

**ZX Spectrum +**

## SPECTRUM SOFTWARE

Celý rozsah dosud vydaných softwarů počítače Spectrum (včetně všech stávajících titulů) je zcela slučitelný s vaším novým počítačem ZX Spectrum +.

### Úvod k počítači ZX Spektrum +

Výzkum společnosti Sinclair vyvinul značné pokroky v mikročipové technologii, které je využito ve výpočetnictví.

S ohledem na tuto výhodu může nabízet nejlevnější mikropočítač na světě - ZX80. Po kombinaci stále větších schopností vnitřního mikroprocesoru se stále větším vybavením paměti bylo vyvinuto několik počítačů - ZX81, ZX Spektrum a QL. Dalším světovým prvenstvím byla snadnost obsluhy, design a způsob, jakým tyto počítače pracují.

Počítač ZX Spektrum + posunuje výzkum společnosti Sinclair opět o jeden krok dál. V tomto počítači máte stroj se všemi nejlepšími rysy počítačů řady Spektrum ve zdokonalené verzi, která jej činí nejvýkonnějším a nejoblíbenějším ze všech mikropočítačů s nejsnadnější obsluhou.

Doufáme, že zcela využijete veškerých možností, které vám náš nový počítač nabízí.

|  |      |
|--|------|
| O B S A H :                                    | str. |
| Spouštění                                      | 3    |
| Zahájení programování                          | 17   |
| Naučte se znát svůj ZX Spectrum+               | 41   |
| Naučte se programovací jazyk<br>Sinclair Basic | 49   |

Napsal Neil Ardley

Vydal Dorling Kindersley Ltd ve spolupráci  
se společností Sinclair Research Ltd

#### JAK POUŽÍVAT TUTO KNIHU :

Tato kniha je průvodce použitím vašeho počítače ZX Spectrum + ,  
obsahuje čtyři barevně odlišené kapitoly. Pro nalistování vhodné  
kapitoly jednoduše otevřete knihu v části se správnou barvou.

#### 1 - Spouštění

Nastavení vašeho počítače ZX Spectrum +

Vyladění vašeho televizoru

Seřízení odstraňovače závad

Co všechno váš ZX Spectrum dokáže

Jak využívat připraveného software

Jak založit program

Odstraňování závad při zakládání software

-----

+ Příručka v českém jazyce odpovídá řazením stran anglickému  
originálu, tzn. že odkazy se vztahují souběžně na obě vydání.  
S ohledem na reprografickou úpravu je čs. vydání černobílé.

## 2 - Zahájení programování

Klávesnice - ovládací deska počítače

Jak pracovat s klávesami

Televizní počítačka

Barvy a jak je využít

Jednoduché grafiky DIY

Nákresy na obrazovce

Konstrukce vlastních vzorů a obrazů

Jak vytvářet vaše vlastní počítačové typy

Animace

Jak vyrábět hudební a zvukové efekty

Jak uchovávat vlastní programy

Odstraňování závad při ukládání software

## 3 - Naučte se znát váš ZX Spectrum+

Co je uvnitř?

Jak vlastně pracuje váš počítač ZX Spectrum+?

Jak připojit periferie

Uspořádání paměti ZX Spectrum+

## 4 - Naučte se programovací jazyk Sinclair Basic

Programovací referenční příručka pro klíčová slova jazyka

Sinclair Basic

Odkazy na obrazovce

Co je za BASICem

Počítačový žargon - co to znamená

## 1. SPOUŠTĚNÍ

Tato kapitola ukazuje, jak začít využívat schopností počítače ZX Spectrum+. Naleznete zde popis spouštění počítače kdykoliv si budete přát. Poté můžete několika jednoduchými programy ověřit schopnosti Spectra zobrazit barevné grafiky a zvuk nebo můžete použít připraveného softwaru, jako jsou např. počítačové hry. Věříme, že brzy oceníte možnosti spolupráce s vaším novým počítačem.

## Příprava vašeho počítače ZX Spectrum+

Před prvním spuštěním vašeho počítače nejprve pročtete seznam příslušenství uvedený níže, přesvědčte se o kompletnosti dodávky. Dále postupujte podle následujících pokynů, které popisují, jak počítač připojit k monitoru a ke zdroji elektrické energie. V tomto stadiu přípravných prací ještě nepotřebujete kazetový magnetofon. Veškerá propojení proveďte pečlivě, jestliže náhodně odpojíte nebo vypnete zdroj elektrické energie během provozu počítače, dojde k nenávratnému znehodnocení programu včetně výsledků nebo informací a vám nezbyvá nic jiného, než vložit program do počítače znovu.

Ukončíte-li práci s počítačem, odpojíte nejprve napáječ od počítače vytažením zástrčky na zadní stěně přístroje a pak odpojte napáječ od elektrické sítě.

Máte veškeré náležitosti?

Při vybalování naleznete:

1. Počítač ZX Spectrum+
2. Síťový napáječ ZX Spectrum, který dodává 9 V ss napětí pro počítač
3. Anténní kabel- tím propojíte váš počítač s televizním přijímačem
4. Kabel ke kazetovému magnetofonu - slouží k propojení kazetového přehrávače-nahrávače s počítačem Spectrum
5. Záruční list
6. Demonstrační kazeta
7. Tento návod k použití

Dále budete ještě potřebovat:

1. Televizní přijímač
2. Kazetový magnetofon
3. Síťovou zástrčku

Otázky a odpovědi:

Musím mít barevný televizor?

