUB 1000
```

```
90 GOSUB 1600
100 A=INT(10*RND(1)+1)
110 B=INT(10*RND(1)+1)
210 MOVE 20,30
220 LABEL 1,1;"ZADAJ POLOHU POKLADU!"
230 FOR K=1 TO 5
250 DISP "VLOZ CISLO RIADKU: " : INPUT X
260 IF X<1 OR X>10 THEN GOTO 250
270 DISP "VLOZ CISLO STLPCA: " : INPUT Y
280 IF Y<1 OR Y>10 THEN GOTO 270
290 IF X=A AND Y=B THEN GOTO 400
300 GOSUB 700
310 NEXT K
320 GOSUB 900
400 IF K=1 THEN L$=STR$(K)+"-VY"
410 IF K=2 THEN L$=STR$(K)+"-HY"
420 IF K=3 THEN L$=STR$(K)+"-TI"
430 IF K>3 THEN L$=STR$(K)+"-TY"
450 MOVE 50+Y*10-8,150-X*10+2
460 LABEL 1,1;"!"
470 MOVE 20,30
480 LABEL 1,1;"ZADAJ POLOHU POKLADU!"
490 MOVE 0,30
500 LABEL 1,1;"GRATULUJEM, POKLAD SI NASIEL NA ";L$;" POKUS!"
510 GOTO 950
700 REM OZNAMENIE SMERU
705 MOVE 50+Y*10-8,150-X*10+2
710 LABEL 1,1;"?"

715 IF K=5 THEN RETURN
720 IF Y=B AND X<A THEN K0$="JUH"
730 IF Y=B AND X>A THEN K0$="SEVER"
740 IF Y<B AND X<A THEN K0$="JUHOVYCHOD"
750 IF Y<B AND X>A THEN K0$="SEVEROVYCHOD"
760 IF X=A AND Y<B THEN K0$="VYCHOD"
770 IF X=A AND Y>B THEN K0$="ZAPAD"
780 IF X<A AND Y>B THEN K0$="JUHOZAPAD"
790 IF X>A AND Y>B THEN K0$="SEVEROZAPAD"
810 MOVE 20,10
820 LABEL 1,1;"NEUHADOL SI. CHOD NA ";K0$
830 PAUSE 30
840 MOVE 20,10
850 LABEL 1,1;"NEUHADOL SI. CHOD NA ";K0$
860 RETURN
900 REM PODPROGRAM KONCA HRY
910 MOVE 20,30
920 LABEL 1,1;"ZADAJ POLOHU POKLADU!"
930 MOVE 0,30
940 LABEL 1,1;"LUTUJEM, TOTO JE TVOJ 5-TY NEUSPESNY POKUS"
950 DISP "CHCES HRAT ZNOVU? (A, ALEBO N): "
960 INPUT Q0$
970 IF Q0$="A" THEN GOTO 90
980 GOSUB 1900
1000 REM HLAVICKA PROGRAMU
1010 GCLEAR
1020 SCALE 0,200,0,200
1030 MOVE 70,180
1040 LABEL 2,2;"H R A"
1090 MOVE 20,100
1100 LABEL 3,9;"P O K L A D"
1110 MOVE 20,80
1120 LABEL 1,1;"TREBA NAJST POKLAD, KTORY JE UKRYTY"
1130 MOVE 20,70
1140 LABEL 1,1;"V JEDNOM Z POLICIEK TABULKY 10 X 10."
1150 MOVE 20,60
1160 LABEL 1,1;"POLOHU POKLADU ZADAVATE VLOZENIM "
1170 MOVE 20,50
1180 LABEL 1,1;"CISLA RIADKU A STLPCA TABULKY. NA"
1190 MOVE 20,40
1200 LABEL 1,1;"NAJDENIE POKLADU MATE 5 POKUSOV. PO"
1210 MOVE 20,30
```

```
1220 LABEL 1,1;"KAZDOM NEUSPESNOM POKUSE VAM OZNA-"
1230 MOVE 20,20
1240 LABEL 1,1;"MIM SMER DALSIEHO HLADANIA."
1250 MOVE 60,2
1260 LABEL 1,2;"VELA USPECHOV!"
1270 DISP "                                STLAC KLAVES EOL!"
1280 INPUT Q0$
1300 RETURN
1600 REM ZOBRAZENIE TABULKY
1610 GCLEAR
1630 FOR I=0 TO 10
1640 MOVE 50,150-I*10
1650 PLOT 150,150-I*10
1660 IF I=0 THEN 1690
1670 MOVE 30,150-I*10+2
1680 LABEL 1,1;I
1690 NEXT I
1700 FOR J=0 TO 10
1710 MOVE 50+J*10,150
1720 PLOT 50+J*10,50
1730 IF J=10 THEN 1760
1740 MOVE 50+J*10-2,156
1750 LABEL 1,1;J+1
1760 NEXT J
1770 RETURN
1900 REM KONIEC PROGRAMU
1910 GCLEAR
1920 MOVE 20,180
1930 LABEL 1,1;"KONIEC PROGRAMU"
1940 MOVE 20,100
1950 LABEL 3,9;"P O K L A D"
1960 MOVE 50,30
1970 LABEL 2,2;"DO VIDENIA"
2000 END
```





2.19–2.21 Dialóg s programom POKLAD

2.2.2 HRA HANOJSKÉ VEŽE

Logická hra známa v literatúre o rekreačnej matematike pod názvom HANOJSKÉ VEŽE pochádza z Juhovýchodnej Ázie. Je s ňou spojená táto legenda:

V Hanoji kedysi pôsobilo tajné spoločenstvo mníchov, ktoré vlastnilo tri veže. Na každú vežu bolo možné vložiť 64 rôzne veľkých zlatých kotúčov, vždy menší kotúč na väčší. Žiadne dva kotúče neboli rovnaké. Pôvodne boli všetky kotúče uložené na ľavej z troch susedných veží. Celé generácie mníchov postupne prenášali kotúče z ľavej veže na pravú – s využitím prostrednej veže – tak, že pohnúť mohli vždy iba jedným kotúčom, pričom nesmeli položiť väčší kotúč na menší. Podľa legendy mal po prenesení všetkých kotúčov z ľavej veže na pravú nastať koniec sveta (skúste vypočítať, koľko rokov by trval svet hanojských mníchov, ak by kotúče premiestňovali „najrozumnejším" spôsobom a na premiestnenie jedného kotúča by vynaložili hodinu práce).

Naša hra s počítačom je trocha jednoduchšia. Na jej začiatku máme pred sebou tri veže, pričom ľavá má (podľa našej voľby) tri až sedem kotúčov (najväčší dole, najmenší hore). Cieľom hry je čo najrýchlejšie vyriešiť „úlohu hanojských mníchov", teda premiestniť kotúčový kužeľ z ľavej veže na pravú. Pri prenose kotúčov využívame prostrednú vežu na prechodné odkladanie nevhodných kotúčov, aby sme vždy zachovali pravidlo „menší kotúč na väčší".

PROGRAM 2.16

Program na riadenie hry HANOJSKÉ VEŽE uvádzame vo výpise 2.16. Výpis je z počítača PMD 85, program sme však zostavili tak, aby po nepatrných úpravách fungoval na väčšine osobných počítačov, preto nevyužívame tie prvky jazyka basic G, ktoré sú v rôznych dialektoch odlišné. Pre Atari stačí program upraviť takto:

Riadok 1560 zmeníme na

```
156Ø POSITION Z-INT(T(K,J)/2),K
```

a riadok 1610 na

```
161Ø POSITION Z,K : PRINT "!"
```

Riadok 1510 je výhodné zmeniť na

```
151Ø GRAPHICS Ø
```

V riadkoch, v ktorých sa vyskytuje príkaz GCLEAR, nahradíme

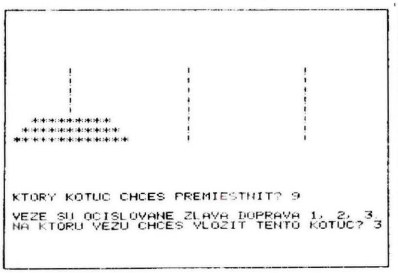
tento príkaz príkazom PRINT CHR$(125); (príkaz vymazania obrazovky).

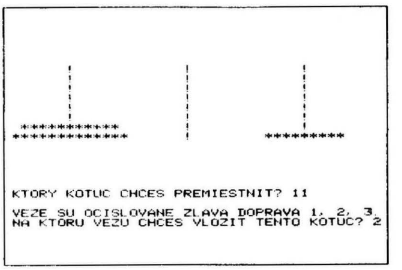
```
10 REM PROGRAM 2.16 - HANOJSKE VEZE, 30. 4. 88
20 REM BASIC G (PMD 85, DIDAKTIK ALFA)
30 DIM T(7,3)
40 GOSUB 2000
100 E=0
110 FOR I=0 TO 7
120 FOR J=1 TO 3
130 T(I,J)=0
140 NEXT J
150 NEXT I
180 GCLEAR
190 PRINT "KOLKO KOTUCOV CHCES PREMIESTNIT?"
200 PRINT "(OD 3 DO 7): "; : INPUT S
210 M=0
220 FOR Q=3 TO 7
230 IF Q=S THEN 350
240 NEXT Q
250 E=E+1
280 IF E>2 THEN 310
290 PRINT
300 PRINT "ZLE, SKUS ESTE RAZ!" : GOTO 190
310 PRINT "PREPAC, ALE NEBUDEM S TEBOU STRACAT CAS."
320 PRINT "KONCIME!!!"
330 STOP
350 PRINT
360 PRINT "V PROGRAME OZNACUJEME KOTUCE TAKTO:"
365 PRINT
370 PRINT "      13              NAJVACSI"
380 PRINT "      11              DRUHY V PORADI"
390 PRINT "       9              TRETI V PORADI"
400 PRINT "       ."
410 PRINT "       ."
420 PRINT "       ."
430 PRINT "       1              NAJMENSI"
435 PRINT
440 PRINT "AK CHCES NAPRIKLAD HRAT S 3 KOTUCMI,"
450 PRINT "BUDU OZNACENE 13, 11 A 9."
460 FOR I=1 TO 1000 : NEXT I : GCLEAR
470 GOSUB 1400
480 PRINT "   KTORY KOTUC CHCES PREMIESTNIT? "; : E=0
490 INPUT D
500 FOR J=1 TO S : J1=13-(J-1)*2
510 IF (J1-D)=0 THEN 610
520 NEXT J
530 PRINT "ZLE, MOZES VOLIT IBA ";
540 FOR J=1 TO S : J1=13-(J-1)*2
550 PRINT J1; : NEXT J : PRINT
560 E=E+1 : IF E>2 THEN 580
570 PRINT "SKUS ESTE RAZ: "; : GOTO 490
580 PRINT "PREPAC, ALE NEBUDEM S TEBOU STRACAT CAS."
590 PRINT "KONCIME!!!"
600 REM KONTROLA PORADIA KOTUCOV
610 FOR R=1 TO 7
620 FOR C=1 TO 3
630 IF T(R,C)=D THEN 650
640 NEXT C : NEXT R
650 FOR Q=R TO 1 STEP -1
660 IF T(Q,C)=0 THEN 680
670 IF T(Q,C)<D THEN 700
680 NEXT Q
690 GOTO 750
700 PRINT "ZLE, TENTO KOTUC LEZI POD INYM KOTUCOM!"
710 PRINT "ZVOL INY KOTUC!" : GOTO 480
750 E=0
760 PRINT
770 PRINT     VEZE SU OCISLOVANE ZLAVA DOPRAVA 1, 2, 3."
```

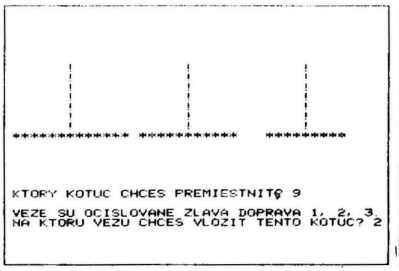
```
780 PRINT "   NA KTORU VEZU CHCES VLOZIT TENTO KOTUC? ";
785 INPUT N
790 IF (N-1)*(N-2)*(N-3)=0 THEN 900

800 E=E+1 : IF E>2 THEN 830
810 PRINT "ZLE, SKUS ESTE RAZ: "; : GOTO 785
820 STOP
830 PRINT
840 PRINT "PREPAC, ALE NEBUDEM S TEBOU STRACAT CAS."
850 PRINT "KONCIME!!!"
860 STOP
900 FOR R=1 TO 7
910 IF T(R,N)<>0 THEN 950
920 NEXT R
930 GOTO 1000
940 REM TEST POLOHY KOTUCA
950 IF D<T(R,N) THEN 1000
960 PRINT
970 PRINT "ZLE, NESMIES POLOZIT VACSI KOTUC NA MENSI!"
980 PRINT "SKUS ESTE RAZ. TEDA, "; : GOTO 480
990 REM PRESUN KOTUCA
1000 FOR V=1 TO 7 : FOR W=1 TO 3
1010 IF T(V,W)=D THEN 1030
1020 NEXT W : NEXT V
1030 FOR U=1 TO 7
1040 IF T(U,N)<>0 THEN 1070
1050 NEXT U
1060 U=7 : GOTO 1080
1070 U=U-1
1080 T(U,N)=T(V,W) : T(V,W)=0
1090 GOSUB 1500
1100 REM TEST UKONCENIA
1110 M=M+1
1120 FOR R=1 TO 7 : FOR C=1 TO 2
1130 IF T(R,C)<>0 THEN 1160
1140 NEXT C : NEXT R
1150 GOTO 1220
1160 Z0=INT(2^S)+S
1170 IF M<=Z0 THEN 480
1180 PRINT
1190 PRINT "LUTUJEM, ALE ";Z0;" POKUSOV BA MALO STACIT."
1200 PRINT "KONCIME!!!"
1210 GOTO 2500
1220 PRINT
1230 PRINT "   BLAHOZELAM!!!"
1235 PRINT "   VYRIESIL SI ULOHU ZA";M;"POKUSOV"
1240 PRINT
1250 PRINT "   SKUSIS ESTE RAZ? (ANO=1, NIE=0): ";
1260 INPUT A0
1270 IF A0=0 THEN 2500
1280 IF A0=1 THEN 100
1290 GOTO 1240
1400 REM ZOBRAZENIE
1410 Y=7 : D=13
1420 FOR X=S TO 1 STEP -1
1430 T(Y,1)=D : D=D-2 : Y=Y-1
1440 NEXT X
1450 GOSUB 1500
1460 RETURN
1500 REM PODPROGRAM ZOBRAZENIA VEZI
1505 GCLEAR
1510 PRINT
1520 FOR K=0 TO 7
1530 Z=6
1540 FOR J=1 TO 3
1550 IF T(K,J)=0 THEN 1610
1560 PRINT TAB(Z+3-INT(T(K,J)/2));
1570 FOR V=1 TO T(K,J)
1580 PRINT "*";
1590 NEXT V
1600 GOTO 1615
```

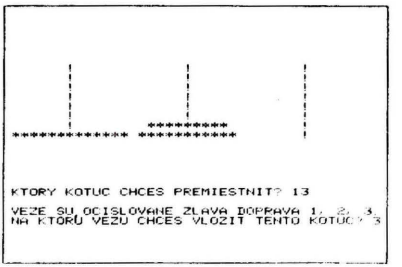
```
1610 PRINT TAB(Z+3);"!";
1615 Z=Z+13
1620 NEXT J
1630 PRINT
1640 NEXT K
1650 PRINT : PRINT : PRINT : PRINT : PRINT
1660 RETURN
2000 REM UVODNA INFORMACIA
2010 GOLEAR
2020 PRINT "ZAHRAJ SI SO MNOU HRU 'HANOJSKE VEZE'"
2030 PRINT : PRINT : PRINT : PRINT : PRINT
2040 PRINT "NA OBRAZOVKE SA ZOBRAZIA TRI VEZE: "
2045 S=7 : GOSUB 1400
2050 PRINT "Z LAVEJ VEZE TREBA PREMIESTNIT VSETKY"
2060 PRINT "KOTUCE NA PRAVU VEZU, PRICOM NESMIES"
2070 PRINT "POLOZIT VACSI KOTUC NA MENSI. NA PRE-"
2080 PRINT "CHODNE ODKLADANIE KOTUCOV POUZIVAJ PRO_"
2090 PRINT "STREDNU VEZU. MOZES SI ZVOLIT POCET"
2100 PRINT "KOTUCOV (OD 3 DO 7)."
2110 PRINT
2120 PRINT "VELA USPECHOV!!!"
2130 PRINT
2140 PRINT "SI PRIPRAVENY? (ANO=1, NIE=0): ";
2150 INPUT Q
2160 IF Q=1 THEN RETURN
2170 GOTO 2130
2500 REM UKONCENIE PROGRAMU
2510 PRINT
2520 PRINT "DO VIDENIA!"
2530 PRINT
2540 END
```

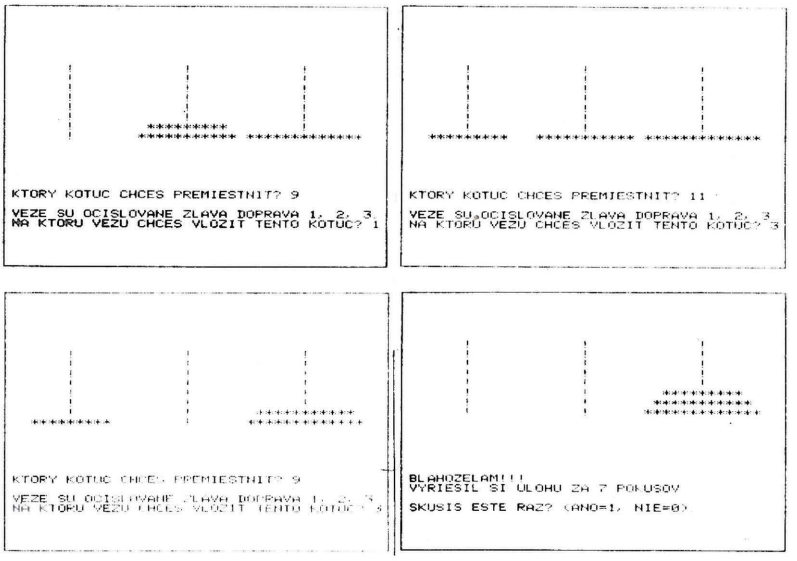
Postup hry pre 3 kotúče uvádzame na obrázkoch 2.22 až 2.29. Keď ho pozorne preskúmate, môžete odvodiť správny postup aj pre väčší počet kotúčov. Možno vám pomôže informácia, že na premiestnenie n kotúčov potrebujeme najmenej 2^n-1 pokusov.











2.22–2.29 Dialóg s programom HANOJSKÉ VEŽE

Spôsob, ktorým v programe zadávate pokyny na prenos kotúčov z jednej veže na druhú, je vysvetlený v úvodnej informácii, ktorá sa zobrazí po odštartovaní programu, a v ďalších informáciách, poskytovaných počítačom v priebehu hry.

2.2.3 HRA ZÁPALKY

Hra ZÁPALKY patrí do veľkej skupiny hier nazývaných hry NIM. Podstatou týchto hier je odoberanie predmetov z jednej, prípadne z viacerých skupín predmetov podľa vopred stanovených pravidiel. Tie určujú, koľko predmetov možno naraz odobrať a za akých podmienok hráč vyhráva.

Hra ZÁPALKY sa začína tak, že určíme počet zápaliek, s ktorými chceme hrať. Z kôpky zápaliek dvaja hráči postupne odoberajú podľa vlastného uváženia jednu, dve alebo tri zápalky. Prehráva ten z hráčov, ktorý odoberie poslednú zápalku.

PROGRAM 2.17

Výpis 2.17 obsahuje program, ktorý s vami hrá na počítačoch PMD 85 a Didaktik Alfa hru ZÁPALKY. Úlohu jedného z hráčov

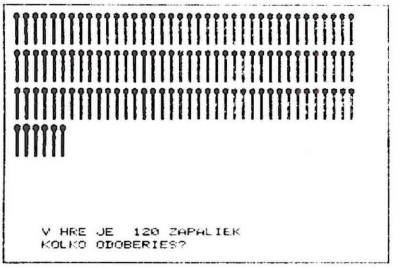
preberá počítač. Po odštartovaní vás oboznámi s pravidlami hry a vyzve vás, aby ste zadali počet zápaliek, s ktorými sa bude hrať (najmenej 13 a najviac 250). Ďalší priebeh hry sa uskutočňuje v dialógu s počítačom, všetky potrebné informácie nájdete na obrazovke.

```
10 REM PROGRAM 2.17 - HRA ZAPALKY, 5. 5. 88
20 REM BASIC G (PMD 85, DIDAKTIK ALFA)
40 GOSUB 1300
50 GOSUB 1000
100 GCLEAR
110 DISP "ZADAJ, S KOLKYMI ZAPALKAMI CHCES HRAT:"
120 INPUT A
130 IF A)=13 AND A (=250 THEN GOTO 200
140 DISP "ZADAJ ZNOVA (NAJMENEJ 13, NAJVIAC 250):"
150 GOTO 120
200 IF A(=42 THEN R1=1
210 IF A)42 AND A(=82  THEN R1=2
220 IF A)82 AND A(=126 THEN R1=3
230 IF A)126 AND A(=168 THEN R1=4
240 IF A)168 AND A(=210 THEN R1=5
250 IF A)210 AND A(=252 THEN R1=6
260 GOSUB 1500
400 E0=0
401 GOSUB 1600
403 MOVE 10,10
405 LABEL 1,1;T$
410 DISP "KOLKO ODOBERIES?
420 INPUT B
422 MOVE 10,10
424 IF E0=0 THEN LABEL 1,1;T$
426 IF E0=1 THEN LABEL 1,1;B0$"- ZLE, MOZES BRAT IBA 1, 2, ALEBO 3!"
430 IF B)=1 AND B(=3 THEN GOTO 500
440 E0=1 : B0=B
445 MOVE 10,10
450 LABEL 1,1;B$"- ZLE, MOZES BRAT IBA 1, 2, ALEBO 3!"
460 GOTO 410
500 GOSUB 1500
510 A=A-B
520 IF A)=1 THEN GOSUB 1500
530 MOVE 10,10
540 LABEL 1,1;"TVOJ TAH: ZOBRAL SI";B$";" ZOSTALO";A
545 DISP "STLAC KLAVES EOL!
550 INPUT Q0$
555 MOVE 10,10
558 LABEL 1,1;"TVOJ TAH: ZOBRAL SI";B$";" ZOSTALO";A
560 IF A=1 THEN GOTO 800
565 IF A(=0 THEN GOTO 900
570 Z=A-1
580 B=INT(Z)-INT(Z/4)*INT(4)
590 IF B=0 THEN B=INT(3*RND(1)+1)
670 GOSUB 1500
680 A=A-B
690 IF A)=1 THEN GOSUB 1500
700 MOVE 10,10
710 LABEL 1,1;"MOJ TAH: ZOBRAL SOM";B$";" ZOSTALO";A
720 DISP "STLAC KLAVES EOL!
730 INPUT Q0$
740 MOVE 10,10
750 LABEL 1,1;"MOJ TAH: ZOBRAL SOM";B$";" ZOSTALO";A
760 IF A=1 THEN GOTO 900
770 GOTO 400
800 MOVE 5,10
810 LABEL 2,2;"GRATULUJEM, VYHRAL SI!"
820 GOSUB 1700
900 MOVE 5,10
910 LABEL 2,2;"LUTUJEM, PREHRAL SI!"
```

```
920 GOSUB 1700
1000 REM HLAVICKA PROGRAMU
1010 GCLEAR
1020 SCALE 0,200,0,200
1030 MOVE 70,180
1040 LABEL 2,2;"H R A"
1090 MOVE 15,100
1100 LABEL 3,9;"Z A P A L K Y"