Ne. Ovšem v tomto případě nebudete moci sledovat barvy, které Spectrum vysílá.

Bude vhodný jakýkoliv televizní přijímač?

Váš počítač Spectrum může být připojen k libovolnému televiznímu přijímači, který je schopen přijímat signály v pásmu UHF a pro barevný příjem je vybaven dekodérem PAL. U starších televizorů a u přijímačů zakoupených v jiných zemích mohou nastat potíže způsobené odlišnostmi v kmitočtu a normě televizního signálu mezi počítačem a televizorem. V tomto případě vám doporučujeme obrátit se na vaši obchodní nebo servisní organizaci.

Lze použít místo televizního přijímače monitor?

Ano, vaše obchodní organizace může zajistit monitory, které ve spojení počítačem Spectrum dávají vynikající výsledky.

K čemu je potřeba síťový zdroj?

Spectrum vyžaduje napájecí napětí 9 V/ 1,4 A. Síťový zdroj dodává toto napětí ze sítě 240 V/50 Hz.

Vyvíjí počítač Spectrum rušivá elektromagnetická pole?

Při provozu rozhlasového přijímače v blízkosti počítače může dojít ke vzniku rušivých interferencí, které však nepoškodí ani přijímač ani Spectrum.

Mohu použít paměť ZX 16 K RAM?

Ne. Tato paměť RAM je použitelná pouze pro počítače ZX 81.

## Oživení vašeho počítače ZX Spectrum+

Na konec kabelu napáječe připojte síťovou zástrčku. Síť by měla být jištěna pojistkou 3 A. Počítač Spektrum nevyžaduje uzemnění, ani tehdy je-li připojen na třívodičovou síť (nulovací kolík v zásuvce zůstává nezapojen).

Dále následuje řada ilustrací, které ukazují, jak připojit počítač k síti a televizoru. Jakmile budete s touto prací hotovi, obraťte na další stránku, kde najdete instrukce, jak naladit televizor.

Popis obrázků na straně 5:

1. Připojte antenní kabel do zdířky označené TV na vašem Spectru. Do této zdířky lze zasunout pouze jeden z konců antenního kabelu.
2. Odpojte anténu od televizoru. Připojte do antenní zdířky druhý konec antenního kabelu od Spectra.
3. Zapněte televizní přijímač a nastavte regulátor hlasitosti zvuku na minimum.
4. Zasuňte malou zdířku síťového napáječe do zásuvky označené 9 V DC na počítači.
5. Zasuňte síťovou zástrčku do zásuvky na zdi a v případě, že je opatřena vypínačem, zapněte jej. Počítač Spectrum nemá vlastní vypínač.

## Zdířky a konektory počítače Spectrum

Zdířka napájení-9V stejnosměrný zdroj, napájení obstarává zdroj ZX, který je připojen do této zdířky.

Přímý konektor - sběrnice pro připojení vnějších periférií, t.j. široká škála hardware včetně microdrive, tiskárny a modemů.

Mikrofonní zdířka-MIC - slouží k propojení mikrofonního vstupu magnetofonu s počítačem při záznamu programů na kazetu.

Sluchátková zdířka-EAR - do této zdířky se přivádí signál z



kazetového magnetofonu při vkládání programů do počítače.  
Televizní zdířka-TV - umožňuje televiznímu přijímači příjem  
signálu vysílaného počítačem Spectrum.

## Vyladění vašeho televizoru

Počítač Spectrum vysílá televizní signál na kmitočtu kanálu 36 v pásmu UHF, takže váš televizní přijímač musí být nalaďen na tento kanál. Pak může přijímat signál z počítače Spectrum.

Jestliže jste propojili počítač s televizorem a oba přístroje jsou zapnuté, nalaďte televizor tak, aby se na obrazovce objevil ve spodní části řádek s nápisem " c 1982 Sinclair Research Ltd ". Pak můžete vyzkoušet barvy Spectra a zahájit počítání. Pokud neobdržíte výše uvedený nápis nebo barvy, postupujte podle diagramu na následující straně.

Plynulé ladění - umožňuje volbu jakéhokoliv kanálu. Otáčejte ladícím prvkem tak dlouho, pokud nezískáte na obrazovce zprávu o copyrightu.

Ladění s předvolbou - zvolte tlačítko, které bude použito pro příjem signálu z počítače a odpovídajícím ladícím prvkem nalaďte přijímač tak, abyste obdrželi zprávu o copyrightu. Doporučujeme použít tlačítko volné, na kterém není nalaďen žádný televizní vysílač. Nebudete muset ladit přijímač pokaždé, když budete chtít počítat na Spektu - viz obr. na straně 6.