1110 MOVE 15,80
1120 LABEL 1,1;"HRAC A POCITAC POSTUPNE ODOBERAJU 1, 2"
1130 MOVE 15,70
1140 LABEL 1,1;"ALEBO 3 ZAPALKY. PREHRA TEN, KTO ZOBE-"
1150 MOVE 15,60
1160 LABEL 1,1;"RIE POSLEDNU ZAPALKU. CELKOVY POCET ZA-"
1170 MOVE 15,50
1180 LABEL 1,1;"PALIEK (NAJMENEJ 13, NAJVIAC 250) ZADA-"
1190 MOVE 15,40
1200 LABEL 1,1;"VA NA ZACIATKU HRY HRAC A AKO PRVY ODO-"
1210 MOVE 15,30
1220 LABEL 1,1;"BERA ZVOLENY POCET ZAPALIEK. "
1230 MOVE 15,20
1240 LABEL 1,1;"     MOZES ZACAT, VELA USPECHOV!"
1250 DISP "                          STLAC KLAVES EOL!"
1270 INPUT Q0$
1280 RETURN
1300 REM PRIPRAVA ZOBRAZENIA ZAPALIEK
1310 FOR I=1 TO 30
1320 READ B
1330 U$=U$+CHR$(B)
1340 NEXT I
1350 RETURN
1500 REM ZOBRAZENIE ZAPALIEK
1505 M1=0
1510 FOR U1=0 TO R1
1520 FOR Z=3 TO 44
1530 BMOVE Z,U1*35+8
1540 BPLOT U$,1
1550 M1=M1+1
1560 IF M1=A THEN GOTO 1590
1570 NEXT Z
1580 NEXT U1
1590 RETURN
1600 REM SKLONOVANIE
1610 Z$="V HRE "
1620 A$=STR$(A)
1630 IF A=1 THEN T$=Z$+"JE 1 ZAPALKA"
1640 IF A>1 AND A<5 THEN T$=Z$+"SU "+A$+" ZAPALKY"
1650 IF A>4 THEN T$=Z$+"JE "+A$+" ZAPALIEK"
1660 RETURN
1700 REM ROZHODOVANIE O UKONCENI
1710 DISP "CHCES HRAT ZNOVU? (A, ALEBO N):"
1720 INPUT Q0$
1730 IF Q0$="A" THEN GOTO 100
1740 GCLEAR
1750 MOVE 20,180
1760 LABEL 1,1;"KONIEC PROGRAMU"
1770 MOVE 15,100
1780 LABEL 3,9;"Z A P A L K Y"
1790 MOVE 50,30
1800 LABEL 2,2;"DO VIDENIA"
1810 END
1900 DATA 12,30,30,30,30,12,12,12,12,12,12,12,12,12,12,12,12,12
1910 DATA 12,12,12,12,12,12,12,12,12,12,12,12,12,12
```

Všimnite si, že v programe sme na kreslenie zápaliek využili bajtový grafický režim (príkazy BMOVE a BPLOT). Údaje v riadkoch 1900 a 1910 v príkazoch DATA definujú motív zápalky. Na obr.

2.30 uvádzame zobrazenie zvoleného počtu zápaliek a ukážku dialógu s počítačom.



2.30 Dialóg s programom ZÁPALKY

Hry NIM patria medzi obľúbené problémy rekreačnej matematiky. Z matematického hľadiska je zaujímavé nájsť v rôznych hrách tohto typu najlepší spôsob hry, ktorý vedie k výhre, tzv. optimálnu stratégiu. Pokúste sa o to najprv sami. Vzhľadom na to, že si môžete zvoliť celkový počet zápaliek a máte prvý ťah pri odoberaní, môžete pri správnom spôsobe hry nad počítačom vždy zvíťaziť.

Tým, ktorí nevedia, ako na to, poradíme. Určenie optimálnej stratégie v našej verzii hry NIM spočíva v tom, že treba nájsť pravidlo voľby celkového počtu zápaliek a pravidlo odoberania zápaliek, ktoré nám zabezpečí výhru proti počítaču. Optimálna stratégia je táto:

1. Celkový počet zápaliek N na začiatku hry treba zvoliť tak, aby
$$N = 4k + p$$
kde k je prirodzené číslo a p sa rovná 2, 3 alebo 4.

2. Predpokladajme, že sme na ťahu a v hre zostalo Z zápaliek. Treba odobrať x zápaliek (kde x sa rovná 1, 2 alebo 3) tak, aby číslo
$$Z - x - 1$$
bolo bez zvyšku deliteľné číslom 4.

Pokúste sa samostatne zdôvodniť, prečo takáto stratégia musí vždy viesť k výhre. Môžete si zabezpečiť výhru v hre, v ktorej zvolíte celkový počet zápaliek 21?

2.2.4 HRA ŽIVOT

Americký vedec John Conway je autorom významných teoretických prác z čistej matematiky. Vo voľnom čase sa však zaoberal aj

„rekreačnou matematikou". Jedným zo zaujímavých darčekov, ktorými potešil priaznivcov matematických hier, je hra ŽIVOT.

Pôvodne táto hra vznikla ako jednoduchý matematický model biologických procesov. Predstavme si, že v určitom priestore existuje kolónia živých organizmov, ktorá sa vyvíja: organizmy sa môžu rozmnožovať, prežívať, zanikať. Pre jednoduchosť si tento priestor môžeme znázorniť štvorcovou tabuľkou, v ktorej sú určité políčka obsadené (je v nich živý organizmus, nabr. bunka) a iné prázdne. Vývoj kolónie sa riadi týmito pravidlami prežitia, zániku a zrodu:

1. V neobsadenom políčku sa v budúcom pokolení zrodí nová bunka práve vtedy, ak sú v danom pokolení presne tri z ôsmich susedných políčok obsadené (teda rodiaca sa bunka musí mať troch živých susedov).
2. Stará bunka v nasledujúcom pokolení zanikne, ak má menej ako dvoch, alebo viac ako troch živých susedov.
3. Stará bunka, ktorá má v danom pokolení dvoch alebo troch živých susedov, prežije aj v nasledujúcom pokolení.

Ak stanovíme nejaké východiskové rozloženie buniek (obsadených a neobsadených políčok v tabuľke), vzniká podľa týchto pravidiel zaujímavý proces vývoja jednotlivých pokolení. Pri niektorých východiskových rozloženiach kolónia buniek postupne odumiera, pri iných sa stabilizuje, pri ďalších „pulzuje", teda po určitom počte pokolení sa znovu opakuje rozloženie, ktoré sa už vyskytlo v minulosti.

Hru ŽIVOT možno hrať aj ručne, na štvorčekovom papieri môžeme podľa pravidiel prežitia, zániku a zrodu postupne kresliť vznikajúce rozloženia buniek a hľadať také východiskové rozloženia, ktoré vedú k zaujímavému vývoju celej kolónie. Je to však veľmi prácne, oveľa výhodnejšie je použiť na tieto účely počítač.

PROGRAM 2.18

Program hry ŽIVOT (pozri výpis 2.18) predpokladá, že bunky sú rozložené v štvorcovej tabuľke 10×10. Východiskové rozloženie kolónie buniek si zvolíte sami a celý ďalší proces vývoja skúma počítač, ktorý na obrazovke postupne zobrazuje rozloženia jednotlivých pokolení (obsadené políčka sa znázorňujú hviezdičkou). Vznikajú pritom často pôvabné obrázky, ponášajúce sa na čipky, snehové vločky, rôzne geometrické obrazce a pohybujúce sa predmety.

Program hry zobrazuje všetky potrebné informácie o spôsobe hry.

František Turnovec

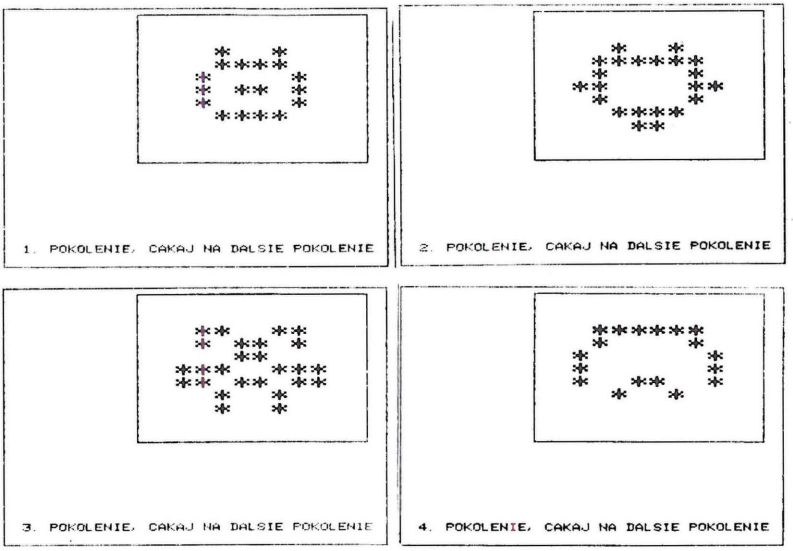
```
10 REM PROGRAM 2.18 - HRA  ZIVOT, 8. 5. 88
20 REM BASIC G (PMD 85, DIDAKTIK ALFA)
50 DIM A(11,11),B(11,11)
90 N=10
100 GOSUB 1000
110 RESTORE
120 GOSUB 1800
130 IF Q0$="A" THEN GOSUB 1600
200 GOSUB 1800
210 GCLEAR
220 MOVE 10,190
230 LABEL 1,1;"ROZLOZENIE PRVEHO POKOLENIA BUNIEK ZA-"
240 MOVE 10,180
250 LABEL 1,1;"DAVAS POLOHOU OBSADENYCH POLICOK (CISLOM"
260 MOVE 10,170
270 LABEL 1,1;"RIADKU A STLPCA). ZADAVANIE OBSADENYCH"
280 MOVE 10,160
290 LABEL 1,1;"POLICOK UKONCIS VLOZENIM NULY AKO"
300 MOVE 10,150
310 LABEL 1,1;"CISLA RIADKU."
320 DISP "VLOZ CISLO RIADKU (1 AZ 10):"
330 INPUT R
335 IF R=0 THEN GOTO 400
340 IF R>10 OR R<1 THEN GOTO 320
350 DISP "VLOZ CISLO STLPCA (1 AZ 10):"
360 INPUT S
370 IF S<1 OR S>10 THEN GOTO 350
375 A(S,R)=1
380 GOSUB 600
390 GOTO 320
400 L=0
410 L=L+1
420 L$=STR$(L)
430 GOSUB 600
440 DISP L$;". POKOLENIE. CHCES POKRACOVAT (A, ALEBO N)?"
450 INPUT Q0$
460 IF Q0$="N" THEN GOTO 1900
465 DISP L$;". POKOLENIE. CAKAJ NA NOVE POKOLENIE!"
470 GOSUB 700
480 GOTO 410
600 REM ZOBRAZENIE TABULKY
605 GCLEAR
610 MOVE 70,200
611 PLOT 190,200
612 PLOT 190,80
613 PLOT 70,80
614 PLOT 70,200
620 FOR I=1 TO N
630 FOR J=1 TO N
635 IF A(I,J)=0 THEN GOTO 670
640 MOVE 70+I*10,190-J*10
660 LABEL 2,2;"*"
670 NEXT J
680 NEXT I
690 RETURN
700 REM NOVE POKOLENIE
710 FOR I=1 TO N
720 FOR J=1 TO N
730 B(I,J)=A(I,J)
740 A(I,J)=0
750 NEXT J
760 NEXT I
770 FOR I=1 TO N
780 FOR J=1 TO N
790 K=B(I-1,J)
800 K=K+B(I+1,J)
810 K=K+B(I,J-1)
820 K=K+B(I,J+1)
830 K=K+B(I-1,J-1)
840 K=K+B(I-1,J+1)
850 K=K+B(I+1,J-1)
```

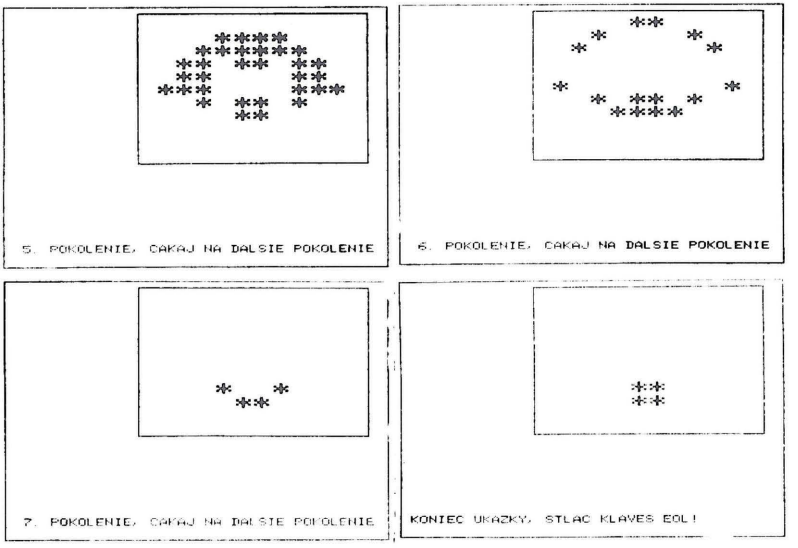
```
860 K=K+B(I+1,J+1)
865 IF K<=1 THEN GOTO 910
866 IF K>=4 THEN GOTO 910
870 IF B(I,J)=0 THEN GOTO 900
880 A(I,J)=1
890 GOTO 910
900 IF K=3 THEN A(I,J)=1
910 NEXT J
920 NEXT I
930 RETURN
1000 REM HLAVICKA PROGRAMU
1010 GCLEAR
1020 SCALE 0,200,0,200
1030 MOVE 70,180
1040 LABEL 2,2;"H R A"
1090 MOVE 30,100
1100 LABEL 3,9;"Z I V O T"
1110 MOVE 10,80
1120 LABEL 1,1;"HRA ZOBRAZUJE VYVOJ POKOLENI SKUPINY"
1130 MOVE 10,70
1140 LABEL 1,1;"BUNIEK PODLA TYCHTO PRAVIDIEL PREZITIA,"
1150 MOVE 10,60
1160 LABEL 1,1;"ZANIKU A ZRODU:"
1165 MOVE 10,50
1167 LABEL 1,1;"1. NOVA BUNKA VZNIKNE VTEDY, AK MA V"
1168 MOVE 10,40
1170 LABEL 1,1;"   DANOM POKOLENI PRESNE 3 SUSEDOV"
1180 MOVE 10,30
1190 LABEL 1,1;"2. STARA BUNKA ZANIKNE, AK MA MENEJ"
1200 MOVE 10,20
1210 LABEL 1,1;"   AKO 2, ALEBO VIAC AKO 3 SUSEDOV,"
1220 MOVE 10,10
1230 LABEL 1,1;"   V OPACNOM PRIPADE PREZIJE"
1250 DISP "                        STLAC KLAVES EOL!"
1260 INPUT Q0$
1270 GCLEAR
1280 MOVE 10,190
1290 LABEL 1,1;"ROZLOZENIE SKUPINY BUNIEK JE ZNAZOR-"
1300 MOVE 10,180
1310 LABEL 1,1;"NENE STVORCOVOU TABULKOU 10 X 10; AK"
1320 MOVE 10,170
1330 LABEL 1,1;"JE V POLICKU TEJTO TABULKY *, NACHADZA"
1340 MOVE 10,160
1350 LABEL 1,1;"SA V NOM BUNKA, POLICKO BEZ BUNKY "
1360 MOVE 10,150
1370 LABEL 1,1;"JE PRAZDNE. ROZLOZENIE BUNIEK PRVEHO"
1380 MOVE 10,140
1390 LABEL 1,1;"POKOLENIA SI MOZETE ZADAT SAMI VLOZENIM"
1400 MOVE 10,130
1410 LABEL 1,1;"CISEL RIADKOV A STLPCOV S OBSADENYMI"
1420 MOVE 10,120
1430 LABEL 1,1;"POLICKAMI A POTOM SLEDOVAT VYVOJ DALSICH"
1440 MOVE 10,110
1445 LABEL 1,1;"POKOLENI."
1450 DISP "CHCES UKAZKU (A, ALEBO N)?"
1460 INPUT Q0$
1500 RETURN
1600 REM UKAZKA
1610 FOR J=3 TO 8
1620 FOR I=3 TO 8
1630 READ A(I,J)
1640 NEXT I
1650 NEXT J
1660 FOR L=1 TO 8
1670 GOSUB 600
1680 IF L=8 THEN 1720
1690 L$=STR$(L)
1700 DISP L$;". POKOLENIE, CAKAJ NA DALSIE POKOLENIE"
1710 GOSUB 700
1720 NEXT L
1730 DISP "KONIEC UKAZKY, STLAC KLAVES EOL!
```

```
1740 INPUT Q0$
1750 RETURN
1800 REM VYNULOVANIE TABULKY
1810 FOR I=1 TO N
1820 FOR J=1 TO N
1830 A(I,J)=0
1840 NEXT J
1850 NEXT I
1860 RETURN
1900 REM PODPROGRAM KONCA HRY
1910 DISP "CHCES HRAT ZNOVA (A, ALEBO N)?
1920 INPUT Q0$
1930 IF Q0$="A" THEN GOTO 100
1940 GCLEAR
1950 MOVE 20,180
1960 LABEL 1,1;"KONIEC PROGRAMU"
1970 MOVE 30,100
1980 LABEL 3,9;"Z I V O T"
1990 MOVE 50,30
2000 LABEL 2,2;"DO VIDENIA"
2010 END
2100 DATA 0,1,0,0,1,0,0,1,1,1,0,1,0,0,0,0
2110 DATA 1,1,0,1,1,0,1,1,0,0,0,0,1,0,1,1,1,1,0
```

Najprv vám ponúkne ukážku vývoja podľa východiskového rozloženia, ktoré vedie k zaujímavej postupnosti pokolení. Potom vás vyzve, aby ste sami (vložením čísel riadkov a stĺpcov, v ktorých sa majú nachádzať obsadené políčka) určili rozloženie východiskového pokolenia, a zobrazí postupne z neho vznikajúce pokolenia.

Na obr. 2.31 uvádzame východiskové rozloženie programom ponúkanej ukážky vývoja. Ukážka sa nazýva „zmiznutie mačky“: mač-





2.31–2.38 Ukážka vývoja v hre ŽIVOT

ka sa postupne mení a stráca (pozri obrázky 2.32 až 2.36), až z nej nakoniec zostáva iba „mačací úsmev" (obr. 2.37) a „stopa mačacej labky" (obr. 2.38), ktorá sa už ďalej nemení.

Program je určený pre PMD 85 a Didaktik Alfa. Keďže neobsahuje žiadne zvláštnosti, možno ho bez problémov prepísať do Atari basicu, napr. v grafickom režime 1. Namiesto príkazov MOVE a LABEL používajte v Atari basicu príkazy POSITION a PRINT #6, ktorými umiestnite text do grafického poľa obrazovky, namiesto príkazu DISP môžete použiť PRINT, ktorý v grafickom režime 1 vypisuje do textového okienka.

2.2.5 HRA ČÍSLA

Hra má jednoduché pravidlá: počítač si „myslí" päťciferné číslo (vytvorené číslicami 0 až 9), pričom jednotlivé číslice sa môžu viackrát opakovať a prvá z nich je rôzna od nuly. Hráč má toto číslo uhádnuť a má na to 10 pokusov.

PROGRAM 2.19

Výpis 2.19 obsahuje program hry ČÍSLA pre počítače PMD 85 a Didaktik Alfa. Po odštartovaní vás počítač vyzve, aby ste vložili

svoje meno a vek (počet rokov), a pripíše vám 10 bodov. Pri každom pokuse vložíte na výzvu počítača päťciferné číslo. Za neúspešný pokus (vaše číslo sa nezhoduje s číslom, ktoré si „zvolil" počítač) ste potrestaný stratou bodu. Po neúspešnom pokuse vám počítač zároveň oznámi, ktoré z číslic vami vloženého čísla svojou polohou zodpovedajú číslicam ním zvoleného čísla. Hra sa pre vás skončí výhrou po prvom úspešnom pokuse (ak ste ešte nevyčerpali 10 pokusov), alebo prehrou po 10 neúspešných pokusoch. Ak budete rozmýšľať, nemôžete prehrať.

```
10 REM PROGRAM 2.19 - HRA CISLA, 11. 5. 88
20 REM BASIC G (PMD 85, DIDAKTIK ALFA)
100 GOSUB 1000
110 L=L+1
200 GOSUB 1800
310 S=10
350 FOR P=1 TO 10
360 GOSUB 700
370 GOSUB 1400
380 IF M$=T$ THEN GOTO 450
390 S=S-1
400 GOSUB 1500
410 NEXT P
420 MOVE 5,20
430 LABEL 2,2;"LUTUJEM, PREHRAL SI!"
440 GOTO 500
450 GOSUB 1500
460 MOVE 5,20
470 LABEL 2,2;"GRATULUJEM, VYHRAL SI!"
500 DISP "STLAC KLAVES EOL!
510 INPUT Q0$
520 GOSUB 2200
530 GOSUB 2000
600 DISP "CHCES SKONCIT? (A, ALEBO N):
610 INPUT Q0$
620 IF Q0$="N" THEN GOTO 100
630 GCLEAR
640 MOVE 20,180
650 LABEL 1,1;"KONIEC PROGRAMU"
660 MOVE 30,100
670 LABEL 3,9;"C I S L A"
680 MOVE 50,30
690 LABEL 2,2;"DO VIDENIA"
695 GOTO 3000
700 REM VLOZENIE CISLA
710 IF P>1 THEN GOTO 749
720 MOVE 50,50
730 LABEL 1,1;"MAS K DISPOZICII"
740 MOVE 125,50
742 GOSUB 1700
745 LABEL 2,2;S
747 MOVE 160,50
748 LABEL 1,1;D$
749 P$=STR$(P)
750 IF P=1 THEN P$=P$+"-VY"
751 IF P=2 THEN P$=P$+"-HY"
752 IF P=3 THEN P$=P$+"-TI"
753 IF P>3 THEN P$=P$+"-TY"
760 DISP P$;" POKUS: VLOZ SVOJE CISLO A STLAC EOL!"
770 INPUT T$
780 IF LEN(T$)=5 THEN GOTO 810
790 DISP T$;" - NIE JE PATCIFERNE, VLOZ ZNOVA!"
800 GOTO 770
```

```
810 FOR I=1 TO 5
820 C$=MID$(T$,I,1)
830 IF ASC(C$)<48 OR ASC(C$)>57 THEN 860
840 NEXT I
850 RETURN
860 DISP T$;" - INE ZNAKY AKO CISLICE, VLOZ ZNOVA!"
870 GOTO 770
1000 REM HLAVICKA PROGRAMU
1010 GCLEAR
1020 SCALE 0,200,0,200
1030 MOVE 70,180
1040 LABEL 2,2;"H R A"
1090 MOVE 30,100
1100 LABEL 3,9;"C I S L A"
1110 MOVE 10,80
1120 LABEL 1,1;"MYSLIM SI 5-CIFERNE CISLO. TVOJOU ULO-"
1130 MOVE 10,70
1140 LABEL 1,1;"HOU JE TOTO CISLO UHADNUT. MAS NA TO"
1150 MOVE 10,60
1160 LABEL 1,1;"10 POKUSOV. NA ZACIATKU TI PRIDELIM 10"
1170 MOVE 10,50
1180 LABEL 1,1;"BODOV, ZA KAZDY NEUSPESNY POKUS STRA-"
1190 MOVE 10,40
1200 LABEL 1,1;"CAS BOD. PO KAZDOM POKUSE TI OZNAMIM
1210 MOVE 10,30
1220 LABEL 1,1;"UHADNUTE CISLICE A ICH POLOHU."
1230 MOVE 20,10
1240 LABEL 1,2;" MOZES ZACAT, VELA USPECHOV!"
1250 DISP "                    STLAC KLAVES ECL!"
1260 INPUT Q0$
1300 RETURN
1400 REM POROVNANIE CISEL
1405 Z$=""
1410 FOR I=1 TO 5
1420 A$=MID$(T$,I,1)
1430 B$=MID$(M$,I,1)
1440 IF A$<>B$ THEN Z$=Z$+"?"
1450 IF A$=B$ THEN Z$=Z$+A$
1460 NEXT I
1470 RETURN
1500 REM ZOBRAZENIE CISEL
1510 GCLEAR
1520 MOVE 20,190
1530 LABEL 1,1;"TVOJE CISLO:"
1540 MOVE 40,160
1550 LABEL 3,3;T$
1560 MOVE 20,140
1570 LABEL 1,1;"MOJE CISLO"
1580 MOVE 40,110
1590 LABEL 3,3;Z$
1600 GOSUB 1700
1610 MOVE 50,50
1612 W$="MAS ESTE
1614 IF P=9 THEN W$="MAS UZ LEN"
1616 IF P=10 THEN W$="KONCIS - "
1618 IF T$=M$ THEN W$="ZISKAL SI "
1620 LABEL 1,1;W$
1630 MOVE 100,50
1640 LABEL 2,2;S
1650 MOVE 135,50
1660 LABEL 1,1;D$
1670 RETURN
1700 REM SKLONOVANIE
1710 D$="BODOV"
1720 IF S>1 AND S<5 THEN D$="BODY"
1730 IF S=1 THEN D$="BOD"
1740 RETURN
1800 REM POCITAC VOLI CISLO
1810 GCLEAR
1820 DISP "VLOZ SVOJE MENO:"
```

```
1830 INPUT N$
1840 DISP "KOLKO MAS ROKOV?"
1850 INPUT R$
1860 MOVE 10,170
1870 LABEL 1,2;N$;" (";R$;" ROKOV)"
1880 FOR I=1 TO LEN(N$)+VAL(R$)
1890 R=RND(1)
1900 NEXT I
1910 M(1)=INT(9*RND(1)+1)
1920 M$=MID$(STR$(M(1)),2,1)
1930 FOR I=2 TO 5
1940 M(I)=INT(9*RND(I))
1950 M$=M$+MID$(STR$(M(I)),2,1)
1960 NEXT I
1970 RETURN
2000 REM PREHLAD VYSLEDKOV
2010 GCLEAR
2020 MOVE 10,180
2030 LABEL 1,2;"DOTERAJSIE NAJLEPSIE VYSLEDKY:"
2040 FOR I=1 TO 10
2050 MOVE 20,160-I*10
2060 S=B(I)
2070 GOSUB 1700
2080 LABEL 1,1;STR$(I);". ";N$(I);TAB(30);B(I);D$
2090 IF I=L THEN GOTO 2110
2100 NEXT I
2110 RETURN
2200 REM USPORIADANIE VYSLEDKOV
2210 FOR I=1 TO 10
2220 IF S>= B(I) THEN GOTO 2250
2230 NEXT I
2240 RETURN
2250 I1=I+1
2260 FOR J=10 TO I1 STEP -1
2270 N$(J)=N$(J-1)
2280 B(J)=B(J-1)
2290 NEXT J
2300 N$(I)=N$+" (";R$;" ROKOV)"
2310 B(I)=S
2320 RETURN
3000 END
```

Po skončení hry sa vás počítač opýta, či chcete pokračovať (skúsiť ešte raz, alebo pustiť kamaráta), a v prípade kladnej odpovede sa hra zopakuje ďalším kolom. Samozrejme, počítač si zvolí iné číslo ako v predchádzajúcom kole. Prvok súťaživosti je v hre zvýraznený tým, že počítač si „pamätá" 10 najúspešnejších hráčov v súvislej postupnosti hier a ich poradie doplní a zobrazí po skončení každého kola.

2.2.6 HRA SLOVÁ

Hru SLOVÁ hrajú pomocou počítača dvaja hráči. Jeden vloží zvolené slovo a druhý sa potom snaží toto slovo uhádnuť. Na obrazovke sa znakmi podčiarknutia vyznačí počet písmen neznámeho slova a druhý hráč postupne vkladá písmená, o ktorých si myslí, že sú súčasťou hľadaného slova. Keď sa vložené písmeno vyskytuje v neznámom slove, zobrazí sa na všetkých miestach jeho výskytu. V opačnom prípade sa zobrazená časť slova nezmení.

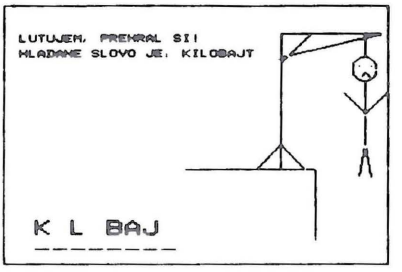
PROGRAM 2.20

Program hry SLOVÁ v basicu G uvádzame vo výpise 2.20. Program pripúšťa pri hádaní zadaného slova iba 8 neúspešných pokusov (neúspešným pokusom je vloženie písmena, ktoré v danom slove nie

```
10 REM PROGRAM 2.20 - HRA SLOVA, 31. 5. 88
20 REM BASIC G (PMD 85, DIDAKTIK ALFA)
50 DIM V$(20)
60 M$="" : FOR I=1 TO 48 : M$=M$+" " : NEXT I
100 GOSUB 1500
110 SCALE 0,255,0,175
120 X=240 : Y=132 : R=8 : D=0
130 GCLEAR : GOSUB 300
150 GOSUB 1000 : GOSUB 1400
180 GOSUB 500 : GOSUB 1300
200 IF Q0$="A" THEN 120
210 END
300 REM PODPROGRAM VLOZENIA SLOVA
305 DISP "DOHOVORTE SA, KTO VKLADA SLOVO A KTO HADA!"
306 PAUSE 20
310 DISP "VLOZ SLOVO, KTORE MA URCIT TVOJ SUPER:
320 INPUT W$ : GCLEAR : B=LEN(W$)
330 FOR N=1 TO B : V$(N)=" " : NEXT N : MOVE 20,10
390 FOR N=0 TO B-1 : LABEL 2,1;"_";
410 NEXT N : DISP "HRA MOZE ZACAT, ZAVOLAJ SUPERA!"
420 RETURN
500 REM PODPROGRAM HLADANIA SLOVA
510 DISP M$  : DISP "HADAJ PISMENO:"
520 INPUT G$ : IF G$="" THEN 510
530 MOVE 20,20
540 FOR I=1 TO B : LABEL 2,2;V$(I);
545 NEXT I
550 G$=LEFT$(G$,1) : U$=""
560 FOR I=1 TO B : U$=U$+V$(I) : NEXT I
570 FOR N=1 TO B
580 IF MID$(W$,N,1)=G$ THEN V$(N)=G$
590 NEXT N : MOVE 20,20
610 FOR N=1 TO B
620 LABEL 2,2;V$(N);
630 NEXT N
640 Z$="" : FOR I=1 TO B : Z$=Z$+V$(I) : NEXT I
650 IF Z$=W$ THEN 750
660 IF Z$<>U$ THEN 510
665 IF D=8 THEN 850
670 D=D+1 : READ R0,S0,R1,S1
690 MOVE R0,S0
700 PLOT R1,S1
710 GOTO 510
750 X=240 : GOSUB 1000 : GOSUB 1400 : X=146 : GOSUB 1000
790 MOVE 143,129
800 PLOT 146,126
810 PLOT 149,129
820 DISP "GRATULUJEM, VYHRAL SI!" : PAUSE 20
830 RETURN
850 REM NEUSPESNY VYSLEDOK
860 MOVE 255,65
870 PLOT 207,65
880 PLOT 207,17
890 GOSUB 1400
910 MOVE 255,117
920 PLOT 240,102
930 PLOT 225,117
935 MOVE 236,81
936 PLOT 240,102
937 PLOT 244,81
940 MOVE 255,66
950 PLOT 240,81
```

```
960 PLOT 225,64
970 MOVE 236,60
975 PLOT 240,81
976 PLOT 244,60
980 MOVE 237,127
981 PLOT 240,130
982 PLOT 243,127
985 DISP "LUTUJEM, PREHRAL SI!" ; PAUSE 10

986 DISP "HLADANE SLOVO JE: ";W$ : PAUSE 20
990 RETURN
1000 REM PODPROGRAM ZOBRAZENIA FIGURKY
1110 PI=3.14159 : MOVE X+R,Y
1140 FOR I=0 TO 2*PI STEP PI/20
1150 PLOT R*COS(I)+X,R*SIN(I)+Y
1160 NEXT I
1170 PLOT X+4,134,1
1180 PLOT X-4,134,1
1190 MOVE X,123
1200 PLOT X,103
1210 MOVE X,101
1220 PLOT X,82
1230 MOVE X-15,66
1240 PLOT X,81
1250 PLOT X+15,66
1260 MOVE X-15,117
1270 PLOT X,102
1280 PLOT X+15,117
1290 RETURN
1300 REM PODPROGRAM UKONCENIA
1310 DISP M$ : DISP "SKUSIS ESTE RAZ? (A, N): "
1320 INPUT Q0$ : IF Q0$="A" THEN 1350
1330 IF Q0$="N" THEN RETURN
1340 GOTO 1310
1350 RESTORE
1360 RETURN
1400 REM USTA
1410 MOVE 238,128
1420 PLOT 242,128
1430 RETURN
1500 REM HLAVICKA PROGRAMU
1510 SCALE 0,255,0,241
1520 GCLEAR
1530 MOVE 90,220
1540 LABEL 2,2;"H R A"
1590 MOVE 60,140
1600 LABEL 2,6;"S L O V A"
1610 MOVE 5,120
1620 LABEL 1,1;"HRU HRAJU POMOCOU POCITACA DVAJA HRACI."
1630 MOVE 5,110
1640 LABEL 1,1;"PRVY VLOZI DO PAMATI POCITACA SLOVO A"
1650 MOVE 5,100
1660 LABEL 1,1;"POTOM ZAVOLA DRUHEHO. TEN SA SNAZI TOTO"
1670 MOVE 5,90
1680 LABEL 1,1;"SLOVO UHADNUT POSTUPNYM VKLADANIM PISMEN,"
1690 MOVE 5,80
1700 LABEL 1,1;"O KTORYCH PREDPOKLADA, ZE SU SUCASTOU"
1710 MOVE 5,70
1720 LABEL 1,1;"HLADANEHO SLOVA. MOZE SI DOVOLIT IBA"
1730 MOVE 5,60
1740 LABEL 1,1;"8 NEUSPESNYCH POKUSOV."
1750 MOVE 70,20
1760 LABEL 1,2;"VELA USPECHOV!"
1770 DISP "                          STLAC KLAVES EOL!"
1780 INPUT Q0$
1790 RETURN
2000 DATA 120,65,255,65,184,65,184,156
2010 DATA 168,65,184,81,184,81,200,65
2020 DATA 184,156,252,156,184,140,252,156
2030 DATA 204,156,184,136,240,156,240,140
```

je obsiahnuté). Pri každom neúspešnom pokuse sa v pravej časti obrazovky mení obrázok, ktorý je pre neúspešného hráča čoraz nepríjemnejší (pozri obr. 2.39).



2.39 Neúspech v hre SLOVÁ

2.2.7 HRA TROJUHOLNÍKY

V hre TROJUHOLNÍKY dvaja hráči (jedným z nich môže byť počítač) striedavo spájajú dvojice zo šiestich bodov, ktoré sú vrcholmi šesťuholníka. Každé tri vrcholy určujú trojuholník a z vrcholov šesťuholníka možno vytvoriť 20 rôznych trojuholníkov. Prehrá ten z hráčov, ktorý ako prvý nakreslí trojuholník. V hre teda treba voliť spojnice bodov tak, aby nevytvorili trojuholník a aby súper nemal v istom momente inú možnosť, ako zvoliť dvojicu bodov, ktorých spojnica spolu so spojnicami, ktoré už zvolil v predchádzajúcich ťahoch, vytvorí trojuholník.

PROGRAM 2.21

Výpis 2.21 obsahuje program hry TROJUHOLNÍKY pre počítače PMD 85 a Didaktik Alfa. Po odštartovaní programu a zobrazení úvodnej informácie sa na obrazovke zobrazí 6 vrcholov pravidelného šesťuholníka a počítač vás vyzve, aby ste sa rozhodli, či chcete začať. Podľa vašej odpovede buď prenechá prvý ťah vám, alebo ťahá sám. Ak ste sa rozhodli začať hru, vyžiada od vás vloženie dvojice bodov a príslušnú spojnicu vykreslí na obrazovke (pozri obrázky 2.40 a 2.41). Potom počítač oznámi svoj ťah a zobrazí ním zvolenú dvojicu bodov. Vaše spojnice sa vykresľujú plnou čiarou, voľba počítača je vykreslená prerušovanou čiarou. Hra pokračuje až do vytvorenia

František Turnovec

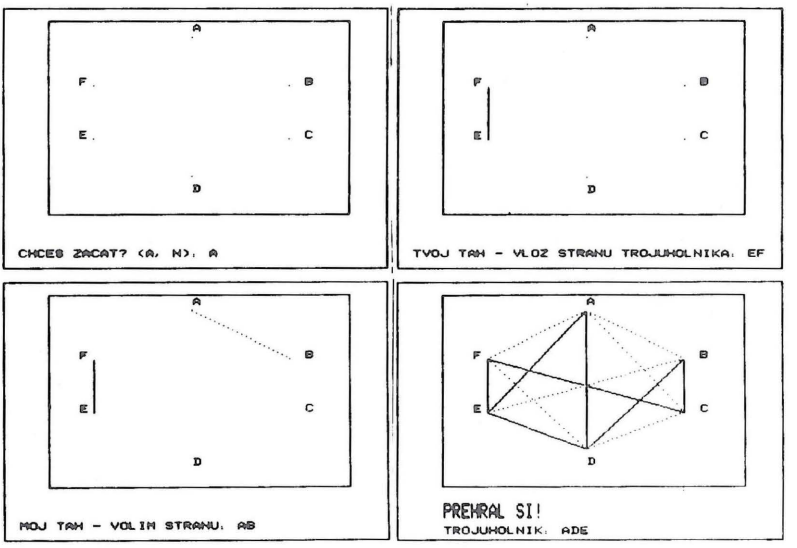
```
10 REM PROGRAM 2.21 - HRA TROJUHOLNIKY, 1. 6. 88
20 REM BASIC G (PMD 85, DIDAKTIK ALFA)
30 DIM G(6,2),C(15),R(15,4),S(15),T(20,3)
40 SCALE 0,255,0,242 : GOSUB 2500
50 RESTORE 3010
60 GCLEAR : GOSUB 140 : GOSUB 600
70 GOSUB 500
80 GOSUB 1200
90 IF Q0$="A" THEN 50
100 DISP "KONIEC HRY  T R O J U H O L N I K Y"
110 MOVE 0,0
120 FILL 255,243;1
130 END
140 REM PRIPRAVA UDAJOV
150 S=15 : T=20 : U=4
160 S$="ABACADAEAFBCBDBEBFCDCECFDEDFEF"
170 T$="ABCABDABEABFACDACEACFADEADFAEF"
180 T$=T$+"BCDBCEBCFBDEBDFBEFCDECDFCEFDEF"
190 FOR I=1 TO S
200 FOR J=1 TO U
210 READ R(I,J)
220 NEXT J : NEXT I
230 FOR I=1 TO S
240 C(I)=1 : S(I)=0
250 NEXT I
260 FOR I=1 TO T
270 FOR J=1 TO 3
280 T(I,J)=0
290 NEXT J : NEXT I
300 FOR I=1 TO 6
310 FOR J=1 TO 2
320 READ G(I,J)
330 NEXT J : NEXT I
340 RETURN
500 REM ZACIATOK HRY
510 DISP "CHCES ZACAT? (A, N): "
520 INPUT Q0$
530 IF Q0$="A" THEN RETURN
540 IF Q0$="N" THEN RETURN
550 GOTO 510
600 REM ZOBRAZENIE SESTUHOLNIKA
610 FOR K=1 TO 6 : X=G(K,1) : Y=G(K,2)
630 MOVE X,Y
640 LABEL 1,1;CHR$(64+K)
650 IF X<125 THEN L1=X+10
660 IF X>125 THEN L1=X-10
670 IF X=125 THEN L1=X
680 IF Y>=219 THEN L2=Y-5
690 IF Y<=71 THEN L2=Y+15
700 IF Y>71 AND Y<219 THEN L2=Y
710 PLOT L1,L2,1
720 G(K,1)=L1 : G(K,2)=L2
730 NEXT K
740 MOVE 30,50
750 PLOT 230,50
760 PLOT 230,230
770 PLOT 30,230
780 PLOT 30,50
790 RETURN
800 REM ZOBRAZENIE TAHU
820 X$=LEFT$(V$,1) : Y$=RIGHT$(V$,1)
830 L0=ASC(X$)-64 : V0=ASC(Y$)-64
840 L1=G(L0,1) : L2=G(L0,2)
850 V1=G(V0,1) : V2=G(V0,2)
855 IF Q=0 THEN 1000
860 MOVE L1,L2
870 PLOT V1,V2
900 RETURN
1000 IF V1-L1=0 THEN 1080
1005 A=(V2-L2)/(V1-L1)
1010 B=L2-A*L1
```

```
1020 MOVE L1,L2
1025 P0=3 : IF L1>V1 THEN P0=-3
1030 FOR Z=L1 TO V1 STEP P0
1040 W=A*Z+B
1050 PLOT Z,W,1
1060 NEXT Z
1070 RETURN
1080 P0=3 : IF L2>V2 THEN P0=-3
1090 FOR W=L2 TO V2 STEP P0
1100 PLOT V1,W,1
1110 NEXT W
1120 RETURN
1200 REM PODPROGRAM HRY
1210 IF Q0$="A" THEN 1400
1220 K=INT(S*RND(1))+1
1230 REM ZACIATOK DVOJTAHU
1240 L=1
1250 S(K)=L
1260 GOSUB 2200
1270 Q=0 : DISP "MOJ TAH - VOLIM STRANU: "+MID$(S$,2*K-1,2)
1275 V$=MID$(S$,2*K-1,2) : GOSUB 800
1280 GOSUB 2300 : IF Z0=1 THEN 1990
1290 FOR I=1 TO S
1300 IF S(I)=0 THEN 1400
1310 NEXT I
1320 GOTO 1970
1400 REM VOLBA HRACA
1410 Q=1 : DISP "TVOJ TAH - VLOZ STRANU TROJUHOLNIKA: "
1420 INPUT V$
1430 IF LEN(V$)<>2 OR LEFT$(V$,1)=RIGHT$(V$,1) THEN 1490
1440 IF LEFT$(V$,1)>RIGHT$(V$,1) THEN V$=RIGHT$(V$,1)+LEFT$(V$,1)
1450 FOR I=2 TO 2*S STEP 2
1470 IF V$=MID$(S$,I-1,2) THEN 1510
1480 NEXT I
1490 DISP "CHYBNE ZADANIE! " : BEEP : PAUSE 10
1500 GOTO 1400
1510 K=I/2
1520 IF S(K)=0 THEN 1550
1530 DISP "STRANA ";V$;" UZ BOLA VYBRANA! " : BEEP : PAUSE 10
1540 GOTO 1400
1550 L=-1
1560 S(K)=L
1580 GOSUB 800
1590 GOSUB 2200 : GOSUB 2300 : IF Z0=1 THEN 1990
1600 REM TAH POCITACA
1610 Y=100 : Z=0
1620 FOR I=1 TO S
1630 C(I)=0
1640 IF S(I)<>0 THEN 1920
1650 FOR J=1 TO U
1660 M=R(I,J)
1670 IF T(M,1)=0 THEN 1850
1680 IF T(M,3)<>0 THEN 1880
1690 X=T(M,1)+T(M,2)+3
1700 ON X GOTO 1810,1830,1850,1810,1870
1810 C(I)=C(I)+4 : GOTO 1880
1830 C(I)=C(I)+2 : GOTO 1880
1850 C(I)=C(I)+1 : GOTO 1880
1870 C(I)=C(I)+20
1880 NEXT J
1890 IF C(I)>=Y THEN 1920
1900 Y=C(I)
1910 Z=I
1920 NEXT I
1930 IF Y=100 THEN 1970
1940 K=Z
1950 GOTO 1240
1960 REM SPRAVA O VYSLEDKU
1970 MOVE 30,20
1975 LABEL 1,2;"NEROZHODNE!"
1980 GOTO 2040
```

```
1990 IF T(I,1)<>1 THEN 2020
2000 MOVE 30,20
2005 LABEL 1,2;"VYHRAL SI!"
2010 GOTO 2030
2020 MOVE 30,20
2025 LABEL 1,2;"PREHRAL SI!"
2030 DISP "TROJUHOLNIK: ";MID$(T$,3*I-2,3)
2040 PAUSE 20
2050 DISP "CHCES SKUSIT ESTE RAZ? (A, N):
2060 INPUT Q0$
2070 IF Q0$="A" OR Q0$="N" THEN 2090
2080 GOTO 2050
2090 RETURN
2200 REM STRANY TROJUHOLNIKA
2210 FOR I=1 TO U
2220 M=R(K,I)
2230 FOR J=1 TO 3
2240 IF T(M,J)<>0 THEN 2270
2250 T(M,J)=L
2260 GOTO 2280
2270 NEXT J
2280 NEXT I
2290 RETURN
2300 REM VYHODNOTENIE
2310 Z0=0 : FOR I=1 TO T
2320 IF T(I,3)=0 THEN 2340
2330 IF T(I,1)=T(I,2) AND T(I,2)=T(I,3) THEN Z0=1 : RETURN
2340 NEXT I
2350 RETURN
2500 REM HLAVICKA PROGRAMU
2510 GCLEAR
2520 RESTORE 2730
2530 MOVE 90,220
2540 LABEL 2,2;"H R A"
2590 MOVE 20,140
2600 LABEL 3,6;"TROJUHOLNIKY"
2610 FOR I=1 TO 10
2620 MOVE 5,130-10*I
2630 READ T$
2640 IF LEFT$(T$,1)="*" THEN 2670
2650 LABEL 1,1;T$
2660 NEXT I
2670 MOVE 70,20
2680 LABEL 1,2;"VELA USPECHOV!"
2690 DISP TAB(30);"STLAC KLAVES ROL!"
2700 INPUT Q0$
2710 RETURN
2730 DATA "SESTUHOLNIK MA 6 VRCHOLOV A,B,C,D,E,F."
2740 DATA "KAZDE 3 VRCHOLY TVORIA TROJUHOLNIK. SPOLU"
2750 DATA "MOZNO VYTVORIT 20 TROJUHOLNIKOV. V NASEJ"
2760 DATA "HRE POSTUPNE TAHA POCITAC A HRAC TAK, ZE"
2770 DATA "POMOCOU DVOJICE VRCHOLOV ZADAVA JEDNU"
2780 DATA "STRANU. PREHRAVA TEN, KTO AKO PRVY VYTVO-"
2790 DATA "RI TROJUHOLNIK!"
2800 DATA "*"
2810 RESTORE
3000 REM UDAJE
3010 DATA 1,2,3,4,1,5,6,7,2,5,8,9,3,6,8,10,4,7,9,10
3020 DATA 1,11,12,13,2,11,14,15,3,12,14,16,4,13,15,16
3030 DATA 5,11,17,18,6,12,17,19,7,13,18,19,8,14,17,20
3040 DATA 9,15,18,20,10,16,19,20
3050 DATA 125,220,200,170,200,120,125,70
3060 DATA 50,120,50,170
```

prvého trojuholníka (napr. obr. 2.43 znázorňuje situáciu v prípade výhry počítača).

Zvolené spojnice sa zadávajú dvojicami bodov, pričom na poradí bodov v dvojici nezáleží (napr. EF a FE označujú tú istú spojnicu).



2.40–2.43 Dialóg s programom TROJUHOLNÍKY

Nemožno voliť dvojicu, ktorá už bola zvolená. Pri odštartovaní hry TROJUHOLNÍKY nesmie byť na počítačoch PMD 85 a Didaktik Alfa nastavený režim rolovania vstupov na obrazovke. Ak ste sa pred odštartovaním programu nachádzali v tomto režime, musíte ho najprv zrušiť súčasným stlačením klávesov SHIFT a PTL na PMD 85, alebo SHIFT a PAL na Didaktiku Alfa.

2.2.8 HRA '15

Túto hru pre vás vymyslel ešte v sedemdesiatych rokoch minulého storočia známy americký hádankár a šachista Samuel Loyd. Hra '15 bola svojho času rovnako populárna ako v nedávnej minulosti Rubikova kocka (a hrá sa doteraz).

V štvorcovej škatuľke s rozmermi 4×4 je vložených 15 štvorčekov, očíslovaných od 1 do 15. Jedno miesto v škatuľke je voľné a umožňuje presúvať štvorčeky tak, aby sme získali ich rôzne rozmiestnenia. Cieľom hry je po „zamiešaní" štvorčekov postupnými presunmi usporiadať štvorčeky do základnej pozície, teda tak, aby boli usporiadané po riadkoch od 1 do 15 a pravý dolný roh škatuľky bol voľný.

František Turnovec

PROGRAM 2.22

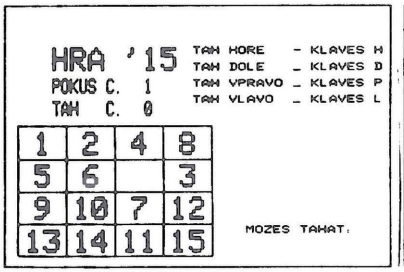
Program HRA '15 (pozri výpis 2.22) zobrazí tabuľku s 15 číslami a náhodne ich zamieša do východiskovej pozície (obr. 2.44). Vašou úlohou je postupným presúvaním čísel vpravo (klávesom P), vľavo (klávesom L), hore (klávesom H) a dole (klávesom D) usporiadať