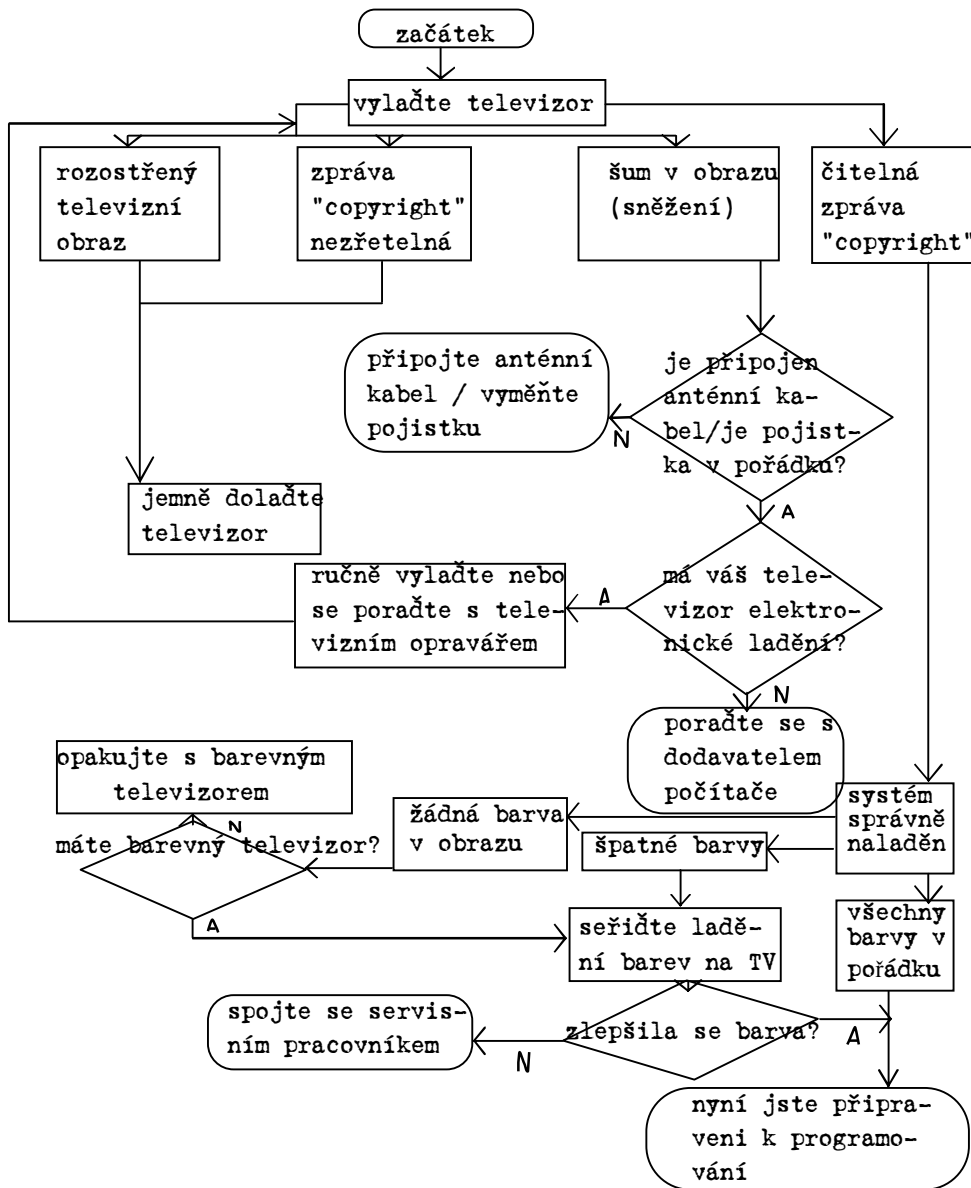
Elektronické ladění - televizory s tímto systémem ladění se samy naladí na požadovaný kanál. Tyto přístroje, které nemají možnost ručního ladění nemusejí být vždy vhodné pro spolupráci s počítačem Spectrum.

Jak vyzkoušet barvy počítače Spectrum:

Vyzkoušení provedete stisknutím klávesy B a následně čísla od 1 do 6. Řádek s nápisem copyright zmizí, nejprve se objeví slovo BORDER a poté číslo. Nyní stiskněte klávesu ENTER.

"Okrajová" plocha obrazovky musí změnit barvu na barvu odpovídající číslu, které následovalo po příkazu B (BORDER).  
Obrázky na straně 6 ukazují správnou reakci počítače na příkazy BORDER 3 , ENTER a BORDER 4 , ENTER. Příkaz BORDER 7 vrátí okraj do původní bílé barvy.

SCHEMA ODSTRAŇOVÁNÍ ZÁVAD



## Co dokáže váš počítač ZX Spectrum+

### První pokus

Nyní, když je počítač zapojen a televizní přijímač naladěn, stiskněte několik kláves. Uvidíte slova a písmena, jež se objeví na televizní obrazovce, možná i některá čísla.

Pokud neumíte programovat, může se stát, že na vaše stisknutí některé klávesy počítač neodpoví. Neobávejte se, nic zlého se počítači nestalo, stisknutím libovolné klávesy nemůžete počítači ublížit.

Nyní stiskněte tlačítko RESET na levé straně počítače, Spectrum bude připraveno pro další práci.

Na několika následujících stranách vidíte příklady hry s televizní obrazovkou - co počítač dokáže.

### Jak dále programovat váš počítač.

Váš počítač Spectrum dokáže velmi mnoho. Aby mohl správně plnit vaše příkazy, musíte mu zadat správný soubor instrukcí, tzv. počítačový program. Nabízíme vám několik krátkých programů, které prověří schopnosti vašeho počítače. Vše, co v této chvíli zbývá ještě udělat, je vložit program přesně tak, jak je uvedeno v ukázkách. Odstavec s názvem "How to alter a program" - jak měnit program - vám ukáže, jak experimentovat, kdy současně začíná nový program (viz obr. na str. 11). Strany 74 a 76-78 mohou pomoci při řešení potíží.