```
10 REM  PROGRAM 2.22 - HRA '15, 5. 6. 88
20 REM BASIC G (PMD 85, DIDAKTIK ALFA)
25 DIM A(4,4),PR(4),PS(4)
30 M$="" : FOR I=1 TO 40 : M$=M$+" " : NEXT I
40 GOSUB 200
50 GOSUB 2000 : GOSUB 300
60 GOSUB 100
70 IF Q0$="A" THEN 40
80 DISP "KONIEC HRY, DO VIDENIA!" : PAUSE 20
90 END
100 REM PODPROGRAM SERIE POKUSOV
110 H=H+1 : PC=0 : K=0
120 IF H>10 THEN GOSUB 800 : RETURN
130 GOSUB 1200 : GOSUB 1600
140 GOSUB 400
150 IF Q0$="A" THEN 110
160 GOSUB 800
170 RETURN
200 REM PODPROGRAM VSTUPU RIADIACICH UDAJOV
210 RESTORE 2320
220 FOR I=1 TO 4 : READ PR(I),PS(I) : NEXT I
230 RETURN
300 REM PODPROGRAM ODSTARTOVANIA SERIE POKUSOV
310 GCLEAR : H=0 : I1=0 : I2=0
320 SCALE 0,255,0,243
330 MOVE 20,200
340 LABEL 1,2;"MAS 10 POKUSOV. V KAZDOM POKUSE MAS"
345 MOVE 20,180
350 LABEL 1,2;"20-P TAHOV NA USPORIADANIE TABULKY"
355 MOVE 20,160
360 LABEL 1,2;"(KDE P JE PORADOVE CISLO POKUSU)."
370 DISP "HRU ODSTARTUJ VLOZENIM CISLA OD 1 DO 100."
375 INPUT L : IF L<1 OR L>100 THEN 370
380 FOR I=1 TO L : F=RND(1) : NEXT I
390 RETURN
400 REM PODPROGRAM RIADENIA HRY
405 IF K>21-H THEN 700
408 DISP TAB(24);"MOZES TAHAT:
409 BEEP:A=USR(-31583) : DISP M$
410 IF A=72 THEN R=MR+PR(1):S=MS+PS(1):GOTO460
420 IF A=68 THEN R=MR+PR(3):S=MS+PS(3):GOTO460
430 IF A=80 THEN R=MR+PR(4):S=MS+PS(4):GOTO460
440 IF A=76 THEN R=MR+PR(2):S=MS+PS(2):GOTO460
450 GOTO 409
460 IF R>4ORR<1ORS>4ORS<1 THEN DISP TAB(24);"NEMOZNY TAH":GOTO 408
463 MOVE 27,43
465 LABEL 1,2;STR$(K)
470 K=K+1
475 MOVE 27,43
480 LABEL 1,2;STR$(K)
490 P=A(R,S):R1=(4-R)*10+CX:R2=(4-MR)*10+CX:PC=PC+1
500 IF P=>10 THEN S1=(S-1)*10+CX:L=2:S2=(MS-1)*10+CX
510 IF P<10 THEN S1=(S-1)*10+CY:L=1:S2=(MS-1)*10+CY
520 FOR I=1 TO 2:MOVE S1,R1
530 LABEL 2,3;RIGHT$(STR$(P),L)
540 S1=S2:R1=R2:NEXT I
550 A(R,S)=0:A(MR,MS)=P:MR=R:MS=S
560 N=1
570 FOR R=1 TO 4
580 FOR S=1 TO 4
```

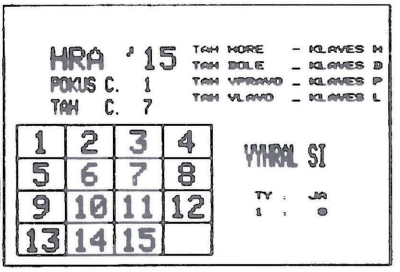
```
590 IF N=16 THEN 620
600 IF A(R,S)<>N THEN 405
610 N=N+1
620 NEXT S
630 NEXT R
640 I1=I1+1
650 FOR I=1 TO 11
660 MOVE 50,30
670 LABEL 1,3;"VYHRAL SI"
680 NEXT I
690 GOTO 750
700 FOR I=1 TO 11
710 MOVE 50,30
720 LABEL 1,3;"SOM LEPSI"
730 NEXT I : I2=I2+1
750 MOVE 50,20
760 LABEL 1,1;" TY :  JA"
770 MOVE 50,15
780 LABEL 1,1;STR$(I1);"  :  ";STR$(I2)
785 DISP TAB(24);"SKUSIS ESTE RAZ? (A,N):" : INPUT Q0$
790 RETURN
800 REM PODPROGRAM ZHODNOTENIA SERIE POKUSOV
810 SCALE 0,255,0,243
820 MOVE 0,0
830 FILL 255,243;1
840 GCLEAR
850 MOVE 20,200
860 LABEL 1,2;"POCET POKUSOV:  ";STR$(H)
870 MOVE 20,170
880 IF I1>12 THEN 920
890 IF I1=12 THEN 940
895 J=I2 : T=I1
900 LABEL 2,3;"VYHRAL SOM"
910 GOTO 950
920 J=I1 : T=I2
925 LABEL 2,3;"VYHRAL SI"
930 GOTO 950
940 J=I1 : T=I2
945 LABEL 2,3;"HRALI SME NEROZHODNE"
950 MOVE 100,100
960 LABEL 3,6;STR$(J);" : ";STR$(T)
970 DISP "CHCES ZACAT NOVU SERIU? (A,N):"
980 INPUT Q0$
990 RETURN
1000 FOR R=1 TO 4
1010 FOR S=1 TO 4
1020 IF A(R,S)=0 THEN 1080
1030 R1=(4-R)*10+CX
1040 IF A(R,S)<10 THEN S1=(S-1)*10+CY:L=1
1050 IF A(R,S)=>10 THEN S1=(S-1)*10+CX:L=2
1060 MOVE S1,R1
1070 LABEL 2,3;RIGHT$(STR$(A(R,S)),L)
1080 NEXT S
1090 NEXT R
1100 RETURN
1200 REM PODPROGRAM VYTVORENIA TABULKY
1210 GCLEAR
1220 SCALE0,80,0,80
1230 POKE-15878,176
1240 FOR I=0 TO 40 STEP 10
1250 MOVE 0,I
1260 PLOT 40,I
1270 MOVE I,0
1280 PLOT I,40
1290 NEXT I
1300 POKE -15878,168
1310 MOVE 10,60
1320 LABEL 2,3;"HRA '15"
1330 MOVE 40,65
1335 LABEL 1,1;"TAH HORE   - KLAVES H"
```

```
1330 MOVE 40,60
1340 LABEL 1,1;"TAH DOLE   - KLAVES D"
1350 MOVE 40,55
1360 LABEL 1,1;"TAH VPRAVO - KLAVES P"
1370 MOVE 40,50
1380 LABEL 1,1;"TAH VLAVO  - KLAVES L"
1390 CX=1.25:CY=3.125:N=1
1400 MOVE 10,50
1410 LABEL 1,2;"POKUS C. "
1420 MOVE 27,50
1430 LABEL 1,2;STR$(H)
1440 MOVE 10,43
1450 LABEL 1,2;"TAH   C. "
1460 MOVE 27,43
1470 LABEL 1,2;STR$(K)
1480 FOR R=1 TO 4
1490 FOR S=1 TO 4:IF N=16 THEN 1510
1500 A(R,S)=N : N=N+1
1510 NEXT S
1520 NEXT R
1525 A(4,4)=0
1530 MR=4:MS=4
1540 GOSUB 1000
1550 RETURN
1600 REM PODPROGRAM ZAMIESANIA HRY
1610 DISP TAB(24);"MIESANIE HRY - CAKAJ!"
1620 N=10+H : A1=0.76
1630 GOSUB 1000
1635 FOR I=1 TO N
1640 A=INT(RND(3)*4)+1
1650 R=MR+PR(A):S=MS+PS(A)
1660 IF S>4 OR S<1 OR R>4 OR R<1 OR A1+A=4 THEN 1640
1670 P=A(R,S):A(R,S)=0:A(MR,MS)=P:MR=R:MS=S:A1=A
1680 NEXT I
1690 GOSUB 1000
1700 RETURN
2000 REM HLAVICKA PROGRAMU
2010 SCALE 0,255,0,242
2020 GCLEAR : RESTORE 2230
2090 MOVE 40,140
2100 LABEL 2,6;"H R A    ' 1 5"
2110 FOR I=1 TO 10
2120 MOVE 5,130-10*I
2130 READ T$
2140 IF LEFT$(T$,1)="*" THEN 2170
2150 LABEL 1,1;T$
2160 NEXT I
2170 MOVE 70,20
2180 LABEL 1,2;"VELA USPECHOV!"
2190 DISP TAB(30);"STLAC KLAVES EOL!"
2200 INPUT Q0$
2220 RETURN
2230 DATA "V TABULKE 4X4 JE ROZLOZENYCH 15 CISEL OD"
2240 DATA "1 DO 15. JEDNO POLICKO TABULKY JE PRAZDNE."
2250 DATA "CIELOM HRY JE POSTUPNYMI PRESUNMI CISEL"
2260 DATA "NA VOLNE POLICKO USPORIADAT TABULKU TAK,"
2270 DATA "ABY CISLA BOLI V TABULKE USPORIADANE OD"
2280 DATA "NAJMENSIEHO PO NAJVACSIE A VOLNE POLICKO"
2290 DATA "BOLO V PRAVOM DOLNOM ROHU. CISLA MOZEME"
2300 DATA "PRESUVAT IBA VPRAVO, VLAVO, HORE A DOLE."
2310 DATA "*"
2320 DATA 1,0,0,1,-1,0,0,-1
```

tabuľku do základnej pozície (obr 2.45). Nie z každej východiskovej pozície možno vytvoriť postupnými presunmi základnú pozíciu, počítač však pripúšťa iba také východiskové pozície, ktoré umožňujú úspešné riešenie.



2.44 Začiatok hry '15



2.45 Výhra v hre '15

Na usporiadanie tabuľky máte k dispozícii určitý počet presunov (ťahov) – v prvej partii je to 20 ťahov. Ak sa vám po ich využití nepodarí vytvoriť základnú pozíciu, prehrali ste a počítač vám ponúkne ďalší pokus s novou východiskovou pozíciou. Hrá sa na 10 partií, pričom v každej ďalšej partii sa počet povolených ťahov znižuje. Po ukončení série partií počítač zhodnotí výsledky a v hre môže pokračovať ďalší záujemca.

2.2.9 HRA ŤAH JAZDCOM

Hry na šachovnici patria k najstarším súčastiam ľudskej kultúry. Niektorí historici tvrdia, že v Indii bola hra podobná dnešnému šachu známa už v 15. storočí pred naším letopočtom. Je zaujímavé, že od prvých detských krôčikov výpočtovej techniky sa matematici a programátori snažili naučiť počítač hrať šach. Už v r. 1949 sa teóriou šachovej hry na počítači zaoberal americký matematik C. Shannon. Majster sveta v šachu z rokov 1948–1957, 1958–1959 a 1961, známy sovietsky šachista M. M. Botvinnik sa po skončení svojej aktívnej športovej kariéry venoval vývoju šachových programov. Od roku 1970 sa v USA konajú pravidelné šachové turnaje počítačov a v roku 1974 sa v Stockholme uskutočnili prvé majstrovstvá sveta v tejto novej „športovej disciplíne". Objavili sa špecializované šachové počítače, s ktorými si môžete zahrať šach a nastaviť si pritom rôzne úrovne hry počítača. Najvyspelejšie počítačové programy a špecializované šachové počítače dokážu dnes s úspechom vzdorovať aj šachovému majstrovi.

Menej dokonalé programy, ktoré hrajú šach, sú k dispozícii aj na počítačoch PMD 85, Didaktik Alfa a Atari. Ide, prirodzene, o programy pomerne zložité a značne rozsiahle, na zostavenie kto-

rých by sme nevystačili s prostriedkami jazyka basic. Preto vám ponúkame program jednoduchšej hry na šachovnici.

PROGRAM 2.23

Výpis 2.23 obsahuje program hry ŤAH JAZDCOM pre počítače PMD 85 a Didaktik Alfa. Pravidlá hry sa zobrazia na úvodnej obrazovke po odštartovaní programu. V ľavom hornom rohu šachovnice je šachový jazdec. V každej partii hry počítač a hráč robia postupne šachové ťahy jazdcom, pričom cieľom hry je umiestniť jazdca do ľubovoľného zo štyroch políčok pravého dolného rohu šachovnice (v šachovej notácii g1, g2, h1 alebo h2). Na začiatku partie máte možnosť rozhodnúť sa, či chcete urobiť prvý ťah (pozri obr 2.46). Podľa vašej voľby potom prebieha partia postupným striedaním ťahov počítača a vašich ťahov, pričom v každej pozícii na šachovnici je smer ďalších ťahov určený pravidlom „uhlopriečky": možno ťahať iba do tých políčok šachovnice, ktoré sú pod uhlopriečkou políčok (prebiehajúcou zľava dole smerom doprava hore), ktorá prechádza políčkom, na ktorom sa práve nachádza jazdec. Aby ste sa nepomýlili, počítač vyznačí na šachovnici všetky možnosti ťahu

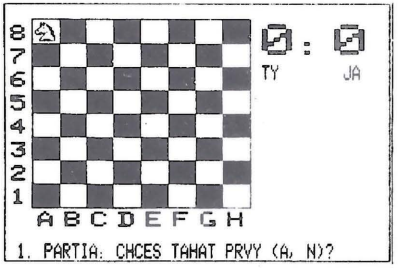
```
10 REM PROGRAM 2.23 - TAH JAZDCOM, 4. 6. 88
20 REM BASIC G (PMD 85, DIDAKTIK ALFA)
60 SCALE 0,255,0,242
70 M$="" : FOR I=1 TO 40 : M$=M$+" " : NEXT I
80 GOSUB 7000
90 RESTORE
100 S1=0 : S2=0 : P=0 : GOSUB 6500
200 MOVE 170,194
210 LABEL 4,4;"0:0"
220 MOVE 170,170
230 LABEL 1,2;"TY        JA"
240 T=0 : I=1 : J=8 : P=P+1 : GOSUB 4000
270 MOVE 0,0
280 LABEL 1,2;STR$(P);". PARTIA: CHCES TAHAT PRVY (A, N)?"
290 A$=CHR$(USR(-31583)) : DISP M$
300 IF A$="A" OR A$="N" THEN 330
310 BEEP : DISP "ZLE - VLOZ A ALEBO N!" : GOTO 290
330 GOSUB 6100
340 IF A$="N" THEN 600
350 IF A$="A" THEN T=T+1
360 GOSUB 2000
370 IF N=0 THEN 950
380 IF N=1 THEN 850
390 GOSUB 3000
400 MOVE 0,0
410 LABEL 1,2;"TVOJ TAH?"
420 X=USR(-31583)-64 : DISP M$
430 IF X>=1 AND X<=N THEN 480
450 BEEP : DISP "ZLE - VLOZ SPRAVNE PISMENO!" : GOTO 420
480 GOSUB 6100
490 GOSUB 3000
500 GOSUB 4000
510 I=ABS(I(X)) : J=J(X) : GOSUB 4000
530 MOVE 170,160-20*T
```

```
535 IF A$="N" THEN MOVE 170,140-20*T
540 LABEL 1,2;CHR$(T+48);".";CHR$(I+64);CHR$(J+48)
600 GOSUB 2000
610 IF N=0 THEN 1050
620 IF N=1 THEN 880
630 MOVE 0,0
640 LABEL 1,2;"MOJ TAH:"
650 GOSUB 3000
670 L=0 : FOR I1=1 TO N-1 : PAUSE 10
690 IF L=0 THEN GOSUB 5500
700 IF L=0 AND K=0 THEN L=1
710 IF L=1 THEN GOSUB 5570
720 GOSUB 3050
730 I(K)=0 : NEXT I1 : GOSUB 5570
760 GOSUB 3050
770 GOSUB 4000
780 I=ABS(I(K)) : J=J(K) : GOSUB 4000
800 IF A$="N" THEN T=T+1
810 MOVE 212,160-20*T
820 LABEL 1,2;CHR$(T+48);".";CHR$(I+64);CHR$(J+48)
830 GOSUB 6100
840 GOTO 350
850 GOSUB 6000
854 MOVE 0,0
855 LABEL 1,2;"TVOJ TAH, MAS JEDINU MOZNOST:"
860 GOSUB 6000
870 X=1 : GOTO 500
880 GOSUB 6000
885 MOVE 0,0
890 LABEL 1,2;"MOJ TAH, NEMAM INU MOZNOST:"
900 GOSUB 6000
910 K=1 : GOTO 770
950 Y$="VYHRAL SOM !!!" : S1=S1+1 : MOVE 194,194
980 LABEL 4,4;STR$(S1-1)
990 MOVE 194,194
1000 LABEL 4,4;STR$(S1)

1010 MOVE 0,0
1020 LABEL 1,2;Y$
1040 GOTO 1400
1050 MOVE 146,194
1060 LABEL 4,4;STR$(S2)
1070 S2=S2+1 : GOSUB 6100
1080 MOVE 146,194
1090 LABEL 4,4;STR$(S2)
1100 Y$="VYHRAL SI !!!" : MOVE 0,0
1120 LABEL 1,2;Y$
1400 GOSUB 4000
1410 MOVE 170,0
1420 POKE -15878,175
1430 FILL 12,160;63
1440 POKE -15878,168
1450 GOSUB 6100
1460 IF P>=10 THEN 1600
1500 MOVE 0,0
1510 LABEL 1,2;"CHCES POKRACOVAT (A, N)?"
1520 A$=CHR$(USR(-31583)) : DISP M$ : GOSUB 6100
1530 IF A$="A" THEN 240
1540 IF A$="N" THEN 1600
1550 BEEP : DISP "ZLE - VLOZ A ALEBO N!" : GOTO 1500
1600 MOVE 0,0
1610 LABEL 1,2;"KONIEC HRY"
1620 GOSUB 6000
1630 MOVE 0,0
1640 IF S2=0 THEN LABEL 1,2;"UPLNE SOM TA ROZDRVIL"
1650 IF S1=0 THEN LABEL 1,2;"UPLNE SI MA ROZDRVIL, BLAHOZELAM"
1660 IF S1>S2 AND S2>0 THEN LABEL 1,2;"BOL SOM LEPSI, ALE BOJOVAL SI"
1670 IF S1<S2 AND S1>0 THEN LABEL 1,2;"BOL SI LEPSI, ALE NABUDUCE UVIDIME"
1680 IF S1=S2 THEN LABEL 1,2;"NIC NIE JE ROZHODNUTE, SKUSME ESTE RAZ"
1690 GOSUB 6000
1700 MOVE 0,0
```

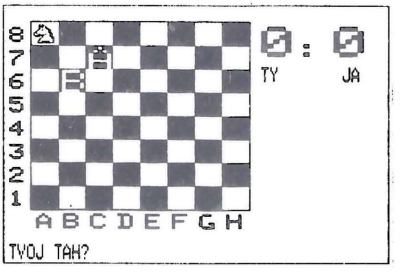
```
1710 LABEL 1,2;"CHCES SKUSIT ESTE RAZ (A,N)?"
1720 A$=CHR$(USR(-31583)) : IF A$="A" THEN 80
1730 REM KONIEC HRY
1740 GCLEAR
1750 MOVE 10,220
1760 LABEL 1,2;"KONIEC PROGRAMU"
1770 MOVE 10,150
1780 LABEL 2,5;"T A H   K O N O M"
1790 MOVE 50,30
1800 LABEL 2,2;"DO VIDENIA"
1810 GOTO 8000
2000 REM ANALYZA TAHU
2010 N=0 : M=0 : I1=I+2 : J1=J+1 : GOSUB 2100
2030 J1=J1-2 : GOSUB 2100
2050 I1=I+1 : J1=J-2 : GOSUB 2100
2070 I1=I1-2
2100 IF I1<1 OR I1>8 OR J1<1 OR J1>8 THEN RETURN
2110 N=N+1 : I(N)=-I1 : J(N)=J1
2120 IF((I1)2 AND I1(5) OR I1)6)AND((J1(7 AND J1)4)OR J1(3) THEN I(N)=I1
2130 M=M+1 : RETURN
3000 REM ZOBRAZENIE MOZNYCH TAHOV
3010 FOR K=1 TO N : GOSUB 3050
3020 NEXT K : RETURN
3050 MOVE 2+ABS(I(K))*18,28+22*J(K)
3060 LABEL 3,3;CHR$(K+64)
3070 RETURN
4000 REM ZOBRAZENIE KONA
4010 BMOVE 2+3*I,194-22*J
4020 BPLOT K$,3
4030 RETURN
5500 K=INT(RND(1)*N)+1
5510 FOR X=1 TO N : GOSUB 5590
5520 IF I(K)<0 THEN RETURN
5530 NEXT X : K=0 : RETURN
5570 K=INT(RND(1)*N)+1
5580 IF I(K)<>0 THEN RETURN
5590 K=K+1 : IF K>N THEN K=1
5610 GOTO 5580
6000 PAUSE 15
6100 POKE -15878,175
6110 MOVE 0,0
6120 FILL 39,16;63
6130 POKE -15878,168
6140 RETURN
6500 REM ZOBRAZENIE SACHOVNICE
6510 GCLEAR
6520 K$="" : FOR I=1 TO 54 : READ X
6530 K$=K$+CHR$(X) : NEXT I
6550 FOR L=0 TO 3
6560 FOR T=1 TO 2
6570 FOR C=0 TO 1
6580 MOVE 18+C*72,28+22*T+44*L
6590 FILL 18,22;5*T
6600 NEXT C
6610 MOVE 0,31+22*T+44*L
6620 LABEL 2,2;RIGHT$(STR$(T+2*L),1)
6630 NEXT T : NEXT L
6640 MOVE 21,32
6650 FOR C=1 TO 8 : LABEL 2,2;CHR$(64+C)
6670 LABEL 1,1;" "
6680 NEXT C : MOVE 18,50
6700 PLOT 18,227
6710 PLOT 163,227
6720 PLOT 163,50
6730 PLOT 18,50
6750 RETURN
7000 REM HLAVICKA PROGRAMU
7010 GCLEAR
7020 MOVE 80,220
7030 LABEL 2,2;"H R A"
7080 MOVE 40,150
```

```
7090 LABEL 2,5;"TAH JAZDCOM"
7100 MOVE 10,130
7110 LABEL 1,1;"V LAVOM HORNOM ROHU SACHOVNICE JE JAZ-"
7120 MOVE 10,120
7130 LABEL 1,1;"DEC. V KAZDEJ PARTII HRY POCITAC A HRAC"
7140 MOVE 10,110
7150 LABEL 1,1;"POSTUPNE ROBIA SACHOVY TAH JAZDCOM. VYHRAVA"
7160 MOVE 10,100
7170 LABEL 1,1;"TEN ZO SUPEROV, KTORY AKO PRVY UMIESTNI"
7180 MOVE 10,90
7190 LABEL 1,1;"JAZDCA DO LUBOVOLNEHO ZO STYROCH POLICOK"
7200 MOVE 10,80
7210 LABEL 1,1;"PRAVEHO DOLNEHO ROHU SACHOVNICE. HRAC SA"
7220 MOVE 10,70
7230 LABEL 1,1;"MOZE NA ZACIATKU PARTIE ROZHODNUT, CI"
7240 MOVE 10,60
7250 LABEL 1,1;"ROBI PRVY TAH SAM, ALEBO HO PRENECHA PO-"
7260 MOVE 10,50
7270 LABEL 1,1;"CITACU. HRA MA 10 PARTII A NA ZAVER PO-"
7280 MOVE 10,40
7290 LABEL 1,1;"CITAC ZHODNOTI VYSLEDKY."
7300 MOVE 60,10
7310 LABEL 1,2;"VELA USPECHOV!"
7320 DISP "                    STLAC KLAVES EOL!"
7330 INPUT Q0$ : RETURN
8000 END
8100 DATA 0,5,0,0,13,0,0,23,0,0,53,0,0,33,1,32,4,3
8110 DATA 16,0,3,8,0,3,20,0,6,4,17,6,36,14,6,56,5,6
8120 DATA 0,4,12,0,2,12,0,1,8,0,1,8,32,60,9,32,63,15
```

veľkými písmenami (pozri obr. 2.47) a svoju voľbu vykonáte stlačením príslušného písmena. Používame tu priame čítanie klávesu, takže netreba stláčať EOL. Každá hra má 10 partií a po jej skončení počítač zhodnotí výsledky. Hoci ide o jednoduchú hru, nie je ľahké nad počítačom zvíťaziť.



2.46 Voľba prvého ťahu v hre ŤAH JAZDCOM



2.47 Prvý ťah v hre ŤAH JAZDCOM

V programe si všimnite využívanie príkazu FILL na zvýrazňovanie časti obrazovky spolu s príkazmi MOVE, PAUSE a POKE (úsek programu od riadku 6000 do 6140). Na zobrazenie jazdca používame príkazy BMOVE a BPLOT, v riadkoch 8100 až 8120 definujeme

v príkazoch DATA motív jazdca. Na zobrazenie šachovnice používame príkaz FILL v cykle v riadkoch 6550 až 6630.

2.2.10 HRA DÁMA

Iste každý z vás pozná hru DÁMA. Skúste si ju zahrať s počítačom.

PROGRAM 2.24

Program hry DÁMA (pozri výpis 2.24) neoplýva efektnou grafikou ani programátorskými zaujímavosťami. Pokúsili sme sa použiť iba tie prvky basicu, ktoré fungujú na väčšine osobných počítačov. Náš program je v Atari basicu, ľahko ho však upravíme pre počítač PMD 85: dvojice príkazov POSITION a PRINT v riadkoch 1430 a 1500 treba nahradiť príkazmi PRINT s použitím funkcie TAB a príkaz PRINT CHR$(125); v riadkoch 1150 a 2010 príkazom GCLEAR.