### Jak zakládat a spouštět program

Každý soubor instrukcí je uveden ve výčtu, kterému se říká "listing" (výpis programu). Uvidíte, že výpisy programů obsahují několik částí, které vždy začínají číslem (10, 20, atd.). To jsou tak zvané řádky programu, a to i tehdy, jestliže zabírají dva řádky výpisu na obrazovce televizoru a obsahuje jednu nebo více instrukcí pro počítač.

V každém programovém řádku se nacházejí slova a zkratky obsahující dvě nebo více písmen, jako např. PRINT, LET, RND, PI, PAPER a GOTO. Těmto slovům se říká klíčová slova a nemůžete je vypisovat písmeno po písmenu. Místo toho najdete slovo na klávesnici (např. slovo PRINT je na klávese P). Postup vkládání programu do počítače dodržujte přesně podle uvedených příkladů (ukázek). Vložené příkazy se vypisují na obrazovce. Dostanete-li se na konec programové řádky, stiskněte klávesu ENTER. Nyní se řádek přemístí z dolní části obrazovky do horní. Tímto způsobem vkládejte každý další řádek. Stisknete-li špatnou klávesu, obraťte se na odstavec s názvem "Jak opravovat chyby" na následující straně. Založili jste-li všechny programové řádky do paměti počítače, stiskněte klávesu označenou R. Nyní se objeví klíčové slovo RUN. Následným stisknutím klávesy ENTER přikročí počítač k provádění programu.

str. 8 originálu

Způsob používání klávesnice

- Založit jakékoli slovo nebo písmeno - nejprve si zapamatujte jeho polohu na klávesnici. Poté použijte následující postup:
- horní klávesy - stiskněte EXTEND MODE, pak příslušnou klávesu
  - spodní klávesy (znaky) - stiskněte EXTEND MODE, podržte klávesu se symbolem SHIFT, pak příslušnou klávesu
  - horní klávesy (zdvižená část) - stiskněte klávesu
  - písmena nebo čísla (zdvižená část) - stiskněte klávesu, podržte CAPS SHIFT pro založení velkých písmen
  - spodní klíčová slova nebo znaky (zdvižená část) - stiskněte symbol SHIFT a stiskněte klávesu

Veškeré další detaily práce s klávesnicí najdete na straně 20-21.

## Jména

(viz popis na straně 8 originálu - žlutý)

Po stisknutí kláves v tomto poradí se objeví jméno ZX Spectrum+ v mnoha barvách po celé obrazovce. Poté počítač zastaví a vypíše "scroll?" na spodní části obrazovky. Pro posunutí obrazu o další "stránku" stiskněte libovolnou klávesu kromě N, SPACE, BREAK nebo STOP. Zastavíte-li posouvání a stisknete-li BREAK, poté R (RUN) a dále ENTER, jména se objeví v odlišném barevném provedení.

Vyzkoušejte toto:

V lince 30 změňte ZX Spectrum + na vaše jméno (")

např. 30 PRINT "John";

Zapamatujte si, že je potřebné zapsat (;). Uvidíte, že vaše jméno se pak objeví na celé obrazovce.

## Jak měnit program

(viz popis na straně 9 originálu - červený)

Počkejte, dokud program neskončí nebo nezastaví pomocí stisknutí klávesy BREAK. Poté stiskněte V(CLS), poté ENTER a poté K(LIST), poté ENTER. Výpis programu (výpis řádek) se objeví na obrazovce.

Prohlédněte si, který řádek chcete změnit, pak vytiskněte celý nový řádek, včetně čísla řádku, a příkaz zakončete stisknutím klávesy ENTER. Nový řádek nahradí starý. Stiskněte R(RUN) a ENTER a nový program začne.

## Vzory

(popis na straně 9 originálu - žlutý vlevo)

Po vytisknutí uvedeného programu se objeví následující obrázek na obrazovce.

Geometrický vzor začne postupně naplňovat obrazovku.

Po zaplnění obrazovky se program zastaví s otázkou "scroll?" (posuv?).

Další pokračování výpisu na obrazovce nastane po stisknutí libovolné klávesy (kromě N, SPACE, BREAK, STOP). Stisknete-li klávesu N, poté BREAK, R (RUN) a ENTER, zastavíte program.

Vyzkoušejte nyní toto:

V řádce 20 nahraďte číslo 7. Na obrazovce se objeví nový vzor. Vyzkoušejte například číslo 8.

Blikavé kruhy

(viz popis na straně 9 originálu - žlutý vpravo)

Po spuštění tohoto programu se na obrazovce objeví soustředěné barevné blikající kruhy. Barevně bliká i okraj a počítač vydává trylkující zvuk.

Vyzkoušejte toto:

Před výpisem programu (použitím klávesy K) stiskněte klávesu PAPER 7 a pak ENTER. Vypište řádek 20 bez dvojice klíčových slov FLASH RND. Kruhy nebudou dále blikat.



### Šílená (roztroušená) mozaika

(viz popis na straně 10 originálu - žlutý nahoře vlevo)  
Po vložení programu do počítače podle uvedeného návodu se celá obrazovka pokryje barevnými čtverci, které tvoří barevný vzorec. Kdykoli znovu spustíte program, vytvoří se jiný vzor.