```
10 REM Program 2.24 - Hra DAMA, 8. 5. 88
20 REM Atari basic
30 DIM R(4),S(7,7):G=-1:R(0)=-99:I=0:Q=0
40 FOR L=0 TO 7:FOR J=0 TO 7:S(L,J)=0:NEXT J:NEXT
   L:FOR K=1 TO 4:R(K)=0:NEXT K
50 GOSUB 2000
60 GOSUB 100
70 GOSUB 200
80 END
100 REM Priprava hry
110 FOR X=0 TO 7
120 FOR Y=0 TO 7
130 READ J
140 IF J=15 THEN 170
150 S(X,Y)=J
160 GOTO 180
170 RESTORE :READ J:S(X,Y)=J
180 NEXT Y:NEXT X
190 RETURN
200 REM Podprogram riadenia hry
210 KON=0:FOR X=0 TO 7:FOR Y=0 TO 7
220 IF S(X,Y)<0 THEN KON=KON+1
230 IF S(X,Y)>-1 THEN 260
240 IF S(X,Y)=-1 THEN FOR A=-1 TO 1 STEP 2:B=G:GOS
   UB 650:NEXT A
250 IF S(X,Y)=-2 THEN FOR A=-1 TO 1 STEP 2:FOR B=-
   1 TO 1 STEP 2:GOSUB 650:NEXT B:NEXT A
260 NEXT Y:NEXT X
270 IF KON=0 THEN PRINT CHR$(125);:GOTO 1420
280 GOTO 1150
650 U=X+A:V=Y+B
660 IF U<0 OR U>7 OR V<0 OR V>7 THEN 870
740 IF S(U,V)=0 THEN GOSUB 910:GOTO 870
770 IF S(U,V)<0 THEN 870
790 U=U+A:V=V+B
800 IF U<0 OR V<0 OR U>7 OR V>7 THEN 870
850 IF S(U,V)=0 THEN GOSUB 910
870 RETURN
```

```
910 IF V=0 AND S(X,Y)=-1 THEN Q=Q+2
920 IF ABS(Y-V)=2 THEN Q=Q+5
960 IF Y=7 THEN Q=Q-2
980 IF Y=0 OR U=7 THEN Q=Q+1
1030 FOR C=-1 TO 1 STEP 2:IF U+C<0 OR U+C>7 OR V+G
    <0 THEN 1080
1035 IF S(U+C,V+G)<0 THEN Q=Q+1:GOTO 1080
1040 IF U-C<0 OR U-C>7 OR V-G>7 THEN 1080
1045 IF S(U+C,V+G)>0 AND (S(U-C,V-G)=0 OR (U-C=X A
    ND V-G=Y)) THEN Q=Q-2
1080 NEXT C
1090 IF Q>R(0) THEN R(0)=Q:R(1)=X:R(2)=Y:R(3)=U:R(
    4)=V
1100 Q=0:RETURN
1150 PRINT CHR$(125);
1230 PRINT "MOJ TAH:  Z  ";R(1);",";R(2);" NA ";R(
    3);",";R(4);:R(0)=-99
1240 IF R(4)=0 THEN S(R(3),R(4))=-2:GOTO 1260
1250 S(R(3),R(4))=S(R(1),R(2))
1260 S(R(1),R(2))=0
1270 IF ABS(R(1)-R(3))<>2 THEN 1420
1280 S((R(1)+R(3))/2,(R(2)+R(4))/2)=0
1290 X=R(3):Y=R(4)
1300 IF S(X,Y)=-1 THEN B=-2:FOR A=-2 TO 2 STEP 4:G
    OSUB 1370
1310 IF S(X,Y)=-2 THEN FOR A=-2 TO 2 STEP 4:FOR B=
    -2 TO 2 STEP 4:GOSUB 1370:NEXT B
1320 NEXT A
1330 IF R(0)<>-99 THEN PRINT " NA  ";R(3);",";R(4)
    ;:R(0)=-99:GOTO 1240
1340 GOTO 1420
1370 U=X+A:V=Y+B:IF U<0 OR U>7 OR V<0 OR V>7 THEN
    1400
1380 IF S(U,V)=0 AND S(X+A/2,Y+B/2)>0 THEN GOSUB 9
    10
1400 RETURN
1410 REM Zobrazenie sachovnice
1420 PRINT :PRINT :PRINT
1430 FOR Y=7 TO 0 STEP -1:POSITION 2,16-2*Y:PRINT
    Y;:FOR X=0 TO 7:SP=5+4*X:POSITION SP,16-2*Y
1440 IF S(X,Y)=0 THEN PRINT ".";
1450 IF S(X,Y)=1 THEN PRINT "O";
1460 IF S(X,Y)=-1 THEN PRINT "X";
1470 IF S(X,Y)=-2 THEN PRINT "X*";
1480 IF S(X,Y)=2 THEN PRINT "O*";
1490 NEXT X:PRINT :NEXT Y:PRINT
1500 FOR X=0 TO 7:POSITION 5+4*X,18:PRINT X;:NEXT
    X:POSITION 2,20
1510 FOR K=0 TO 7:FOR L=0 TO 7:IF S(K,L)>0 THEN 15
    30
1520 NEXT L:NEXT K:GOTO 1880
1530 IF KON=0 THEN 1890
1580 PRINT "TVOJ TAH: ";
1590 PRINT "Z  ";:INPUT E,H:X=E:Y=H
1600 IF S(X,Y)<=0 THEN PRINT "Chyba...  ";:GOTO 15
    90
1670 PRINT "          NA ";
1675 INPUT A,B:X=A:Y=B
1680 IF S(X,Y)=0 AND ABS(A-E)<=2 AND ABS(A-E)=ABS(
    B-H) THEN 1750
1690 PRINT "Chyba...  ";:GOTO 1590
1750 S(A,B)=S(E,H):S(E,H)=0
1760 IF ABS(E-A)<>2 THEN 1860
1800 S((E+A)/2,(H+B)/2)=0
1820 PRINT "          +NA ";
1830 INPUT A1,B1:IF A1<0 THEN 1860
1840 IF S(A1,B1)<>0 OR ABS(A1-A)<>2 OR ABS(B1-B)<>
    2 THEN PRINT "Chyba... +NA ";:GOTO 1830
1850 E=A:H=B:A=A1:B=B1:GOTO 1750
1860 IF B=7 THEN S(A,B)=2
```

```
1870 GOTO 210
1880 PRINT "VYHRAL SOM!!!":GOTO 1900
1890 PRINT "VYHRAL SI!!!"
1900 RETURN
2000 REM Uvodna informacia
2010 PRINT CHR$(125);
2020 PRINT "Zahraj si so mnou damu!"
2030 PRINT
2040 PRINT "Policka sachovnice su ocislovane od"
2050 PRINT "0 do 7 zlava doprava a od 0 do 7 zdo-"
2060 PRINT "la hore. Moje figurky su oznacene X a"
2070 PRINT "tvoje O, damy X* a O*. Tah vkladas "
2080 PRINT "odpovedou na dotazy"
2090 PRINT "    Z"
2100 PRINT "    NA"
2110 PRINT "vlozenim dvojice cisel s,r (s=stlpec,"
2120 PRINT "r=riadok), urcujucich vychodiskove a"
2130 PRINT "koncove policko tahu. V pripade, ze"
2140 PRINT "preskocis a zoberies moju figurku,"
2150 PRINT "spytam sa este"
2160 PRINT "  +NA"
2170 PRINT "Ak vidis moznost dalsieho skoku,"
2180 PRINT "vlozis dvojicu cisel, urcujucich"
2190 PRINT "koncove policko dalsieho skoku,"
2200 PRINT "v opacnom pripade vlozis dvojicu"
2210 PRINT "cisel -1,-1. Nerob naraz viacna-"
2220 PRINT "sobne skoky, cakaj na pokyn! Damy"
2230 PRINT "mozu brat iba susedne figurky.":PRINT
2240 PRINT "                    Cakaj..."
2250 FOR K=1 TO 1500:NEXT K
2260 RETURN
3000 DATA 1,0,1,0,0,0,-1,0,0,1,0,0,0,-1,0,-1,15
```

Po zobrazení úvodnej informácie sa na obrazovke vykreslí šachovnica s figúrkami – vaše figúrky sú zobrazené znakom O a figúrky počítača znakom X. Dámy sa označujú pripísaním hviezdičky, teda O* a X*. Prvý ťah robí počítač. Políčka šachovnice sú očíslované od 0 do 7 zľava doprava a od 0 do 7 zdola nahor (políčko v ľavom dolnom rohu šachovnice má teda súradnice 0, 0). Ťah vkladáme odpoveďou na dotaz

Tvoj ťah: z
na

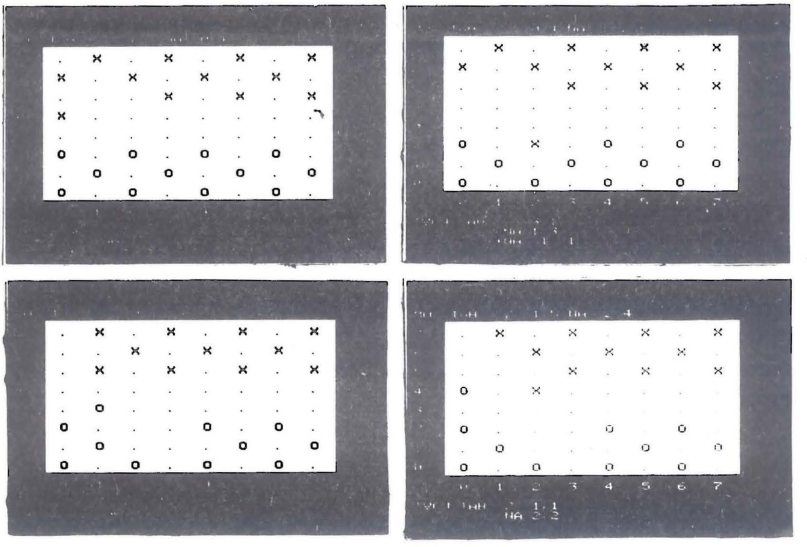
vložením dvojice čísel, určujúcich východiskové a koncové políčko ťahu. Napr. váš prvý ťah môže vyzerať takto:

Tvoj ťah: z 0,2
na 1,3

V prípade, že preskočíte a zoberiete figúrku počítača, objaví sa ešte dotaz

+na

Ak vidíte možnosť ďalšieho skoku a zisku figúrky, vložíte ďalšiu dvojicu čísel určujúcu koncové políčko pokračovania ťahu, v opačnom prípade vložíte dvojicu čísel –1,–1, ktorá ukončí váš ťah. Pre pohyb dámy sa používajú zjednodušené pravidlá: dáma sa síce môže



2.48–2.51 Hráme DÁMU s počítačom

pohybovať vo všetkých smeroch, postupuje však iba o jednu pozíciu a preskakovať a brať môže iba susedné figúrky (podobne ako každá iná figúrka). Nepokúšajte sa preto o väčšie skoky dámy.

Na obrázkoch 2.48 až 2.51 uvádzame zobrazenie prvých štyroch ťahov partie hry. Program, podľa ktorého hrá počítač, nie je príliš dokonalý, takže máte šancu zvíťaziť. Počítač nekontroluje vašu povinnosť brať, preto regulárny priebeh hry závisí od toho, ako zachovávate pravidlá.

2.2.11 HRA HÚSENICA

Hru HÚSENICA nájdete pod rôznymi názvami v zbierkach hier pre väčšinu osobných počítačov (napr. pod anglickým názvom CATERPILLAR). Niektorí z vás sa možno už stretli s jej verziou pod názvom CHROBÁK WURMI na počítačoch PMD 85. Hra HÚSENICA je dobrou ilustráciou spojenia grafiky s programovaním pohybu (tzv. animáciou).

František Turnovec

PROGRAM 2.25

Výpis 2.25 obsahuje program hry HÚSENICA pre počítače PMD 85 a Didaktik Alfa.

Najprv niekoľko slov o pravidlách našej verzie hry. Po odštartovaní programu a úvodnej informácii sa na obrazovke zjaví obdĺžnikové pole a uprostred neho malá húsenica. Pole sa zaplní lístočkami a krížikmi (pozri obr. 2.52). Funkčnými klávesmi riadime pohyb húsenice. Húsenicou treba pohybovať tak, aby pri svojej ceste nazbierala čo najviac lístočkov. Za každý lístoček, ktorý húsenica „zje" na svojej ceste, získate bod a húsenica sa pritom zväčší o jeden diel. Pohyb treba riadiť tak, aby húsenica obišla prekážky (krížiky), vyhýbala sa okrajom herného poľa a „nenarazila" do svojho zväčšujúceho sa tela.

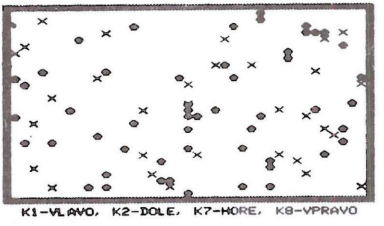
```
10 REM PROGRAM 2.25 - HRA HUSENICA, 10. 5. 87
20 REM BASIC G (PMD 86, DIDAKTIK ALFA)
30 DIM X(999),Y(999)
40 DEF FNCA(WW)=6*WW+40
50 M$="" : FOR I=1 TO 48 : M$=M$+" " : NEXT I
60 SCALE 0,255,0,242
100 GOSUB 2000
110 RESTORE
120 T$="" : H$="" : P$="" : HT$="" : J$="" : N$=""
130 GOSUB 200
140 GOSUB 500
150 GOSUB 1000
190 END
200 REM ZOBRAZENIE TABULKY
210 GCLEAR
230 FOR I=1 TO 6 : READ A : HT$=HT$+CHR$(A) : NEXT
240 FOR I=1 TO 6 : READ A : P$=P$+CHR$(A) : NEXT I
250 FOR I=0 TO 40 : BMOVE I,FNCA(30)
260 BPLOT P$,1 : BMOVE I,40 : BPLOT P$,1 : NEXT I
270 FOR I=1 TO 29 : BPLOT P$,1 : NEXT I
280 BMOVE 0,46 : FOR I=1 TO 29 : BPLOT P$,1 : NEXT I
290 FOR I=1 TO 6 : READ A : T$=T$+CHR$(A) : NEXT I
300 FOR I=1 TO 6 : READ A : H$=H$+CHR$(A) : NEXT I
310 FOR I=1 TO 6 : READ A : N$=N$+CHR$(A) : NEXT I
320 FOR I=1 TO 6 : READ A : J$=J$+CHR$(A) : NEXT I
330 BMOVE 20,FNCA(15) : BPLOT H$,1 : BPLOT T$,1
340 BPLOT T$,1
350 FOR I=1 TO 50 : GOSUB 400 : BPLOT J$,1 : NEXT I
360 FOR I=1 TO 30 : GOSUB 400 : BPLOT N$,1 : NEXT I
370 RETURN
400 REM NAHODNE ROZMIESTNENIE LISTKOV
410 X=INT(RND(1)*39)+1 : Y=INT(RND(1)*29)*6+46
420 BMOVE X,Y+3  : IF USR(-31501)>0 THEN 410
430 BMOVE X,Y
440 RETURN
500 REM POHYB HUSENICE
510 MOVE 0,5
520 LABEL 1,1;"K1-VLAVO, K2-DOLE, K7-HORE, K8-VPRAVO"
550 Z=2 : A=1000 : CM=2
560 FOR I=0 TO 2 : X(I)=17-I : Y(I)=20 : NEXT I
570 FOR I=0 TO 8 : READ A(I),B(I) : NEXT I
580 DISP "ZISKAL SI:";CM-2;TAB(20);"DOTERAJSI REKORD:";RC
600 BEEP
610 IN=INKEY : IF IN>8 THEN 610
620 IF IN=0 OR (IN)2 AND IN(7) THEN 610
630 GOTO 660
```

```
640 IF CM<50 THEN FOR I=1 TO 99 : NEXT I
650 IN=INKEY
660 IF IN<9 THEN M=IN
665 Q=Z-CM : IF Q<0 THEN Q=Q+A
670 Z1=Z+1 : IF Z1>A THEN Z1=Z1-A
680 X(Z1)=X(Z)+A(M) : Y(Z1)=Y(Z)+B(M)
690 BMOVE Y(Z1),FNCA(X(Z1))+4 : C=USR(-31501)
700 BMOVE Y(Q),FNCA(X(Q)) : BPLOT T$,1
710 X(Q)=0 : Y(Q)=0
720 BMOVE Y(Z),FNCA(X(Z)) : BPLOT HT$,1
730 Z=Z1 : BMOVE Y(Z),FNCA(X(Z))
740 IF C=0 THEN BPLOT H$,1 : GOTO 640
750 IF C<>12 THEN RETURN
760 BPLOT J$,1
770 BMOVE Y(Z),FNCA(X(Z)) : BPLOT H$,1
780 OUT 134,1 : CM=CM+1 : DISP M$
785 DISP "ZISKAL SI:";CM-2;TAB(20);"DOTERAJSI REKORD:";RC
790 GOSUB 400
800 BPLOT J$,1
810 OUT 134,0
820 IF CM<>50 THEN 640
830 DISP "VYBORNE, POKRACUJ V ROVNAKOM STYLE"
840 FOR I=1 TO 200  : OUT 134,1 : OUT 134,0 : NEXT I
850 DISP M$
860 DISP "ZISKAL SI:";CM-2;TAB(20);"DOTERAJSI REKORD:";RC
870 GOTO 640
900 RETURN
1000 REM HODNOTENIE
1010 DISP M$
1020 IF C=18 THEN DISP "KONIEC, SOM V PASTI"
1030 IF C=30 THEN DISP "KONIEC, ZJEDLA SOM SAMA SEBA"
1040 IF C=21 THEN DISP "KONIEC, HLAVOU MUR NEPRERAZIM"
1050 PAUSE 20
1060 GCLEAR
1070 MOVE 10,210
1080 LABEL 1,2;"ZISKAL SI"
1090 MOVE 100,150
1100 LABEL 3,6;CM-2
1110 H=CM-2 : GOSUB 1500
1120 MOVE 10,130
1130 LABEL 1,2;H0$
1140 IF H<=RC THEN 1250
1160 LABEL 1,2;", TO JE NOVY REKORD, STARY BOL"
1170 MOVE 100,60
1180 H=RC : GOSUB 1500
1190 LABEL 3,6;RC
1200 MOVE 10,40
1210 LABEL 1,2;H0$
1220 RC=CM-2
1230 GOTO 1310
1250 LABEL 1,2;", ALE TO TU UZ BOLO, REKORD JE"
1260 H=RC : GOSUB 1500
1270 MOVE 100,60
1280 LABEL 3,6;RC
1290 MOVE 10,40
1300 LABEL 1,2;H0$
1310 DISP "CHCES POKRACOVAT? (A, ALEBO N):"
1320 Q0$=CHR$(USR(-31583)) : IF Q0$="A" THEN 110
1330 DISP "KONIEC HRY, DO VIDENIA
1340 PAUSE 20
1400 RETURN
1500 REM SKLONOVANIE
1510 L$="BOD"
1520 IF H=1 THEN H0$=L$
1530 IF H>1 AND H<5 THEN H0$=L$+"Y"
1540 IF H>4 OR H=0 THEN H0$=L$+"OV"
1550 RETURN
2000 REM HLAVICKA PROGRAMU
2010 GCLEAR
2020 MOVE 80,220
```

```
2030 LABEL 2,2$"H R A"
2080 MOVE 20,140
2090 LABEL 2,6$"H U S E N I C A"
2100 MOVE 10,120
2110 LABEL 1,1$"PROGRAM UMOZNUJE RIADIT FUNKCNYMI KLAVES-"
2120 MOVE 10,110
2130 LABEL 1,1$"MI POHYB HUSENICE NA OBRAZOVKE. HUSENICA"
2140 MOVE 10,100
2150 LABEL 1,1$"SA MUSI VYHYBAT PREKAZKAM A PO CESTE ZBE-"
2160 MOVE 10,90
2170 LABEL 1,1$"RA LISTOCKY. ZA KAZDY LISTOCEK ZISKAS BOD."
2180 MOVE 10,80
2190 LABEL 1,1$"HRA SA SKONCI STRETNUTIM HUSENICE S PRE-"
2200 MOVE 10,70
2210 LABEL 1,1$"KAZKOU, OKRAJOM POLA NA OBRAZOVKE ALEBO"
2220 MOVE 10,60
2230 LABEL 1,1$"S VLASTNYM CHVOSTOM."
2250 MOVE 70,20
2260 LABEL 1,2$"VELA USPECHOV!"
2270 DISP "                    STLAC KLAVES EOL!"
2280 INPUT Q$$
2290 RETURN
3000 DATA 45,12,0,18,45,18
3010 DATA 21,42,21,42,21,42
3020 DATA 12,30,63,63,30,12
3030 DATA 33,18,63,45,51,30
3040 DATA 33,18,12,12,18,33
3050 DATA 12,30,63,63,12,30
3060 DATA 0,0,0,-1,1,0
3070 DATA 0,0,0,0,0,0
3080 DATA 0,0,-1,0,0,1
```

Pohyb húsenice riadime stláčaním funkčných klávesov:

K1 pohyb vľavo,
K2 pohyb dole,
K7 pohyb hore,
K8 pohyb vpravo



2.52 Herné pole v hre HÚSENICA

Po stlačení príslušného klávesu sa húsenica začne pohybovať. Smer jej pohybu sa zmení až po stlačení ďalšieho klávesu. Po „zjedení" lístočka sa na niektorom prázdnom mieste poľa objaví nový lístoček, takže hru možno hrať veľmi dlho, pravda, iba ak sme dostatočne pozorní.

Hra sa končí stretnutím húsenice s prekážkou, okrajom poľa či

vlastným telom. Po skončení hry sa na obrazovke zjaví zhodnotenie výsledku hry (počet získaných bodov a doterajší rekordný výsledok) a počítač ponúkne možnosť zopakovania hry.

Všimnite si, že program skloňuje slovo BOD (úsek programu v riadkoch 1500 až 1550). Na zobrazovanie húsenice a jej pohybu používame príkazy BMOVE a BPLOT.

2.3 POČÍTAČ – POMOCNÍK V PRÁCI

Hry nám pomáhajú oboznámiť sa s počítačom, bolo by však nesprávne pokladať osobné počítače za hracie automaty. Počítač má predovšetkým uľahčovať náš život, pomáhať nám v práci. S využitím počítača možno riadiť výrobu, automatizovať projektovacie práce, lepšie rozhodovať, riešiť náročné výpočtové úlohy a rýchlo získavať potrebné informácie. V školách je počítač pomocníkom učiteľa pri výučbe a overovaní vedomostí.

Uvedieme na záver 5 programov, ktoré môžete prakticky využívať doma aj v škole.

2.3.1 KVADRATICKÉ ROVNICE

Počítače sa od svojho vzniku využívajú na riešenie matematických úloh. Ukážeme si, ako počítač využiť na riešenie kvadratických rovníc

$ax^2 + bx + c = 0$

PROGRAM 2.26

Vo výpise 2.26 uvádzame program na riešenie kvadratickej rovnice. Program je zapísaný v Atari basicu a rieši kvadratické rovnice iba v obore reálnych čísel.

Program ľahko prepíšeme do basicu G: stačí iba nahradiť príkaz

2Ø PRINT CHR$(125);

príkazom

2Ø GCLEAR

a malé písmená v komentároch a zobrazovaných textoch nahradiť veľkými.