Vyzkoušejte toto:

V řádku 50 změňte 143 na 142 a uvidíte hvězdy! Zkuste další čísla od 33 do 142. Podívejte se na str.51 originálu - seřazení typů, a uvidíte, co se stane.

### Jak znovu spustit program

(viz popis na straně 10 originálu - fialový vlevo dole)  
Některé z těchto programů - jako Hvězdy a Kruhy - se zastavují a na konci hlásí zprávu O OK, a číslo posledního řádku v programu. To znamená, že celý program byl ukončen. Pro opětovné spuštění jednoduše stiskněte R(RUN) a ENTER.

Další programy buď cyklují, jako např. Šílená mozaika, nebo automaticky opět začínají "Žhavý východ slunce".

Pro zastavení těchto programů stiskněte klávesu BREAK. Podržte tuto klávesu, dokud se program nezastaví a nevypíše se zpráva BREAK. Pro opětovné spuštění stiskněte R a ENTER.

### Jak opravit chyby

(viz popis na straně 10 originálu - fialový vpravo nahoře)  
V případě, že stisknete omylem špatnou klávesu nebo zapomenete stisknout SHIFT nebo EXTEND MODE, můžete chybu smazat pomocí klávesy DELETE. Poslední klíčové slovo, písmeno, znaménko nebo číslice zmizí. Podržíte-li DELETE, bude se umazávat program dále.

Jestliže jste udělali chybu v řádku a pak stiskli klávesu ENTER, bude u místa, kde je chyba blikat symbol "?". Celý tento řádek zrušíte i s chybou příkazem DELETE. Pak vypíšete celý řádek znovu a stiskněte ENTER.

Pokud vložíte do počítače chybný řádek a neprovedete opravu, program nebude možné spustit, objeví se výpis zprávy, kterou vám Spectrum oznámí počet chybných řádků. Tyto chybné řádky musíte znovu správně vypsat, pak stiskněte ENTER, R a znovu ENTER. Nyní program musí pracovat.

Polyedr (mnohostěn)

(viz popis na straně 10 originálu - žlutý vlevo dole)  
Nejprve bude obrazovka čistá. Stisknete-li klávesu s číslem 6 a ENTER, objeví se šestiúhelník. Znovu spusťte program a zadejte jiné číslo. Uvidíte obrazec s jiným počtem stran.

Vyzkoušejte toto:

V řádku 20 zaměňte č. 2 na jiné číslo. Vzor bude vytvářen rychleji, je-li číslo vyšší, a mnohostěny (polyedry) jsou více vzdáleny.

### Hvězdy a pruhy

(viz popis na straně 11 originálu - žlutý vlevo nahoře)  
Po spuštění programu se na obrazovce objeví vlajka Spojených států.

Vyzkoušejte toto:

Změňte čísla udávající barvu vlajky. Barva pruhů je na řádku 10, hvězdy na ř. 120 a pozadí hvězd na ř. 110.

### Jak začít nový program

(viz popis na straně 10 originálu - fialový vlevo dole)  
Dokončili-li jste program a chcete-li zadat program nový, počkejte do konce nebo jej zastavte tlačítkem BREAK.  
Pak máte volbu dvou způsobů, jak vymazat starý program z paměti počítače. Buď stlačením dvou kláves A(NEW) a ENTER. Obrazovka v tomto případě zčerná a objeví se řádek s nápisem "copyright". Druhá, jednodušší možnost, je stisknout tlačítko pro RESET, což má stejný účinek, jako vypnutí a opětovné zapnutí počítače.

### Východ slunce

(viz popis na straně 11 originálu - žlutý vpravo nahoře)  
Po založení programu se na obrazovce objeví "Východ slunce" v různých barvách, které se mění každých 5 sekund. Pokud nevidíte na obrazovce žádný obraz, čekejte, nový "Východ slunce" se záhy objeví.

Vyzkoušejte toto:

V ř. 210 změňte č. 200 na jiné číslo, v souladu s tím se změní doba východu každého slunce. Číslo 200 odpovídají 4 sekundy.

Co dále?

Máte následující možnosti:

Chcete-li uchovat některý z předchozích programů, abyste jej mohli opětovně vyvolat, můžete jej zaznamenat na kasetový magnetofon. Otočte na str. 38, kde je uvedeno, jak uchovat vlastní programy.

Chcete-li se zabývat programováním, experimentováním s vaším Spektrem, najdete poučení o programování v kapitole 2 (Zahájení programování). Zatím jste pouze vyzkoušeli programy bez nezbytného pochopení, jak pracují. Kapitola 2 objasní způsob programování počítače Spectrum.

Chcete-li vyzkoušet některé programy zaznamenané na kazetě jako například počítačové hry, které jste si koupili, nalistujte stránku 14 (Jak nahrávat programy).

## Používání připraveného software

Při zadávání programu do Spectra, převádí se stisknutí kláves na elektrické signály. Tyto kódy přicházejí do paměti, která je uchovává tak, aby je počítač v průběhu procházení programu mohl použít. Kódy zůstávají v paměti, dokud je nesmažete (klávesou NEW nebo RESET) nebo neodpojíte Spectrum od napájení. Není samozřejmě nezbytné vždy program vkládat do počítače vypsáním všech řádků na klávesnici, jako tomu bylo při prvním vložení. Místo toho lze zakoupit předem připravené programy (software), které můžete přímo automaticky vložit z kazety do počítače. Použití těchto připravených programů nejen spoří váš čas a ušetří vás problémů při vkládání programu klávesnicí pokaždé, když budete chtít pracovat na počítači ale umožní vám také vlastnit knihovnu programů, která by při normálním způsobu opisování zabrala dny a týdny psaní. Výrobci software vyrábějí programy všech druhů. Připravili je pro vás nejlepší programátoři. Pro Spectrum existuje široká škála těchto programů. Využijte katalogu Sinclair Spectrum software, který vám poskytne informace o druzích programů využitelných vaším počítačem. Takovéto programy jsou pro vás připraveny a jsou pohotově schopny okamžitého spuštění.

Jak se přepisují programy z kazety do počítače

Kódované signály na softwarových páskách sestávají z vysokých a nízkých "pípání", zaznamenaných v poměru přibližně 1500 za sekundu. Přehráváte-li softwarový pásek na kazetovém magnetofonu, slyšíte tento kód, který tvoří program. Stačí pouze připojit kazetový magnetofon k počítači a kódy se zaznamenávají do paměti Spectra. To je tzv. plnění počítače programem. Na stranách 12 a 13 je ukázáno, jak připojit magnetofon k počítači.

Strany 14 a 15 ukazují, jak využít spolupráce počítače s magnetofonem.

## Softwarové otázky a odpovědi

Co znamená software?

Software je všeobecný pojem používaný pro programy, které se vkládají do počítačů, aby pracovaly.

Hardware je termín pro vlastní fyzické vybavení - samotné počítače a další zařízení určené a využívané ve výpočetnictví.

Proč je software vydáván na kazetových páskách?

Kazetové pásky jsou snadno použitelné a nevyžadují zvláštní vybavení. Levné kazetové přehrávače zcela stačí pro zakládání programů tohoto druhu.

Jak zní programy na magnetofonových páscích?

Přehrajte si jednu kazetu bez připojení do počítače Spectrum. Uslyšíte skřípot vysokých tónů. To je způsobeno kódovými signály, které procházejí reproduktorem přehrávače místo do počítače. Signály jsou vysílány z kazety do počítače takovou rychlostí, že je nemožné rozlišit jednotlivé zvuky.

Jsou ještě další druhy softwaru?

Ano. Můžete získat programy na **destičkách** ROM místo na kazetových páscích. Tyto paměti ROM se zasunou do **spojky interface**, která se **nachází** na zadní straně počítače. Programy zapsané na ROM se do počítače dostávají nepřetržitě bez jakéhokoli čekacího času.

Software je rovněž možno získat na mikrodrivech, které obsahují programy zaznamenané na magnetických médiích obdobných kazetovým páskům. Několik programů může být zaznamenáno na jednom tomto médiu a na rozdíl od kazetových pásků může být jakýmkoli program vložen do počítače v rozmezí sekund místo minut, které zabere program na kazetách. Média microdrive jsou používána spolu s jednotkami microdrive (viz str 46 originálu)

Který kazetový přehrávač je nejlepší?

Počítač Spectrum se spokojí s levným kazetovým přehrávačem, který je lepší připojit do sítě než na bateriový pohon. Přehrávač musí mít vlastní ovládání hlasitosti, avšak tónová clona není podstatná. Zvláštní počítačové kazetové přehrávače lze rovněž využít - jsou konstruovány pro uschovávání a zakládání programů přesněji než běžné přístroje. Kazetové magnetofony typu "deck", které jsou součástí zvukové aparatury není snadné připojit. Navíc jejich audiovýstup nevydává dostatečně silné signály pro počítač.

Vyžadují programy nahrané na páscích zvláštní pozornost?

Jako jakákoli forma magnetického záznamu i programy na kazetách mohou být narušeny silnými magnetickými poli. Tzn., že není vhodné skladovat kazety v blízkosti jakéhokoli silného elektromagnetického pole. Kazety software rovněž vyžadují uschování v bezprašném prostředí.

Pracuje jakýkoli druh software?

Ne. Pouze software určené pro ZX Spectrum nebo ZX Spectrum+ je možné použít.

## Jak použít váš kazetový přehrávač?

Kazetová přívodní šňůra dodaná s počítačem je určena pro propojení kazetového přehrávače s počítačem. Je to vodič s dvojicí malých zástrček, na každém konci. Umístěte kazetový přehrávač do blízkosti počítače a zasuňte kabel podle vyobrazení. Kazetový přehrávač i počítač mohou být zapnuty nebo vypnuty při provádění tohoto úkonu, avšak je lepší vyjmout kazetu z přehrávače dříve, než jej zapnete nebo vypnete. Tím je zaručena bezpečnost programů, které jsou na kazetě nahrány.

## Provádění správných připojení

1. obr. : Zasuňte jakýkoli ze čtyř konektorů do zdičky pro sluchátko na zadní straně počítače.
2. obr. : Zasuňte druhou zástrčku stejné barvy do zdičky pro sluchátko na kazetovém přehrávači, má-li tuto zdičku.
3. obr. : Nemá-li kazetový přehrávač zdičku pro sluchátko, připojte jej do obdobné zdičky (EAR nebo headphone). Pokud zde taková zdička není, vyzkoušejte zapojení na vnější zdičku pro reproduktor.
4. obr. : Nelze-li zasunout kabel do zdičky v kazetovém přehrávači, potřebujete adaptér nebo zvláštní kabel se správnými zdičkami. Kabel Spectrum EAR vyžaduje 3,5 mm jack a vstupní signál přibližně 1 Volt.