```
10 REM Prosram 2.26 - KVADRATICKA ROVNICA, 7.5.88
20 REM Atari basic
30 PRINT CHR$(125);
40 PRINT "@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@" P
   RINT
```

```
50 PRINT "   RIESENIE KVADRATICKEJ ROVNICE":PRINT
60 PRINT "        a*x^2 + b*X + c = 0":PRINT
70 PRINT "@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@"
80 PRINT
90 PRINT "Vloz koeficient      a = ";:INPUT A:IF A=
   0 THEN PRINT "To je linearna rovnica":GOTO 90
100 PRINT "Vloz koeficient      b = ";:INPUT B
110 PRINT "Vloz absolutny clen c = ";:INPUT C
120 PRINT
130 PRINT "Rovnica:"
140 PRINT
150 PRINT "          ";STR$(A);"*x^2 ",
160 ON SGN(B)+2 GOTO 190,210,170
170 PRINT "+ ";
180 GOTO 200
190 PRINT "- ";
200 PRINT STR$(ABS(B));"*x ",
210 ON SGN(C)+2 GOTO 240,260,220
220 PRINT "+ ",
230 GOTO 250
240 PRINT "- ";
250 PRINT STR$(ABS(C));
260 PRINT "  =  0"
270 PRINT
280 D=B^2-4*A*C
290 PRINT "Diskriminant = ",D
300 ON SGN(D)+2 GOTO 310,330,360
310 PRINT :PRINT "Rovnica nema riesenie v obore":P
    RINT "realnych cisel!"
320 GOTO 420
330 PRINT :PRINT "Rovnica ma jeden dvojnasobny kor
    en":PRINT
340 Y=-(B+D)/(2*A)
350 GOTO 410
360 PRINT :PRINT "Rovnica ma dva korene":PRINT
370 D=SQR(D)
380 X=(-B+D)/(2*A)
390 Y=(-B-D)/(2*A)
400 PRINT "          X1 = ";X
410 PRINT "          X2 = ";Y
420 PRINT
430 PRINT "Koniec vypoctu"
440 END
```

2.3.2 MALÝ POČTÁR

Program MALÝ POČTÁR môžete pripraviť pre vašich mladších súrodencov, prvákov a druhákov. Pomôže im hravou formou zvládnuť úskalia aritmetiky. Program vytvára skupiny príkladov na sčítanie, odčítanie, násobenie a delenie v obore prirodzených čísel do 100.

PROGRAM 2.27

Výpis 2.27 obsahuje program na precvičovanie aritmetických operácií v jazyku basic G. Program funguje takto:

Po odštartovaní programu sa zobrazí ponuka jednotlivých typov príkladov (pozri obr. 2.53). Vložením príslušného poradového čísla si zvolíte typ príkladov. Na obrazovke sa zjaví rámček, do ktorého

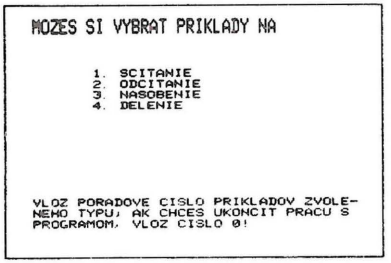
počítač začne zapisovať príklady (napr. pri vložení čísla 1 ide o príklady na sčítanie, pozri obr. 2.54). Každý príklad sa zobrazí bez výsledku (namiesto výsledku je na obrazovke otáznik). Počítač čaká na vloženie výsledku, čo oznámi zvukovým signálom. Po vložení výsledku a stlačení klávesu EOL počítač zhodnotí odpoveď: ak je výsledok správny, zobrazí ho na obrazovke a pripočíta vám jeden bod, pri nesprávnom výsledku zostáva na obrazovke otáznik a počet bodov sa nemení. Zobrazí sa ďalší príklad atď.

Počítač vytvára vždy sériu 10 príkladov a po skončení série zhodnotí výsledky (pozri obr. 2.55). Stlačením klávesu EOL sa znova

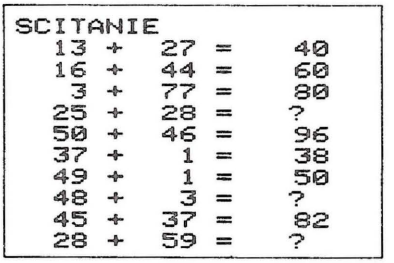
```
10 REM PROGRAM 2.27 - MALY POCTAR, 1. 9. 87
20 REM BASIC G (PMD 85, DIDAKTIK ALFA)
40 SCALE 0,255,0,242
50 M$="" : FOR I=1 TO 48 : M$=M$+" " : NEXT I
100 GOSUB 2600
130 GOSUB 500
140 IF L=0 THEN GOTO 800
150 ON L GOSUB 1000,1100,1200,1300
200 GOTO 130
500 REM PONUKA PRIKLADOV
505 GCLEAR
510 S1=2 : R1=2 : D=250 : S=235 : GOSUB 2400
520 MOVE 20,210
530 LABEL 1,2;"MOZES SI VYBRAT PRIKLADY NA"
540 MOVE 60,170
550 LABEL 1,1;"1. SCITANIE"
560 MOVE 60,160
570 LABEL 1,1;"2. ODCITANIE"
580 MOVE 60,150
590 LABEL 1,1;"3. NASOBENIE"
600 MOVE 60,140
610 LABEL 1,1;"4. DELENIE"
620 MOVE 20,50
630 LABEL 1,1;"VLOZ PORADOVE CISLO PRIKLADOV ZVOLE-"
640 MOVE 20,40
650 LABEL 1,1;"NEHO TYPU! AK CHCES UKONCIT PRACU 8"
660 MOVE 20,30
670 LABEL 1,1;"PROGRAMOM, VLOZ CISLO 0!"
680 K$=CHR$(USR(-31583)) : L=VAL(K$)
690 IF L<0 OR L>4 THEN 680
700 RETURN
800 REM KONIEC PROGRAMU
810 GCLEAR
820 MOVE 10,220
830 LABEL 1,2;"KONIEC PROGRAMU"
840 MOVE 20,140
850 LABEL 3,6;"MALY POCTAR"
860 MOVE 50,30
870 LABEL 2,2;"DO VIDENIA"
880 END
1000 REM SCITANIE
1010 GOSUB 1500
1020 FOR I=1 TO 10
1030 A=INT(50*RND(1)+1)
1040 B=INT((100-A)*RND(1)+1)
1050 C=A+B
1060 GOSUB 1600
1070 NEXT I
1080 GOSUB 1700
1090 RETURN
1100 REM ODCITANIE
1110 GOSUB 1500
```

```
1120 FOR I=1 TO 10
1130 C=INT(50*RND(1)+1)
1140 B=INT((100-C)*RND(1)+1)
1150 A=C+B
1160 GOSUB 1600
1170 NEXT I
1180 GOSUB 1700
1190 RETURN
1200 REM NASOBENIE
1210 GOSUB 1500
1220 FOR I=1 TO 10
1230 A=INT(10*RND(1)+1)
1240 B=INT(10*RND(1)+1)
1250 C=A*B
1260 GOSUB 1600
1270 NEXT I
1280 GOSUB 1700
1290 RETURN
1300 REM DELENIE
1310 GOSUB 1500
1320 FOR I=1 TO 10
1330 B=INT(10*RND(1)+1)
1340 C=INT(10*RND(1)+1)
1350 A=C*B
1360 GOSUB 1600
1370 NEXT I
1380 GOSUB 1700
1390 RETURN
1500 REM PRIPRAVA TABULKY PRIKLADOV
1510 H=0 : GCLEAR
1520 S1=2 : R1=2 : D=250 : S=235 : GOSUB 2400
1530 IF L=1 THEN W$="SCITANIE"
1540 IF L=2 THEN W$="ODCITANIE"
1550 IF L=3 THEN W$="NASOBENIE"
1560 IF L=4 THEN W$="DELENIE"
1570 MOVE 10,210
1580 LABEL 2,2;W$
1590 RETURN
1600 REM ZOBRAZENIE PRIKLADOV
1610 GOSUB 2000
1620 MOVE 10,210-I*20
1630 LABEL 2,2;P$
1640 MOVE 190,210-I*20
1650 LABEL 2,2;"?"
1660 DISP "VLOZ VYSLEDOK: ";A$;Z$;B$;" = ?"
1670 INPUT X
1680 GOSUB 2200
1690 RETURN
1700 REM CELKOVE ZHODNOTENIE VYSLEDKOV
1710 GCLEAR
1720 S1=2 : R1=2 : D=250 : S=235 : GOSUB 2400
1730 MOVE 10,210
1740 IF H=10 THEN LABEL 1,2;"VYBORNE"
1750 IF H>7 AND H<10 THEN LABEL 1,2;"MOHLO BY TO BYT LEPSIE"
1760 IF H>4 AND H<8 THEN LABEL 1,2;"PRILIS SI SA NEVYZNAMENAL"
1770 IF H>1 AND H<5 THEN LABEL 1,2;"NEVIES POCITAT"
1780 IF H>=0 AND H<2 THEN LABEL 1,2;"ZE SA NEHANBIS"
1790 MOVE 10,190
1800 LABEL 1,1;"ZA ";W$;" SI ZISKAL"
1810 MOVE 100,100
1820 LABEL 5,9;H
1830 MOVE 10,60
1840 GOSUB 2500
1850 LABEL 1,1;Q$;" Z 10 MOZNYCH"
1860 DISP "                    STLAC KLAVES EOL!"
1870 INPUT Q0$
1880 RETURN
2000 REM UPRAVA PRIKLADU
2010 ON L GOTO 2020,2030,2040,2050
2020 Z$=" +"
```

```
2025 GOTO 2060
2030 Z$=" -"
2035 GOTO 2060
2040 Z$=" X"
2045 GOTO 2060
2050 Z$=" :"
2060 N=A : GOSUB 2100 : A$=N$
2070 N=B : GOSUB 2100 : B$=N$
2080 N=C : GOSUB 2100 : C$=N$
2090 P$=A$+Z$+B$+" ="
2095 RETURN
2100 REM UPRAVA CISLA
2110 NS=INT(N/100) : ND=INT((N-NS*100)/10)
2120 NJ=INT(N-NS*100-ND*10)
2130 IF NS>0 THEN N$=STR$(N)
2140 IF NS=0 AND ND>0 THEN N$=" "+STR$(N)
2150 IF NS=0 AND ND=0 THEN N$="  "+STR$(N)
2160 RETURN
2200 REM HODNOTENIE PRIKLADU
2210 IF X<>C THEN 2300
2220 MOVE 190,210-I*20
2230 MOVE 190,210-I*20
2240 LABEL 2,2;"?"
2250 MOVE 170,210-I*20
2260 LABEL 2,2;C$
2270 H=H+1
2280 DISP "SPRAVNE, ZISKAVAS 1 BOD"
2285 PAUSE 10 : DISP M$
2290 GOTO 2310
2300 DISP "ZLE - 0 BODOV " : PAUSE 10 : DISP M$
2310 GOSUB 2500   .
2320 DISP "DOTERAZ SI ZISKAL ";STR$(H);" ";Q$
2330 PAUSE 10 : DISP M$
2340 RETURN
2400 REM RAMCEK
2410 MOVE S1,R1
2420 PLOT S1+D,R1
2430 PLOT S1+D,R1+S
2440 PLOT S1,R1+S
2450 PLOT S1,R1
2460 RETURN
2500 REM SKLONOVANIE
2510 T$="BOD"
2520 IF H=1 THEN Q$=T$
2530 IF H>1 AND H<5 THEN Q$=T$+"Y"
2540 IF H>4 OR H=0 THEN Q$=T$+"OV"
2550 RETURN
2600 REM HLAVICKA PROGRAMU
2610 GCLEAR
2620 MOVE 80,200
2630 LABEL 2,2;"PROGRAM"
2680 MOVE 20,140
2690 LABEL 3,6;"MALY POCTAR"
2700 MOVE 10,120
2710 LABEL 1,1;"PROGRAM UMOZNUJE PRECVICIT SI SCITANIE,"
2720 MOVE 10,110
2730 LABEL 1,1;"ODCITANIE, NASOBENIE A DELENIE V OBORE"
2740 MOVE 10,100
2750 LABEL 1,1;"PRIRODZENYCH CISEL DO 100. PREDKLADA"
2760 MOVE 10,90
2770 LABEL 1,1;"VZDY 10 PRIKLADOV ZVOLENEHO TYPU A ZA"
2780 MOVE 10,80
2790 LABEL 1,1;"SPRAVNE VYRIESENY PRIKLAD PRIDELUJE"
2800 MOVE 10,70
2810 LABEL 1,1;"JEDEN BOD."
2820 MOVE 70,20
2830 LABEL 1,2;"VELA USPECHOV!"
2840 DISP "                          STLAC KLAVES EOL!"
2850 INPUT Q0$
2860 RETURN
```



2.53 Ponuka programu MALÝ POČTÁR



2.54 MALÝ POČTÁR učí sčítať

2.55 MALÝ POČTÁR hodnotí výsledky

vrátite k ponuke typov príkladov. Potom môžete buď pokračovať, alebo vložením čísla 0 prácu s programom ukončiť.

Príklady vytvárame pomocou funkcie RND. Niekedy môže byť účelné zmeniť rozsah čísel, z ktorých sa vytvárajú príklady. Ak napríklad chceme, aby sme vytvárali príklady na sčítanie iba z čísel do 20 s tým, že výsledok bude číslo do 40, zmeníme programové riadky 1110 a 1120 takto:

```
111Ø A=INT(2Ø*RND(1)+1)
112Ø B=INT(2Ø*RND(1)+1)
```

2.3.3 MIKROTEST

Svoje znalosti o práci s počítačmi PMD 85 a Didaktik Alfa si môžete overiť pomocou programu MIKROTEST. Program postupne predkladá na obrazovku otázky o jednotlivých prvkoch jazyka basic G. Spolu s každou otázkou ponúka aj niekoľko odpovedí,

z ktorých iba jedna je správna. Vašou úlohou je vložiť poradové číslo správnej odpovede. Za každú správnu odpoveď získate bod.

PROGRAM 2.28

Program z výpisu 2.28 vytvára test z 20 otázok s výberovými odpoveďami. Ku každej otázke ponúka 3 odpovede. Otázku spolu s ponukou odpovedí nazývame testovou položkou (zobrazenie testovej položky pozri na obr. 2.56).

```
10 REM PROGRAM 2.28 - MIKROTEST, 10. 9. 87
20 REM BASIC G (PMD 85, DIDAKTIK ALFA)
30 DIM P$(5),L(5)
40 SCALE 0,200,0,200
100 GOSUB 2500
110 B=0 : RESTORE
120 READ N,M
130 FOR K=1 TO N : GOSUB 1000 : NEXT K
160 GOSUB 1800 : IF A$="A" THEN 110
180 END
1000 REM PREZENTACIA TESTOVYCH POLOZIEK
1010 GCLEAR
1020 GOSUB 1200 : GOSUB 1300 : GOSUB 1400
1100 RETURN
1200 REM VYTVORENIE NAHODNEJ PERMUTACIE
1210 FOR I=1 TO M : L(I)=I : NEXT I : M1=M-1
1230 FOR I=1 TO M1 : J=INT(I+RND(1)*(M-I+1))
1240 K1=L(J) : L(J)=L(I) : L(I)=K1 : NEXT I
1250 FOR I=1 TO M : IF L(I)=1 THEN 1270
1260 NEXT I
1270 S=I
1280 RETURN
1300 REM PODPROGRAM NACITANIA POLOZKY
1310 FOR I=0 TO M
1320 READ P$(I) : NEXT I : RETURN
1400 REM PODPROGRAM ZOBRAZENIA POLOZKY
1410 MOVE 1,180
1420 LABEL 1,1;"PROGRAM  M I K R O T E S T   POLOZKA ";STR$(K)"
1430 MOVE  1,170
1440 LABEL 1,1;"DOTERAZ ZISKANE BODY:";STR$(B)
1450 MOVE 1,150
1460 LABEL 1,1;P$(0)
1470 FOR I=1 TO M
1480 II=L(I)
1490 MOVE 20,140-I*10
1500 LABEL 1,1;STR$(I);")    ";P$(II)
1510 NEXT I
1520 MOVE 1,50
1530 LABEL 1,1;"VLOZ PORADOVE CISLO SPRAVNEJ ODPOVEDI:"
1540 O=USR(-31583)-48 : IF O)0 AND O(M+1 THEN 1545
1543 BEEP : GOTO 1540
1545 LABEL 1,1;O
1550 IF O=S THEN 1570
1560 T$="NESPRAVNE" : GOTO 1580
1570 T$="SPRAVNE" : B=B+1
1580 MOVE 130,10
1590 LABEL 1,2;T$
1600 PAUSE 20
1610 RETURN
1800 REM ZHODNOTENIE TESTU
1810 GCLEAR
1820 B0=B/N : D$=" " : IF B0(0.9 AND B0)0 THEN D$="IBA"
1830 MOVE 10,180
```

```
1840 IF B0=1 THEN T$="BLAHOZELAM, ANI RAZ SI SA NEPOMYLIL"
1850 IF B0<1 AND B0>=0.9 THEN T$="TVOJE VEDOMOSTI SU VELMI DOBRE"
1860 IF B0<0.9 AND B0>=0.5 THEN T$="MAS ESTE ZNACNE MEDZERY"
1870 IF B0<0.5 AND B0>=0.2 THEN T$="ZOBER SI PRIRUCKY A ZACNI STUDOVAT"
1880 IF B0<0.2 THEN T$="S TAKYMI VEDOMOSTAMI ZA MNOU NECHOD"
1890 LABEL 1,2;T$
1900 MOVE 10,160
1910 LABEL 1,1;"ZISKAL SI ";D$
1920 MOVE 80,90
1930 LABEL 5,9;STR$(B)
1940 MOVE 10,70
1950 GOSUB 2100
1960 LABEL 1,1;W$;" Z ";STR$(N);" MOZNYCH!"
1970 MOVE 10,50
1980 LABEL 1,1;"CHCES SI TEST ZOPAKOVAT? (A,N): "
1990 A$=CHR$(USR(-31583)) : IF A$="A"THEN 2020
2000 IF A$="N" THEN 2050
2010 BEEP : GOTO 1990
2020 LABEL 1,1;A$
2025 MOVE 50,20
2030 LABEL 1,2;"PRIPRAV SA, ZACINAME ZNOVA!"
2040 PAUSE 20 : GOTO 2080
2050 LABEL 1,1;A$"
2060 MOVE 50,20
2070 LABEL 1,2;"KONIEC PROGRAMU - DO VIDENIA!"
2080 RETURN
2100 REM PODPROGRAM SKLONOVANIA
2110 T$="BOD"
2120 IF B=1 THEN W$=T$
2130 IF B>1 AND B<5 THEN W$=T$+"Y"
2140 IF B>4 OR B=0 THEN W$=T$+"OV"
2150 RETURN
2500 REM HLAVICKA PROGRAMU
2510 GCLEAR
2520 MOVE 60,180
2530 LABEL 2,2;"PROGRAM"
2580 MOVE 10,100
2590 LABEL 2,6;"M I K R O T E S T"
2600 MOVE 5,80
2610 LABEL 1,1;"PROGRAM POSTUPNE PREDKLADA NA OBRAZOVKE"
2620 MOVE 5,70
2630 LABEL 1,1;"OTAZKY O PRVKOCH PRACE S BASICOM. SPOLU S"
2640 MOVE 5,60
2650 LABEL 1,1;"KAZDOU OTAZKOU PONUKA AJ NIEKOLKO ODPOVE-"
2660 MOVE 5,50
2670 LABEL 1,1;"DI, Z KTORYCH IBA JEDNA JE SPRAVNA. TREBA"
2680 MOVE 5,40
2690 LABEL 1,1;"VLOZIT PORADOVE CISLO SPRAVNEJ  ODPOVEDI. "
2700 MOVE 5,30
2710 LABEL 1,1;"ZA KAZDU SPRAVNU ODPOVED ZISKATE BOD."
2720 MOVE 60,10
2730 LABEL 1,2;"VELA USPECHOV!"
2740 DISP "                              STLAC EOL!"
2750 INPUT Q0$
2760 RETURN
5000 REM UDAJE TESTU
5010 DATA 20,3
5020 DATA "PROGRAM ZAPISEME NA KAZETU POVELOM:"
5030 DATA  "SAVE","LOAD","PRINT"
5040 DATA "ZAZNAM PROGRAMU NA KAZETE OVERIME POVELOM:"
5050 DATA "CHECK","CONTROL","TEST"
5060 DATA "PROGRAM PRECITAME Z KAZETY POVELOM:"
5070 DATA "LOAD","INPUT","READ"
5080 DATA "PROGRAM V PAMATI ODSTARTUJEME POVELOM:"
5090 DATA "RUN","NEW","GO"
5100 DATA "PROGRAM VYPISEME NA OBRAZOVKU POVELOM:"
5110 DATA "LIST","PRINT","WRITE"
5120 DATA "RIADKY AUTOMATICKY CISLUJEME POVELOM:"
5130 DATA "AUTO","INI","NEW"
5140 DATA "PROGRAM Z PAMATI VYMAZEME POVELOM:"
```

```
5150 DATA "NEW","DELETE","CONT"
5160 DATA "VYKONAVANIE PROGRAMU PRERUSIME KLAVESOM:"
5170 DATA "STOP","END","CLR"
5180 DATA "PRERUSENY PROGRAM OBNOVIME PRIKAZOM:"
5190 DATA "CONT","NEW","START"
5200 DATA "RIADOK PROGRAMU VYPISEME PRIKAZOM:"
5210 DATA "LLIST","DISP","PRINT"
5220 DATA "PRVYM ZNAKOM NAZVU PREMENNEJ V BASICU JE:"
5230 DATA "VZDY PISMENO","PISMENO ALEBO CISLICA","AKYKOLVEK ZNAK"
5240 DATA "OBRAZOVKU VYMAZEME PRIKAZOM:"
5250 DATA "GCLEAR","CLEAR","NEW"
5260 DATA "NA POZNAMKY V PROGRAME POUZIVAME PRIKAZ:"
5270 DATA "REM","LIST","DISP"
5280 DATA "HODNOTU PREMENNEJ A VYPISEME PRIKAZOM:"
5290 DATA "PRINT A","INPUT A","WRITE A"
5300 DATA "DO DIALOGOVEHO RIADKU VYPISUJEME PRIKAZOM:"
5310 DATA "DISP","PRINT","LET"
5320 DATA "ZVUKOVY SIGNAL PROGRAMUJEME PRIKAZOM:"
5330 DATA "BEEP","SOUND","PEEP"
5340 DATA "UDAJE Z KLAVESNICE CITAME PRIKAZOM"
5350 DATA "INPUT","READ","PRINT"
5360 DATA "HODNOTY RIADIACEJ PREMENNEJ CYKLU FOR-NEXT:"
5370 DATA "MOZU BYT LUBOVOLNE","MUSIA BYT KLADNE","MUSIA BYT CELOCISELNE"
5380 DATA "DLZKA KROKU CYKLU FOR-NEXT-STEP:"
5390 DATA "MOZE BYT LUBOVOLNA","MUSI BYT CELOCISELNA","MUSI BYT KLADNA"
5400 DATA "PRIKAZOM PODMIENENEHO SKOKU JE PRIKAZ:"
5410 DATA "IF A$=B$ THEN 120","GOTO 120","LET X=120"
```

Texty testových položiek (otázok a ponúkaných odpovedí) sú uložené na konci programu v príkazoch DATA. V prvom príkaze DATA (riadok 5010) sú uvedené dve čísla: prvé z nich udáva počet položiek konkrétneho testu (v našom prípade 20), druhé počet ponúkaných odpovedí v každej položke (v našom prípade 3). Ďalej nasledujú jednotlivé položky: text otázky a texty troch ponúkaných odpovedí. Program je dostatočne všeobecný – test môžeme ľahko rozšíriť pripojením ďalších položiek a zväčšením počtu ponúkaných odpovedí v každej položke (treba však príslušným spôsobom pozmeniť údaje o počte položiek a ponúkaných odpovedí v riadku 5010). Program možno použiť aj na overovanie vedomostí z iných oblastí zmenou skupiny textov od príkazového riadka 5020. Program predpokladá, že dĺžka otázky nie je väčšia ako 46 znakov a dĺžka každej

```
PROGRAM  M I K R O T E S T    POLOZKA  11
DOTERAZ ZISKANE BODY: 10

PRVYM ZNAKOM NAZVU PREMENNEJ V BASICU JE:

    1)   AKYKOLVEK ZNAK
    2)   PISMENO ALEBO CISLICA
    3)   VZDY PISMENO



VLOZ PORADOVE CISLO SPRAVNEJ ODPOVEDI: 3

                          SPRAVNE
```

2.56 Testová položka v programe MIKROTEST

z ponúkaných odpovedí neprekročí 40 znakov, ľahko však môžeme program upraviť tak, aby pracoval aj s dlhšími textmi. Pri zachovaní uvedenej dĺžky textov možno v každej položke ponúkať až 8 výberových odpovedí.

Aby test nebol príliš jednoduchý a aby nebolo možné mechanicky si zapamätať poradie správnych odpovedí, toto poradie sa náhodne mení. Pri vkladaní textov ponúkaných odpovedí treba iba zabezpečiť, aby správna odpoveď bola uvedená ako prvá. O vhodné „zamiešanie" poradia a správne zhodnotenie odpovedí sa už postará počítač.

2.3.4 PORADCA

Pôvodne sa počítače rozvíjali ako výpočtové automaty – stroje na rýchle vykonávanie zložitých a rozsiahlych výpočtov. Vtedy sa aj vžil doteraz používaný názov. Počítače síce aj dnes počítajú, ale presnejšie by azda bolo nazývať ich „informačnými automatmi". Ich najdôležitejšou úlohou v súčasnosti je uchovávať, spracúvať a poskytovať informácie najrôznejšieho druhu.

Informačný charakter má aj program PORADCA, ktorý slúži ako pomocník začiatočníka pri práci s počítačmi PMD 85 a Didaktik Alfa.

PROGRAM 2.29

Program PORADCA (pozri výpis 2.29) je určený pre počítače PMD 85 a Didaktik Alfa a umožňuje v prípade potreby privolať si a zobraziť informácie o spôsobe použitia jednotlivých povelov interpreta jazyka basic G.