## Zdičky EAR a MIC:

Při zakládání programu můžete mít propojeny zdičky EAR i MIC jak je ukázáno na obrázku. Avšak zaznamenáváte-li program (viz str. 38), musíte odpojit kabel EAR.

## Konce pro software

Kabel pro kazetový přehrávač Spectrum má barevná zakončení pro zabránění překřížení mezi zdičkami na počítači a kazetovém přehrávači. Používáte-li kazetový přehrávač, vždy se snažte používat stejný systém s jednou barvou pro zdičku EAR a druhou pro zdičku MIC.



Některé kazetové přehrávače mohou být narušovány jiným elektrickým vybavením které je v blízkosti. Některé tyto vlivy mohou narušit signály vysílané mezi počítačem a kazetovým přehrávačem s tím, že programy se správně nenahrají. Připadá-li vám, že váš kazetový přehrávač správně nefunguje, vyzkoušejte, aby neležel vedle počítače nebo televizoru.

### Jak zakládat program

Nyní jste propojili kazetový přehrávač s počítačem, tím jste připraveni založit a spustit program. Můžete využít připravené softwarové pásky nebo vlastní programy zaznamenané na kazetách. Postup je stejný v obou případech.

Zapněte kazetový magnetofon. Přesvědčte se, že počítač je rovněž zapnut. Pak vložte kazetu do kazetového přehrávače. Je-li již program v počítači, počkejte do jeho konce nebo jej zastavte stisknutím klávesy BREAK. Nyní můžete vytisknout NEW nebo stisknout tlačítko RESET, čímž odstraníte program z paměti počítače. To však není podstatné, neboť zakládání nového programu nejdříve vymaže paměť. Je důležité si zapamatovat, že při zakládání nového programu bude předchozí program z paměti vymazán. Nyní postupujte podle očíslovaných instrukcí. Pokud cokoli nefunguje, obraťte se na str. originálu - graf s vyznačením odstraňování závad.

1. Vložte kazetu a převiňte ji na začátek.
2. Nastavte hlasitost a tónovou clonu na kazetovém přehrávači do požadované úrovně. Vyzkoušejte hlasitost přibližně na 2/3 maxima a máte-li na přístroji tónovou clonu, nastavte ji na nejvyšší výšky.
3. Stiskněte J a LOAD - objeví se na obrazovce. Pak zaveďte programové jméno v čísle - např.: LOAD "Prog 1" - viz. obr. na str. 14
4. Stiskněte ENTER - obrazovka bude slepá.
5. Spusťte pásek. Hranice okraje obrazovky budou červené nebo modré nebo budou blikat červeně a modře. To značí, že počítač vyhledává program.
6. Po několika sekundách se objeví na okrajích červené a modré pruhy, které se budou pohybovat nahoru nebo dolů po okrajích. To značí, že počítač začal přijímat signál.

7. Na obrazovce se objeví buď slovo Program: následuje jméno programu, nebo slovo Bytes: následuje jméno nebo písmeno. To označuje, že počítač skutečně našel program.
8. Červené a modré pruhy se opět objeví, když počítač čeká na plnění programem.
9. Vzor žlutých a bílých čar se objeví na okrajích. To značí, že počítač zakládá program. Zakládání může trvat několik minut, je-li program dlouhý.
10. Operace 7, 8, 9 se mohou opakovat jednou nebo několikrát, je-li program rozdělen do částí.
11. Program může být spuštěn automaticky, když byl založen. Nezapomeňte zastavit pásek.
12. Nezačne-li program automaticky poté, co je založen, obrazovka zbledlá a předá správu O OK,0:1, zastavte pásek. Viz obr. na str. 15 vpravo nahoře.
13. Stiskněte klávesu RUN a ENTER a program nyní začne.