```
5000 REM PROGRAM 2.29 - PORADCA, 10. 9. 87
5010 REM BASIC G (PMD 85, DIDAKTIK ALFA)
5020 SCALE 0,255,0,242
5030 GOSUB 5500
5040 GOSUB 5200
5050 IF K=0 THEN GOTO 5400
5060 GOSUB 5800
5070 GOTO 5040
5200 REM PONUKA
5210 GCLEAR
5220 MOVE 20,220
5230 LABEL 1,2;"PONUKA POSKYTOVANYCH INFORMACII:"
5240 RESTORE 6000
5250 I=0
5260 READ T$
5270 IF LEFT$(T$,1)="#" THEN 5320
5280 I=I+1
```

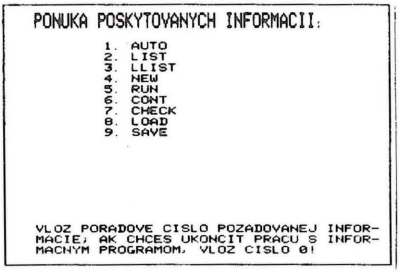
```
5290 MOVE 60,210-I*10
5300 LABEL 1,1;STR$(I);". ";T$
5310 GOTO 5260
5320 MOVE 20,30
5330 LABEL 1,1;"VLOZ PORADOVE CISLO POZADOVANEJ INFOR-"
5340 MOVE 20,20
5350 LABEL 1,1;"MACIE; AK CHCES UKONCIT PRACU S INFOR-"
5360 MOVE 20,10
5370 LABEL 1,1;"MACNYM PROGRAMOM, VLOZ CISLO 0!"
5380 K$=CHR$(USR(-31583)) : K=VAL(K$)
5390 RETURN
5400 REM KONIEC PROGRAMU
5410 GCLEAR
5420 MOVE 10,220
5430 LABEL 1,2;"KONIEC PROGRAMU"
5440 MOVE 10,150
5450 LABEL 2,5;"   P O R A D C A"
5460 MOVE 50,30
5470 LABEL 2,2;"DO VIDENIA"
5480 GOTO 5990
5500 REM HLAVICKA PROGRAMU
5510 GCLEAR
5520 MOVE 70,220
5530 LABEL 2,2;"PROGRAM"
5580 MOVE 5,150
5590 LABEL 2,5;"   P O R A D C A"
5600 MOVE 10,130
5610 LABEL 1,1;"PROGRAM POSKYTUJE INFORMACIE O POUZIVANI"
5620 MOVE 10,120
5630 LABEL 1,1;"RIADIACICH PRIKAZOV BASICOVSKEHO INTER-"
5640 MOVE 10,110
5650 LABEL 1,1;"PRETATORA. PROGRAM MOZE BYT V PAMATI PO-"
5660 MOVE 10,100
5670 LABEL 1,1;"CAS PRACE S INYM PROGRAMOM A MOZETE SA"
5680 MOVE 10,90
5690 LABEL 1,1;"K NEMU OBRACAT O RADU V PRIEBEHU VKLA-"
5700 MOVE 10,80
5710 LABEL 1,1;"DANIA, EDITOVANIA A SPUSTANIA PROGRAMU"
5720 MOVE 10,70
5730 LABEL 1,1;"A PRI PRACI S MAGNETOFONOM"
5740 MOVE 60,20
5750 LABEL 1,2;"VELA USPECHOV!"
5760 DISP "                          STLAC KLAVES EOL!"
5770 INPUT Q0$
5780 RETURN
5800 REM ZOBRAZOVANIE INFORMACII
5810 GCLEAR
5820 L=6000+K*200
5830 RESTORE L
5840 READ T$
5850 MOVE 20,220
5860 LABEL 2,2;T$

5870 I=0
5880 READ T$
5890 IF LEFT$(T$,1)="*" THEN 5950
5900 I=I+1
5910 MOVE 30,210-I*10
5920 LABEL 1,1;T$
5930 GOTO 5880
5950 DISP "AK CHCES POKRACOVAT, STLAC EOL!"
5960 INPUT Q0$
5970 RETURN
5990 END
6000 DATA "AUTO","LIST","LLIST","NEW","RUN","CONT","CHECK","LOAD","SAVE","*"
6200 DATA "AUTO"
6210 DATA "TENTO POVEL ZABEZPECUJE AUTOMATICKE"
6220 DATA "CISLOVANIE PROGRAMOVYCH RIADKOV PRI"
6230 DATA "VKLADANI BASICOVSKYCH PROGRAMOV. JEHO"
6240 DATA "POUZITIE BEZ PARAMETROV CISLUJE RIAD-"
6250 DATA "KY OD RIADKU S CISLOM 10 S KROKOM 10"
```

```
6260 DATA "(ALEBO OD POSLEDNEHO VLOZENEHO RIAD-"
6270 DATA "KU, ZVACSENEHO O 10, S KROKOM 10)."
6280 DATA "AK VLOZIME POVEL AUTO V TVARE"
6290 DATA "          AUTO M,N"
6300 DATA "KDE  M,N  SU CELE KLADNE CISLA, POTOM"
6310 DATA "AUTOMATICKE CISLOVANIE SA USKUTOCNUJE"
6320 DATA "OD RIADKU S CISLOM M A S KROKOM N"
6330 DATA "(NAPR.  AUTO 100,50  CISLUJE PROGRAM"
6340 DATA "OD RIADKU S CISLOM 100 S KROKOM 50)."
6350 DATA "POSOBNOST POVELU AUTO ZRUSIME VYMA-"
6360 DATA "ZANIM CISLA RIADKU A VLOZENIM INEHO"
6380 DATA "RIADIACEHO POVELU, NAPR. LIST.","*"
6400 DATA "LIST"
6410 DATA "POVELOM LIST VYPISEME NA OBRAZOVKU"
6420 DATA "BASICOVSKY PROGRAM, KTORY SA NACHADZA"
6430 DATA "V PAMATI MIKROPOCITACA. PROGRAM SA VY-"
6440 DATA "PISUJE ROLOVANIM ZDOLA NAHOR OD PRVE-"
6450 DATA "HO PROGRAMOVEHO RIADKU. AK POUZIJEME"
6460 DATA "POVEL LIST V TVARE"
6470 DATA "          LIST CR"
6480 DATA "KDE  CR  JE CISLO RIADKU, PROGRAM SA"
6490 DATA "VYPISUJE OD PROGRAMOVEHO RIADKU S PO-"
6500 DATA "ZADOVANYM CISLOM (NAPR.  LIST 100  VY-"
6510 DATA "PISUJE PROGRAM OD RIADKU S CISLOM 100)."
6520 DATA "VYPIS PROGRAMU MOZEME ZASTAVIT STLACE-"
6530 DATA "NIM KLAVESU  STOP  A ZNOVU HO OBNOVIT"
6540 DATA "VLOZENIM POVELU  LIST CR  OD RIADKU,"
6550 DATA "V KTOROM SME HO PRERUSILI.","*"
6600 DATA "LLIST"
6610 DATA "POVELOM "
6620 DATA "          LLIST CR"
6630 DATA "KDE  CR  JE CISLO PROGRAMOVEHO RIADKU,"
6640 DATA "VYPISEME DO DIALOGOVEHO RIADKU BASI-"
6650 DATA "COVSKY PROGRAMOVY RIADOK S CISLOM CR."
6660 DATA "TENTO RIADOK MOZEME UPRAVIT (ZEDI-"
6670 DATA "TOVAT) POSUVANIM KURZORA KLAVESAMI"
6680 DATA "'SIPKA VLAVO', 'SIPKA VPRAVO' A VYUZI-"
6690 DATA "TIM EDITOVACICH KLAVESOV INS, DEL."
6700 DATA "PO UKONCENI UPRAV ODOSLEME ZEDITOVANY"
6710 DATA "RIADOK KLAVESOM EOL.","*"
6800 DATA "NEW"
6810 DATA "TENTO POVEL ZRUSI PROGRAM, NACHADZA-"
6820 DATA "JUCI SA V PAMATI, A ZAROVEN VYNULUJE"
6830 DATA "VSETKY PREMENNE. POUZIVA SA PRED ZACA-"
6840 DATA "TIM VKLADANIA NOVEHO PROGRAMU.","*"
7000 DATA "RUN"
7010 DATA "TYMTO POVELOM ODSTARTUJEME PROGRAM,"
7015 DATA "NACHADZAJUCI SA V PAMATI, OD PROGRAMO-"
7020 DATA "VEHO RIADKU S NAJNIZSIM CISLOM. AK"
7030 DATA "TENTO POVEL POUZIJEME V TVARE"
7040 DATA "          RUN CR"
7050 DATA "KDE  CR  JE CISLO RIADKU, PROGRAM SA"
7060 DATA "ODSTARTUJE OD ZADANEHO RIADKU. PRED "
7070 DATA "ODSTARTOVANIM PROGRAMU POVELOM RUN"
7080 DATA "SA AUTOMATICKY VYNULUJU VSETKY PRE-"
7090 DATA "MENNE A UKAZOVATELE DAT.","*"
7200 DATA "CONT"
7210 DATA "VYKONAVANIE PROGRAMU, ODSTARTOVANEHO"
7220 DATA "POVELOM RUN, MOZEME PRERUSIT STLACE-"
7230 DATA "NIM KLAVESU STOP. PO TAKOMTO PRERUSE-"
7240 DATA "NI OBNOVIME VYKONAVANIE PROGRAMU VLO-"
7250 DATA "ZENIM POVELU CONT. PROGRAM POKRACUJE"
7260 DATA "OD RIADKU, V KTOROM DOSLO K PRERUSENI.","*"
7400 DATA "CHECK"
7410 DATA "POVEL SA POUZIVA V TVARE"
7420 DATA "          CHECK CZ"
7430 DATA "KDE  CZ  JE CISLO ZAZNAMU (PROGRAMO-"
7440 DATA "VEHO SUBORU) NA KAZETE MAGNETOFONU."
7450 DATA "PO VLOZENI A ODOSLANI TOHTO POVELU"
7460 DATA "TREBA SPUSTIT MAGNETOFON NA PREHRAVA-"
7470 DATA "NIE. POVEL KONTROLUJE SPRAVNOST ZAZ-"
```

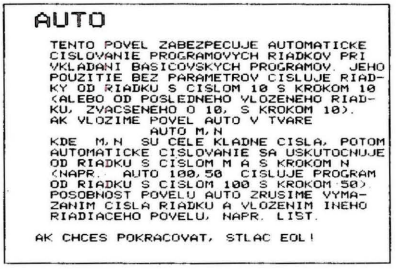
```
7480 DATA "NAMU NA KAZETE A ZAROVEN NASTAVI KAZE-"
7490 DATA "TU ZA PRISLUSNY SUBOR. V PRIPADE BEZ-"
7500 DATA "CHYBNEHO ZAZNAMU SA V DIALOGOVOM RIAD-"
7510 DATA "KU OBRAZOVKY VYPISE SPRAVA"
7520 DATA "            OK"
7530 DATA "V OPACNOM PRIPADE (ZAZNAM JE CHYBNY)"
7540 DATA "SA VYPISE SPRAVA"
7550 DATA "            FILE ERROR"
7560 DATA "(CHYBA SUBORU).","*"
7600 DATA "LOAD"
7610 DATA "TYMTO POVELOM PREPISEME BASICOVSKY"
7620 DATA "PROGRAM Z KAZETY MAGNETOFONU DO OPE-"
7630 DATA "RACNEJ PAMATI POCITACA. POUZIVAME HO"
7640 DATA "V TVARE"
7650 DATA "            LOAD CZ"
7660 DATA "KDE  CZ  JE CISLO ZAZNAMU (PROGRAMOVE-"
7670 DATA "HO SUBORU) NA KAZETE. PRV, AKO POUZI-"
7680 DATA "JEME TENTO POVEL, NASTAVIME KAZETU "
7690 DATA "PRED PRISLUSNY ZAZNAM (NAPRIKLAD PRI-"
7700 DATA "KAZOM CHECK). PO ODOSLANI POVELU ZAP-"
7710 DATA "NEME MAGNETOFON NA PREHRAVANIE. PO"
7720 DATA "USPESNOM NACITANI PROGRAMU DO PAMATI"
7730 DATA "SA V DIALOGOVOM RIADKU VYPISE SPRAVA"
7740 DATA "            OK"
7750 DATA "AK DOSLO PRI CITANI K CHYBE, VYPISE"
7760 DATA "SA SPRAVA"
7770 DATA "            FILE ERROR"
7780 DATA "A TREBA SA ZNOVA POKUSIT O JEHO PRECI-"
7790 DATA "TANIE.","*"
7800 DATA "SAVE"
7810 DATA "TYMTO POVELOM ZAPISEME BASICOVSKY"
7820 DATA "PROGRAM Z OPERACNEJ PAMATI NA KAZETU"
7830 DATA "MAGNETOFONU. POUZIVAME HO V TVARE"
7840 DATA "            SAVE CZ NAZOV"
7850 DATA "KDE  CZ  JE CISLO ZAZNAMU, POD KTORYM"
7860 DATA "SA MA PROGRAMOVY SUBOR ZAPISAT NA KA-"
7870 DATA "ZETE, A NAZOV JE MAXIMALNE 8-ZNAKOVY"
7880 DATA "NAZOV PROGRAMU (KOMENTAR), PRED KTORY"
7890 DATA "SA ZAPISUJE UVODZOVKA. PRV, AKO ODO-"
7900 DATA "SLEME POVEL SAVE, MUSIME NASTAVIT KA-"
7910 DATA "ZETU ZA KONIEC POSLEDNEHO SUBORU, A"
7920 DATA "ZAPNUT MAGNETOFON NA NAHRAVANIE. PO"
7930 DATA "USPESNOM ZAPISE PROGRAMU NA KAZETU SA"
7940 DATA "V DIALOGOVOM RIADKU OBJAVI SPRAVA"
7950 DATA "            OK"
7960 DATA "CISLA SUBOROV NA KAZETE MOZU BYT V IN-"
7970 DATA "TERVALE 0 - 63.","*"
```

Po odštartovaní programu sa na obrazovke zjaví ponuka poskytovaných informácií (pozri obr. 2.57). Ponuka obsahuje zoznam povelov. Vložením poradového čísla požadovaného povelu (podľa jeho poradia v zozname ponuky) si zabezpečíme zobrazenie informácie o práci s týmto povelom. Ak napríklad vložíme číslo 1, zobrazí nám počítač popis práce s povelom AUTO (pozri obr. 2.58), ak vložíme číslo 2, získame popis práce s povelom LIST atď. Po prezretí príslušných informácií a stlačení klávesu EOL sa vrátime znovu do ponuky a môžeme buď privolať informáciu o ďalšom povele, alebo číslom 0 ukončiť prácu s programom.

Všimnime si, že program sa začína riadkom s číslom 5000. Takýto začiatok číslovania riadkov sme zvolili zámerne. Program totiž možno používať ako „poradcu" pri práci s inými programami (pri vklada-



2.57 Ponuka programu PORADCA



2.58 Informačný výstup programu PORADCA

ní programov, editovaní programových riadkov, spúšťaní programov, kontrolovaní záznamov na kazete a pod.). Keď program PORADCA vložíme do pamäte (napr. načítame ho z kazety magnetofónu), zostáva ešte v pamäti dostatok miesta na iné programy. Ďalší program, ktorý chceme vložiť z klávesnice, vkladáme s číslami riadkov menšími ako 5000 a ukončíme ho príkazom END. Na novovkladaný program, ktorý štartujeme jednoducho povelom RUN, nemá program PORADCA vplyv. Kedykoľvek sa však chceme obrátiť na počítač o pomoc a spýtať sa na jednotlivé riadiace povely interpreta, môžeme odštartovať program PORADCA povelom

RUN 5000

a vyžiadať si potrebnú informáciu.

Vlastný program PORADCA je pomerne krátky, končí sa riadkom 5390. Všetky ďalšie riadky obsahujú údaje zapísané v príkazoch DATA. Zmenou týchto údajov program ľahko upravíme na iné informačné účely. Môžeme napr. vložiť údaje o spôsobe používania jednotlivých príkazov jazyka basic, o jednotlivých krajinách, s ktorými sa oboznamujeme na hodinách zemepisu, o význačných historických udalostiach, o slávnych vedcoch, zaujímavých knihách, planétach slnečnej sústavy alebo o jednotlivých žiakoch v triede.

V riadku 6000 je zoznam názvov údajov, ktorý sa zobrazuje v ponuke. Môže zaberať aj viac riadkov, pričom posledným údajom zoznamu musí byť hviezdička. Jednotlivé skupiny údajov (informačné texty) sa vkladajú od riadka 6200 (prvá skupina), 6400 (druhá skupina), 6600 (tretia skupina) atď., teda prvý riadok s údajmi nasledujúcej skupiny má číslo o 200 vyššie ako prvý riadok s údajmi predchádzajúcej skupiny.

Skupina údajov sa začína názvom zhodným s názvom v zozname

ponuky a končí sa hviezdičkou. Dĺžka textov v jednotlivých údajových riadkoch (v úvodzovkách príkazov DATA) by nemala byť väčšia ako 38 znakov, aby sa príslušné riadky dali v naprogramovanej grafickej úprave celé zobraziť na obrazovke. Rozsah jednej skupiny údajov by nemal byť väčší ako 20 riadkov. Pokúste sa zmenou údajov programu PORADCA vytvoriť si vlastný informačný program.

2.3.5 ADRESÁR

Rozlúčime sa užitočným programom, ktorý vám umožní založiť si, doplňať a využívať na počítačoch Atari adresár. Súbor údajov s adresárom sa uchováva na kazete. V prípade, že máte k dispozícii aj tlačiareň Atari 1029, program vám umožní vytlačiť si celý adresár alebo jeho časť. Je však použiteľný aj bez tlačiarne – údaje adresára v tomto prípade zobrazuje na obrazovke.

PROGRAM 2.30

Program z výpisu 2.30 je napísaný v Atari basicu. Po jeho odštartovaní sa zobrazí ponuka týchto činností:

1. Načítanie adries z kazety.
2. Zapísanie adries na kazetu.
3. Vloženie novej adresy.
4. Zmena adresy.
5. Vymazanie adresy.
6. Vyhľadanie adresy.
7. Ukončenie práce s programom.

Keď s programom pracujeme prvý raz, začneme vkladaním adries, zvolíme teda činnosť 3. Každý záznam v adresári má túto štruktúru:

Označenie (napr. titul),
meno,
priezvisko,
ulica,
miesto a PSČ,
telefón,
poznámka.

Skladá sa teda zo 7 položiek, pričom na každú z týchto položiek je rezervovaných 20 znakov. Pri vkladaní novej adresy sa zobrazí názov každej položky a počítač očakáva vloženie príslušného textu, ktoré ukončíme stlačením klávesu Return.

Po vložení prvých adries zapíšeme súbor adries na kazetu (v ponuke teda zvolíme činnosť 2). Súbor s údajmi adresára uchovávame na samostatnej kazete (je uložený vždy na začiatku kazety). Zápis na kazetu uskutočníme podľa pokynov zobrazených na obrazovke.

Pri každej ďalšej práci s programom začíname načítaním údajov z kazety (v ponuke zvolíme činnosť 1). Doterajší obsah adresára sa prepíše z kazety do pamäte počítača a môžeme vkladať ďalšie adresy, opravovať už existujúce adresy, vymazávať staré adresy, vyhľadávať adresy a vypisovať ich na obrazovku alebo na tlačiareň.

Pri vyhľadávaní adries najprv odpovieme na dotaz, či chceme výstupy na obrazovku, alebo na tlačiareň, a potom vložíme príznak hľadania. Príznak hľadania má rovnakú štruktúru ako záznam v adresári; počítač nám ponúkne zadanie príznakov v 7 položkách. Ak nevložíme žiadne príznaky (pre všetky položky stlačíme Return), zobrazujú sa nám (alebo vytlačia sa) všetky záznamy adresára. Ak chceme nájsť nejaký konkrétny záznam (alebo skupinu záznamov), vložíme ako príznak hľadania text niektorej položky, podľa ktorého je počítač vstave záznam (alebo záznamy) identifikovať. Ak napr. vložíme ako príznak hľadania konkrétne priezvisko, počítač nájde a zobrazí všetky záznamy adresára týkajúce sa osôb s týmto priezviskom.

Pri opravách záznamov v adresári (činnosť 4) používame 4 riadiace príkazy:

+	zobrazí a pripraví na opravy nasledujúci záznam,
–	zobrazí a pripraví na opravy predchádzajúci záznam,
*	umožní vykonanie opravy daného záznamu,
P	umožní návrat do ponuky činností.

Po každej zmene adresára (vložení nových záznamov, oprave starých záznamov) nezabudnite súbor s adresárom znovu zapísať na kazetu. Odporúčame pracovať s 3 kazetami, na jednej uchovávať program ADRESÁR (po jeho vložení a odladení ho zapíšeme na kazetu povelom CSAVE a pred každým použitím načítame do pamäte počítača povelom CLOAD) a na ďalších dvoch uchovávať starú a novú verziu súboru údajov adresára.

```
10 REM Program 2.30 - ADRESAR, 29. 10. 88
20 REM Atari basic
100 REM ============================
110 REM Deklaracie retazcov
120 REM ============================
130 DIM ZOZNAM$(21000),CHYBA$(38)
140 ZOZNAM$(1)=" ":ZOZNAM$(21000)=" "
150 ZOZNAM$(2)=ZOZNAM$
160 DIM FUNKCIA$(119)
```

```
165 FUNKCIA$(1,17)="NACITANIE ADRIES  "
170 FUNKCIA$(18,34)="ZAPISANIE ADRIES "
180 FUNKCIA$(35,51)="VLOZENIE ADRESY  "
190 FUNKCIA$(52,68)="ZMENA ADRESY     "
200 FUNKCIA$(69,85)="VYMAZANIE ADRESY "
210 FUNKCIA$(86,102)="VYHLADANIE ADRESY"
220 FUNKCIA$(103,119)="KONIEC PROGRAMU  "
230 DIM POLE$(70)
240 POLE$(1,10)="Oznacenie "
250 POLE$(11,20)="Meno      "
260 POLE$(21,30)="Priezvisko"
270 POLE$(31,40)="Ulica     "
280 POLE$(41,50)="Mesto, PSC"
290 POLE$(51,60)="Telefon   "
300 POLE$(61,70)="Poznamka  "
310 DIM ZN$(20)
320 DIM ODPOVED$(1)
330 DIM V$(20),N$(20)
340 DIM HLAD$(140)
350 GOTO 1000
400 REM ============================
410 REM Hlavicka programu
420 REM ============================
430 PRINT CHR$(125)
440 FOR I=1 TO 38:PRINT "=";:NEXT I
450 POSITION 12,2
460 PRINT " A D R E S A R "
470 FOR I=1 TO 38:PRINT "=";:NEXT I
480 RETURN
500 REM ============================
510 REM Hlasenie chyb
520 REM ============================
530 POSITION 1,23
540 PRINT CHYBA$;
550 PRINT CHR$(253);CHR$(253);:REM Dvakrat pipnutie
570 PRINT CHR$(156);:REM Vymazanie riadku
580 RETURN
600 REM ============================
610 REM Zahlavie cinnosti programu
620 REM ============================
630 GOSUB 400
640 POSITION 10,5
650 FOR I=1 TO 19:PRINT "*";:NEXT I
660 POSITION 10,6
670 PRINT "*";FUNKCIA$(FUNKCIA#17-16,FUNKCIA#17);"*"
680 POSITION 10,7
690 FOR I=1 TO 19:PRINT "*";:NEXT I:PRINT
700 RETURN
800 REM ============================
810 REM Kontrola obsadenia pamati
820 REM ============================
830 IF UKAZ>0 THEN RETURN
840 CHYBA$="V pamati nie su ziadne udaje!"
850 GOSUB 500
860 POP :GOTO 1000
900 REM ============================
910 REM Pozicia adresy
920 REM ============================
930 OD=(UKAZ#140-139)+(INDEX#20-20)
940 DO=OD+19
950 RETURN
1000 REM ============================
1010 REM Hlavna ponuka programu
1020 REM ============================
1030 GOSUB 400
1040 POSITION 1,7
1050 PRINT "Funkcie programu ":POSITION 1,8
1060 PRINT "................."
1070 PRINT
1080 PRINT "     1. Nacitanie udajov"
1090 PRINT "     2. Zapisanie udajov"
```

```
1100 PRINT "     3. Vlozenie adresy"
1110 PRINT "     4. Zmena adresy"
1120 PRINT "     5. Vymazanie adresy"
1130 PRINT "     6. Vyhladanie adresy"
1140 PRINT "     7. Ukoncenie prace"
1150 POSITION 10,18:PRINT "VASA VOLBA: ";
1160 POSITION 23,18:INPUT FUNKCIA
1170 IF FUNKCIA=0 OR FUNKCIA>7 THEN CHYBA$="Neplatna
     hodnota":GOSUB 500:GOTO 1160
1175 GOSUB 600
1180 ON FUNKCIA GOTO 5000,10000,15000,20000,25000,30
     000,31000
5000 REM ============================
5010 REM Nacitanie udajov z kazety
5020 REM ============================
5040 POSITION 2,10
5050 PRINT "Vykonajte tieto pripravy:"
5060 PRINT "........................."
5070 PRINT
5080 PRINT "1. Vlozte kazetu"
5090 PRINT "2. Pretocte kazetu na zaciatok"
5100 PRINT "3. Po odzneni zvukoveho signalu"
5105 PRINT "   stlacte 'PLAY'"
5110 PRINT "4. Stlacte 'RETURN'"
5120 PRINT
5130 POSITION 5,19
5140 PRINT "Ste pripraveny  (A,N): ";
5150 INPUT ODPOVED$
5160 IF ODPOVED$="A" THEN 5180
5170 CHYBA$="Stlacte klaves  'A'!":GOSUB 500:GOTO 51
     30
5180 OPEN #1,4,0,"C:ADRESY"
5190 INPUT #1;Z
5200 FOR UKAZ=1 TO Z
5210 FOR INDEX=1 TO 7
5220 GOSUB 900
5225 INPUT #1;ZW$
5230 ZOZNAM$(OD,DO)=ZW$
5240 NEXT INDEX
5250 NEXT UKAZ:UKAZ=UKAZ-1
5260 CLOSE #1
5270 GOSUB 600
5275 POSITION 2,15
5280 PRINT "Adresy su v pamati"
5285 PRINT "Cakajte..."
5290 FOR I=1 TO 1000:NEXT I
5300 GOTO 1000
10000 REM ============================
10010 REM Zapis udajov na kazetu
10020 REM ============================
10030 GOSUB 800
10040 POSITION 2,10
10050 PRINT "Vykonajte tieto pripravy:"
10060 PRINT "........................."
10070 PRINT
10080 PRINT "1. Vlozte kazetu"
10090 PRINT "2. Pretocte kazetu"
10100 PRINT "3. Po odzneni zvukoveho signalu"
10105 PRINT "   stlacte 'RECORD' a 'PLAY'"
10110 PRINT "4. Stlacte 'RETURN'"
10120 PRINT
10125 POSITION 5,19
10130 PRINT "Ste pripraveny (A,N): ";
10140 INPUT ODPOVED$
10160 IF ODPOVED$="A" THEN 10180
10170 CHYBA$="Stlacte klaves  'A'!":GOSUB 800:GOTO 1
     0125
10180 OPEN #1,8,0,"C:ADRESY"
10190 PRINT #1;UKAZ
10195 Z=UKAZ
10200 FOR UKAZ=1 TO Z
```

```
10210 FOR INDEX=1 TO 7
10220 GOSUB 300
10230 PRINT #1;ZOZNAM$(OD,DO)
10240 NEXT INDEX
10250 NEXT UKAZ:UKAZ=UKAZ-1
10260 CLOSE #1
10270 GOSUB 500
10275 POSITION 2,15
10280 PRINT "Adresy boli zapisane na kazetu!":PRINT
    "Cakajte..."
10290 FOR I=1 TO 1000:NEXT I
10300 GOTO 1000
15000 REM ==========================
15010 REM Vlozenie adresy
15020 REM ==========================
15030 UKAZ=UKAZ+1
15040 PRINT
15050 FOR INDEX=1 TO 7
15055 GOSUB 300
15060 PRINT POLE$(INDEX*10-9,INDEX*10);" ";
15065 INPUT ZW$
15070 ZOZNAM$(OD,DO)=ZW$
15080 NEXT INDEX
15090 POSITION 2,22
15100 PRINT "Su udaje spravne? (A,N) ";
15110 INPUT ODPOVED$
15120 IF ODPOVED$="A" THEN 15150
15130 IF ODPOVED$="N" THEN UKAZ=UKAZ-1:GOSUB 600:GOT
    O 15000
15140 CHYBA$="Stlacte 'A' alebo 'N'!":GOSUB 500:GOTO
    15090
15150 POSITION 2,22
15160 PRINT "Dalsia adresa? (A, N)";
15170 INPUT ODPOVED$
15180 IF ODPOVED$="A" THEN GOSUB 600:GOTO 15000
15190 IF ODPOVED$="N" THEN 1000
15200 CHYBA$="Stlacte 'A' alebo 'N'!":GOSUB 500:GOTO
    15150
20000 REM ==========================
20010 REM Zmena adresy
20020 REM ==========================
20025 GOSUB 800
20027 V$(1)=" ":V$(20)=" "
20028 V$(2)=V$
20030 ZZZZ=1:Z=UKAZ
20040 POSITION 2,10
20050 FOR INDEX=1 TO 7
20055 UKAZ=ZZZZ:GOSUB 300
20060 PRINT INDEX;"-";POLE$(INDEX*10-9,INDEX*10);
20062 PRINT " ";
20065 PRINT ZOZNAM$(OD,DO)
20070 NEXT INDEX
20075 POSITION 5,18
20080 PRINT "Prikaz (+,-,* P)";
20085 INPUT ODPOVED$
20090 IF ODPOVED$="P" THEN UKAZ=Z:GOTO 1000
20100 IF ODPOVED$="+" AND ZZZZ<Z THEN ZZZZ=ZZZZ+1:GO
    TO 20040
20110 IF ODPOVED$="-" AND ZZZZ>1 THEN ZZZZ=ZZZZ-1 GO
    TO 20040
20120 IF ODPOVED$="*" THEN 20140
20130 CHYBA$="Chyba!":GOSUB 500:GOTO 20075
20140 POSITION 5,19 .
20145 PRINT "CISLO POLA (1-7)  0 ";
20150 POSITION 22,19:INPUT INDEX
20155 IF INDEX>0 AND INDEX<8 THEN 20165
20160 CHYBA$="Zvolte cislo 1 - 7!":GOSUB 500:GOTO 20
    150
20165 UKAZ=ZZZZ:GOSUB 300
20170 PRINT "Novy obsah  ";
20173 ZOZNAM$(OD,DO)=V$
```

```
20174 PRINT ZOZNAM$(OD,DO) POSITION 13,20
20180 INPUT ZW$
20185 ZOZNAM$(OD,DO)=ZW$
20190 GOSUB 600:GOTO 20040
25000 REM ============================
25010 REM Vymazanie adresy
25020 REM ============================
25030 GOSUB 800
25035 Z=UKAZ
25036 V$(1)=" ":V$(20)=" ":V$(2)=V$
25037 N$(1)=" ":N$(20)=" ":N$(2)=N$
25040 POSITION 2,10
25050 PRINT "Meno          ";:INPUT V$
25060 PRINT "Priezvisko    ";:INPUT N$
25070 FOR UKAZ=1 TO Z
25080 INDEX=2:GOSUB 900
25090 IF V$<>ZOZNAM$(OD,OD+LEN(V$)-1) THEN 25130
25100 INDEX=3:GOSUB 900
25110 IF N$<>ZOZNAM$(OD,OD+LEN(N$)-1) THEN 25130
25120 POP :GOTO 25150
25130 NEXT UKAZ
25140 CHYBA$="Adresa nie je v pamati":GOSUB 500:GOTO
    1000
25150 GOSUB 600:POSITION 2,10
25160 FOR INDEX=1 TO 7
25170 GOSUB 900:PRINT POLE$(INDEX*10-9,INDEX*10);:PR
   INT " ";
25190 PRINT ZOZNAM$(OD,DO)
25200 NEXT INDEX
25210 POSITION 1,18
25220 PRINT "Chcete adresu vymazat (A,N): ";
25230 INPUT ODPOVED$
25240 IF ODPOVED$="A" THEN 25270
25250 IF ODPOVED$="N" THEN UKAZ=Z:GOTO 1000
25260 CHYBA$="Stlacte 'A' alebo 'N'":GOSUB 500:GOTO
   25210
25270 FOR VETA=UKAZ TO Z-1
25280 FOR INDEX=1 TO 7
25290 UKAZ=VETA:GOSUB 900
25300 ZOZNAM$(OD,DO)=ZOZNAM$(OD+140,DO+140)
25310 NEXT INDEX
25320 NEXT VETA
25330 UKAZ=Z-1
25340 GOTO 1000
30000 REM ============================
30010 REM Vyhladanie adresy
30020 REM ============================
30030 GOSUB 800
30040 POSITION 1,10:PRINT "Tlaciaren alebo obrazovka
   ? (T, O): ";
30050 INPUT ODPOVED$
30060 IF ODPOVED$="T" THEN OPEN #1,8,0,"P:":GOTO 300
   90
30070 IF ODPOVED$="O" THEN 30090
30080 CHYBA$="Stlacte 'T' alebo 'O'":GOSUB 500:GOTO
   30040
30090 POSITION 1,12
30100 PRINT " Priznak hladania:"
30110 PRINT "................."
30120 PRINT
30125 Z=UKAZ
30130 HLAD$(1)=" ":HLAD$(140)=" ":HLAD$(2)=HLAD$
30135 FOR INDEX=1 TO 7
30140 PRINT POLE$(INDEX*10-9,INDEX*10);
30145 PRINT "  ";
30150 INPUT ZW$
30160 HLAD$(INDEX*20-19,INDEX*20)=ZW$
30170 NEXT INDEX
30180 FOR UKAZ=1 TO Z
30190 NAJDENE=0
```

```
30200 FOR INDEX=1 TO 7
30210 ZW$=HLAD$(INDEX*20-19,INDEX*20)
30215 IF ZW$(1,5)="     " THEN NAJDENE=NAJDENE+1:GOT
    O 30230
30216 GOSUB 900
30220 IF ZW$=ZOZNAM$(OD,DO) THEN NAJDENE=NAJDENE+1
30230 NEXT INDEX
30240 IF NAJDENE<>7 THEN 30350
30250 GOSUB 600
30260 IF ODPOVED$="T" THEN PRINT #1
30265 IF ODPOVED$="O" THEN PRINT
30270 Z1=INDEX
30275 FOR I=1 TO 7
30280 IF ODPOVED$="T" THEN PRINT #1,POLE$(I*10-9,I*1
    0);";";
30285 IF ODPOVED$="O" THEN PRINT POLE$(I*10-9,I*10),
    ";";
30290 INDEX=I:GOSUB 900
30300 IF ODPOVED$="T" THEN PRINT #1,ZOZNAM$(OD,DO)
30305 IF ODPOVED$="O" THEN PRINT ZOZNAM$(OD,DO)
30310 NEXT I
30320 INDEX=Z1
30325 PRINT
30340 PRINT "Stlacte 'RETURN'!";:INPUT V$
30350 NEXT UKAZ:UKAZ=UKAZ-1
30360 GOSUB 600:POSITION 1,10
30370 PRINT "** KONIEC UDAJOV **"
30380 PRINT "Stlacte 'RETURN'!";:INPUT V$:IF ODPOVED
    $="T" THEN CLOSE #1
30390 UKAZ=Z:GOTO 1000
31000 REM ==========================
31010 REM Ukoncenie prace
31020 REM ==========================
31030 IF UKAZ=0 THEN 31150
31040 POSITION 2,12
31050 PRINT "Ulozili ste udaje na kazetu (A/N)  ";
31060  INPUT ODPOVED$
31070 IF ODPOVED$="N" THEN 1000
31080 IF ODPOVED$="A" THEN 31100
31090 CHYBA$="Stlacte 'A' ALEBO 'N'!":GOSUB 500:GOTO
    31040
31100 GOSUB 600:POSITION 2,12
31110 PRINT "Program mozete znova odstartovat"
31120 PRINT "prikazom 'GOTO 1000' bez toho, ze"
31130 PRINT "by sa stratili udaje, ulozene v"
31135 PRINT "pamati pocitaca"
31140 PRINT
31150 END
```

František Turnovec

Príloha 1

CHYBOVÉ HLÁSENIA V JAZYKU BASIC G

Správy o chybách sa uvádzajú v dvoch formátoch:
* * * správa * * *
v priamom režime a
* * * správa IN LINE čr * * *
v programovom režime (*čr* je číslo programového riadka, v ktorom sa vyskytla chyba). Hlásenia chýb na počítačoch PMD 85–1, PMD 85–2 a Didaktik Alfa sú rovnaké. Pre používateľov počítača PMD 85–2 uvádzame okrem textu správy o chybe a jeho vysvetlenia aj kód chyby, pomocou ktorého možno ošetriť chyby v programe využitím príkazu ON ERR.

kód chyby	správa	význam
1	SUBSCR. RNG	Index použitej indexovanej premennej je mimo rozsah deklarovaný v príkaze DIM, alebo ste použili rovnaké meno pre jednorozmernú a dvojrozmernú, resp. trojrozmernú indexovanú premennú; správa môže znamenať aj to, že ste sa pokúsili použiť nedefinovanú používateľskú funkciu.
2	ARR. ALOC.	Nesprávny rozsah poľa; požadujete väčšie pole, než umožňuje rozsah voľnej pamäte, alebo dvakrát deklarujete to isté pole.
3	FNC. PARAM.	Argument použitej funkcie nezodpovedá formálnemu parametru, uvedenému v definícii funkcie.
4	ONLY IN PG.	V priamom režime ste použili príkaz, ktorý je prípustný len v programovom režime (napr. INPUT).
5	NO FOR STM	Nesprávne vytvorený cyklus; najčastejšie k príkazu FOR chýba príslušný príkaz NEXT, alebo zle vložené viacnásobné cykly.
6	DATA EXHAU	Príkazom READ požadujete ďalšie údaje, ale zoznam konštánt definovaný prí-

		kazmi DATA je už vyčerpaný, alebo úplne chýba.
7	PG TOO BIG	Príliš dlhý program, nezmestí sa do pamäte (pri úprave použite viacpríkazové riadky, skráťte reťazce, vynechajte nadbytočné medzery a pod.).
8	OVERFLOW	Hodnota vyhodnocovaného aritmetického výrazu je mimo povoleného rozsahu od 10^{-38} do 10^{38} (v absolútnej hodnote).
9	SYNTAX ERR	Príkaz je zapísaný v zlom formáte, alebo obsahuje preklep.
10	RETURN ERR	Chyba pri návrate z podprogramu, alebo RETURN bez príslušného GOSUB.
11	NUMB. NONEX	V príkazoch GOTO, GOSUB, IF THEN alebo ON sa odvolávate na číslo riadka, ktoré sa v programe nevyskytuje.
12	DV BY ZERO	Pri výpočte došlo k deleniu nulou.
13	CAN'T CONT	Nesprávne použitý povel CONT pre pokračovanie v spracovaní programu; postupnosť spracovania ste porušili zadaním iného príkazu alebo povelu.
14	STRNG LONG	Dĺžka reťazca prevyšuje povolených 255 znakov.
15	NO STR. SPC	V pamäti už nie je voľné miesto pre reťazce.
16	STR. ALGRTH	Dĺžka reťazcového výrazu je príliš veľká, alebo výraz je príliš zložitý; výraz treba rozdeliť na viac jednoduchších výrazov.
17	TYPE CONV.	Chyba pri priraďovaní, typ konštanty alebo výrazu sa nezhoduje s typom premennej.
18	FILE SMALL	Údajový súbor na kazete má iné deklarované pole, ako je príslušné pole v programe.
19	INPUT ERR	Chyba pri vkladaní údajov.
20	FIELD LOST	Prekročený počet vstupných údajov, ktoré požaduje príkaz INPUT.
21	FILE BOUND	Prekročený maximálny prípustný počet záznamov na kazete.
22	STOP	Prerušenie spracovania programu.

23	FILE ERROR	Chyba pri prenose programových alebo údajových súborov z kazety.

Príloha 2

CHYBOVÉ HLÁSENIA V JAZYKU ATARI BASIC

Správy o chybách sa uvádzajú v tvare
ERROR – kód AT LINE čr
kde *kód* je číselný kód chyby a *čr* je číslo programového riadka, v ktorom sa vyskytla chyba. Poznamenávame, že syntaktické chyby v jednotlivých príkazoch hlási Atari basic už pri vkladaní príkazov. Ďalej uvádzame kódy a charakteristiky jednotlivých chýb.

kód chyby	popis chyby
2	Nepostačujúca pamäť, nemožno už vložiť ďalšie príkazy a premenné.
3	Nesprávna hodnota indexu v indexovanej premennej.
4	Viac ako 128 premenných (v programe možno použiť najviac 128 rôznych názvov premenných).
5	Príliš dlhý reťazec (pri pokuse vložiť reťazec, ktorého dĺžka presahuje deklaráciu v príkaze DIM).
6	Príkaz READ požaduje viac údajov, ako je k dispozícii v zozname konštánt definovanom príkazmi DATA.
7	Číslo riadka väčšie ako 32767.
8	Chyba pri vstupe (pokus vložiť do číselnej premennej nečíselný údaj).
9	Chyba v príkaze DIM; celkový rozsah dimenzovaných číselných polí presahuje 5460, alebo celkový rozsah dimenzovaných reťazcových premenných presahuje 32767, alebo pole bolo dimenzované dvakrát, alebo sa použil nedimenzovaný reťazec.
11	Delenie nulou alebo použitie čísel mimo rozsahu od 10^{-99} do 10^{99} (v absolútnej hodnote).
12	V príkazoch ON, GOTO, GOSUB, IF THEN sa odvolávame na neexistujúce číslo riadka.
13	Nesprávne vytvorený cyklus; k príkazu FOR chýba NEXT, alebo zle vložené viacnásobné cykly.
14	Príkazový riadok je príliš zložitý, alebo príliš dlhý.
15	K príkazu RETURN alebo NEXT chýba príkaz GOSUB alebo FOR.
16	Chyba v návrate z podprogramu.

17	Syntaktická chyba.
18	Reťazec vo funkcii VAL nie je číselným reťazcom.
19	Príliš dlhý program, zavádzaný z kazety alebo z diskety.
20	Chyba v čísle súboru – číslo väčšie ako 7, alebo menšie ako 1.
21	Používateľ sa pokúša zaviesť do pamäte súbor, ktorý nie je kompatibilný s basicom.
128	Chybné použitie klávesu Break (napr. pri vstupno-výstupnej operácii).
129	Súbor s rovnakým číslom je už otvorený.
130	Neexistujúce zariadenie (používateľ sa pokúša pracovať so zariadením, ktoré nie je definované).
131	Zariadenie iba pre výstup (pokus o vstup z výstupného zariadenia, napr. z tlačiarne).
132	Príkaz nie je prípustný pre dané zariadenie.
133	Súbor, s ktorým chceme pracovať, nebol otvorený príkazom OPEN.
134	Neplatné číslo súboru.
135	Zariadenie iba pre vstup (pokus o výstup na zariadenie, ktoré bolo definované ako vstupné).
136	Koniec súboru (požaduje sa vstup zo súboru, z ktorého sa už prečítal posledný záznam).
137	Čítanie záznamu, ktorý je dlhší ako maximálna prípustná dĺžka (v basicu jeden záznam v súbore môže mať najviac 119 B).
138	Zariadenie neodpovedá (napr. vypnutá tlačiareň pri pokuse o výstup na tlačiareň).
139	Chyba v periférnom zariadení.
140	Chyba prenosu vstupných dát z periférneho zariadenia (napr. pri čítaní z kazety).
141	Kurzor mimo dosahu (pokus o umiestnenie kurzora mimo oblasti, definovanej grafickým režimom).
142	Chyba prenosu dát.
143	Chyba prenosu dát.
144	Chyba pri zápise na disketu (napr. pokus o zápis na disketu chránenú proti zápisu).
145	Nesprávny grafický režim (pokus definovať grafický režim s nesprávnym číslom).
146	Funkcia sa nevykonala.
147	Nestačí pamäť pre operácie v danom grafickom režime.

148	Chybné číslo zariadenia (napr. disketovej jednotky).
161	Priveľa otvorených súborov.
162	Plná disketa (celá kapacita diskety využitá).
163	Chyba vstupu-výstupu pri práci s disketou.
164	Chybné číslo súboru.
165	Chybný názov súboru.
166	Chybný parameter v príkaze POINT.
167	Chránený súbor (pokus o zápis do chráneného súboru).
168	Neplatný príkaz.
169	Plný adresár diskety (obsahuje už 64 súborov).
170	Pokus o prácu s neexistujúcim súborom.
171	Chyba v príkaze POINT.
172	Pokus o prácu so súbormi vytvorenými operačným systémom DOS 1 v operačnom systéme DOS 2.
173	Chybné sektory na diskete.

Príloha 3

TABUĽKA ASCII KÓDU

NUL	0	space	32	@	64	'	96
SOH	1	!	33	A	65	a	97
STX	2	"	34	B	66	b	98
ETX	3	#	35	C	67	c	99
EOT	4	$	36	D	68	d	100
ENQ	5	%	37	E	69	e	101
ACK	6	&	38	F	70	f	102
BEL	7	'	39	G	71	g	103
BS	8	(	40	H	72	h	104
HT	9	)	41	I	73	i	105
LF	10	*	42	J	74	j	106
VT	11	+	43	K	75	k	107
FF	12	,	44	L	76	l	108
CR	13	-	45	M	77	m	109
SO	14	.	46	N	78	n	110
SI	15	/	47	O	79	o	111
DLE	16	Ø	48	P	80	p	112
DC1	17	1	49	Q	81	q	113
DC2	18	2	50	R	82	r	114
DC3	19	3	51	S	83	s	115
DC4	20	4	52	T	84	t	116
NAK	21	5	53	U	85	u	117
SYN	22	6	54	V	86	v	118
ETB	23	7	55	W	87	w	119
CAN	24	8	56	X	88	x	120
EM	25	9	57	Y	89	y	121
SUB	26	:	58	Z	90	z	122
ESC	27	;	59	[	91	{	123
FS	28	<	60	\	92	|	124
GS	29	=	61	]	93	}	125
RS	30	>	62	^	94	~	126
US	31	?	63	_	95	DEL	127

V štyroch dvojiciach stĺpcov tabuľky prvý stĺpec udáva príslušný znak a druhý stĺpec desiatkovú hodnotu ASCII kódu. Kódy 0 až 31 a 127 sa na počítačoch PMD 85 a Didaktik Alfa nezobrazujú, slúžia ako kontrolné a riadiace znaky. Na PMD 85–1 a Didaktiku Alfa sa nezobrazujú ani malé písmená a ďalšie znaky od kódu 96 (tieto sa však zobrazujú na PMD 85–2).

V Atari basicu sa kódy 0 až 31 zobrazujú ako špeciálne grafické znaky, kódy 32 až 122 a 124 sa zobrazujú rovnako, ako je uvedené v tabuľke, kódy 123 a 125 až 127 skúžia ako riadiace znaky.

Pre kódy z intervalu 128 až 255 sa na PMD 85 a Didaktiku Alfa všetky znaky opakujú. V Atari basicu reprezentujú kódy z intervalu 128 až 255 inverzne zobrazené znaky, zodpovedajúce kódom 0 až 127.

LITERATÚRA

Barát, Ľ. – Palkovič, F.: Príručka užívateľa – Didaktik Alfa. Banská Bystrica, Učebné pomôcky, n. p., 1985.
Burger, I. – Čičák, P. – Kadlic, A.: Mládež, elektronika, mikropočítače. Bratislava, Smena 1988.
Gabčo, P.: Basic Alfa, slovník jazyka. Banská Bystrica, Učebné pomôcky, n. p., 1986.
Janev, K.: Zabavni kompjut'rni programi. Sofia, Narodna mladež 1987.
Schreiber, L. M.: Atari Programming with 55 Programs. Blue Ridge Summit, TAB Books, Inc. 1982.
Szczepanowski, K.: Atari 130/XE, 600/XL, 800/XL für Einsteiger. Düsseldorf, Data Becker 1986.
Vickers, S.: Sinclair ZX-Spectrum Basic Programming. Cambridge, Sinclair Research, Ltd. 1982.
Vojtek, V. – Návrat, P. – Drlík, P.: Programovanie v jazyku basic. Bratislava, SÚV SZM 1985.
Výrut, K. – Vejmola, S.: Programy pro počítače IQ 151. Praha, SPN 1987.
Firemná literatúra
Atari 130/XE Handbuch. Rauenheim, Atari Corp. 1985.
Commodore 64 Basic. Commodore Electronics, Ltd. 1984.
DCD–PRT–80GS Graphics Matrix Printer. Diósd, Datacoop 1985.
Osobný mikropočítač PMD 85. Príručka pre používateľa. I. Návod na použitie. Bratislava, Tesla 1985.
Osobný mikropočítač PMD 85. Návod na použitie a obsluhu mikropočítača PMD 85–2. Bratislava, Tesla 1986.
Posledovateľno-pečatajuščeje ustrojstvo D–100. Instrukcija po ekspluatacii. Mera-Blonie 1986.
Časopisy
Atari Magazin, das unabhängige Magazin für alle Ataris. Ročník 1987, 1988.
Elektrón, mesačník pre mladých o vede a technike. Ročník 1985, 1986, 1987.
Elektronika, měsíčník pro elektronizaci národního hospodářství. Ročník 1988.
Nauka i žizň. Ročník 1986, 1987, 1988.
Zenit pionierov, dvojtýždenník o vede a technike. Ročník 1986–87, 1987 až 88.

Edícia Sputnik

František Turnovec

SKÚSTE TO S BASICOM

Vydala Smena, vydavateľstvo SÚV SZM, Bratislava,
ako svoju 3 686. publikáciu.
Rok vydania 1989.
Vydanie prvé.
Edícia Sputnik.
Typografia Jozef Pernecký.
Zodpovedný redaktor Hviezdoslav Herman.
Výtvarný redaktor Jozef Gális.
Technická redaktorka Mária Kvíteková.
Jazyková redaktorka Marta Kastlová.
AH 16,02 /text 10,01; ilustr. 6,01/. VH 16,25.
Náklad 8 000 výtlačkov. SÚKK č. 121/I–88.
Vytlačila Pravda, tlačový kombinát KSS,
TZP, závod 01, Bratislava, Štúrova 4.

ISBN 80-221-0006-4
14 Kčs 30,–

BASIC JE ZÁKLADNÝM programovacím jazykom 8-bitových osobných počítačov. Táto publikácia ponúka úvod do programovania v basicu na počítačoch PMD 85 a Didaktik Alfa (tzv. basic G) a na počítačoch Atari 800/XL a Atari 130/XE (tzv. Atari basic). Výklad sa opiera o 95 názorných príkladov s praktickými ukážkami jednotlivých programátorských zručností a obsahuje 50 ucelených programov na jednoduché výpočty, spracovanie textov, počítačovú grafiku, počítačové hry a výučbové aplikácie. Osobitnú pozornosť venuje „kresleniu na obrazovke", t. j. počítačovej grafike, a je bohato ilustrovaný vyše 80 obrázkami grafických výstupov z počítača.

Publikácia je určená pre žiakov základných a stredných škôl, ktorí majú záujem o prácu s osobnými počítačmi a ich využívanie pri odpočinku aj pri riešení praktických úloh, či už doma, v záujmovej činnosti krúžkov mladých programátorov alebo v škole. Hoci je adresovaná predovšetkým začiatočníkom, môžu v nej nájsť užitočné informácie a inšpiráciu na tvorbu vlastných programov aj skúsenejší počítačoví „fanúšikovia".

14
Kčs 30,–

KNIHY ZO SMENY,
VYDAVATEĽSTVA SÚV SZM
O ELEKTRONIKE
A MIKROPOČÍTAČOCH

I. Burger – P. Návrat
MIKROPOČÍTAČE, PROGRAMY, ĽUDIA
E. Drobný
VEĽKÝ SVET MALÝCH VECÍ
I. Burger a kol.
MLÁDEŽ, ELEKTRONIKA A MIKROPOČÍTAČE
P. Gabriš a kol.
SVET POČÍTAČOV
I. Burger
ZVEDAVÉ OTÁZKY OKOLO MIKROPOČÍTAČOV

PRIPRAVUJEME:
Š. Kassay
INFORMATIZÁCIA SPOLOČNOSTI

ISBN 80-221-0006-4