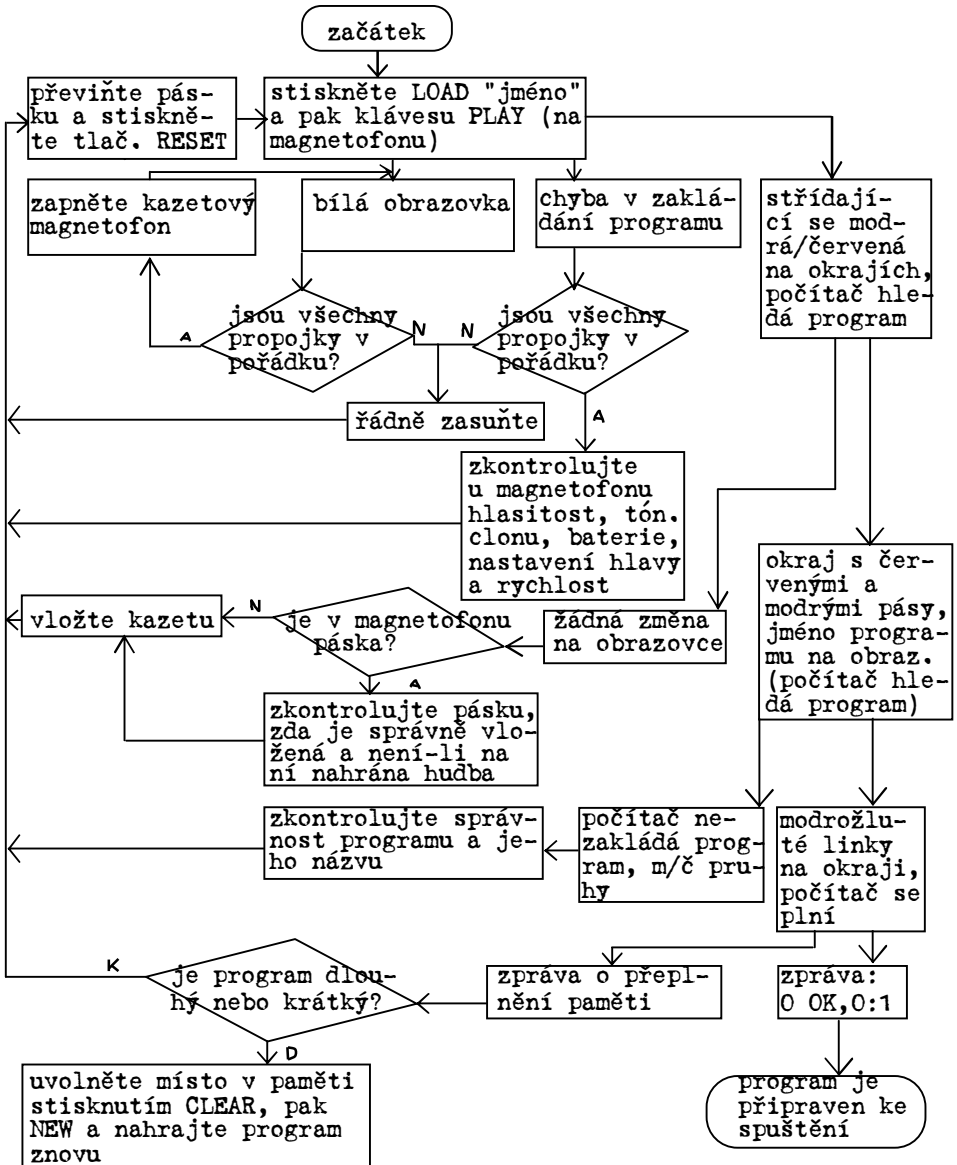
Pomoc při zakládání software  
(str. 14 a 15 originálu dole)

Předáváme vám několik instrukcí, které vám pomohou uspořít čas při zakládání programu.

1. Označte si nálepkami zřetelně veškeré pásy, abyste mohli snadno nalézt příslušné programy. Obsahuje-li páska více než jeden program, napište si jména programů na nálepku. Nezapomeňte na přesné hláskování programu, neboť počítač bude toto hláskování potřebovat.
2. Má-li váš kazetový magnetofon počítadlo, použijte jej k rychlému nalezení programu na dané pásce, je-li na jedné straně více než jeden program. Vynulujte počítadlo na začátku pásky, stiskněte LOAD následované jménem programu (v uvozovkách), které není jménem na pásku. Přehrajte pásku a počítač vám ukáže každé jméno programu, které nalezne bez toho, aby jej zakládal. Napište číslo počítadla na nálepku se jménem programu. Tím si usnadníte nalezení žádaného programu.

3. Je-li páska na žádaném programu nebo nepotřebujete-li znát jméno programu, stiskněte LOAD "", namísto LOAD následovaného jménem programu v uvozovkách. Mezi dvěma uvozovkami nesmí být prostor. Počítač Spectrum pak založí první program, který nalezne na pásce. Je-li jméno programu, které se objeví na obrazovce, jiné než požadované, stiskněte klávesu BREAK, převiňte a pokuste se znova.
4. Poznamenejte si úroveň hlasitosti a tónu, což usnadní ukládání do paměti. Nastavte tyto úrovně na přehrávači vždy, když zakládáte program.

Odstraňování závad při zakládání softwaru.



## 2. Zahájení programování

Tato kapitola je úvodem k programování na počítači ZX Spectrum+. Řekne vám, jak využívat počítač Spectrum tím, že vám ukáže, jak naleznete vlastní způsob přístupu ke klávesnici. Uvidíte, jak můžete provádět vlastní práce na počítači Spectrum. Krátké programy pro prozkoušení, které jsou zde koncentrovány na vyzkoušení zvláštních rysů počítače Spectrum, jsou takové, že budete-li psát své vlastní programy podobným způsobem, využijete téměř veškeré schopnosti počítače.

## Klávesnice - ovládací deska počítače

ZX Spectrum+ má vlastní počítačový jazyk, kterému se říká Basic. Aby se počítač řídil vašimi příkazy, musíte programovat tak, že k němu promlouváte v tomto programovacím jazyce. To se provádí pomocí klávesnice. Klávesnice navíc umožňuje řídit počítač i při zpracovávání programu.

Verze Basic, které počítač Spectrum rozumí, je jednoduchá. Je konstruována na základě především anglických slov, aby bylo snadné ji použít. Počítač má navíc velkou výhodu usnadňující programování. Je to tzv. jednoklíčový vstupní systém klíčových slov.

### Klíče a klíčová slova

Klíčová slova jsou zvláštní slova v programové řeči Basic, které dávají počítači příkaz k provedení nějaké operace - slova jako PRINT a INPUT. U většiny počítačů musíte klávesami vypsát každé klíčové slovo, jako musíte i na psacím stroji, a každé slovo musíte hláskovat. Avšak u počítače Spectrum jednoduše stisknete klávesu, čímž získáte celé klíčové slovo na obrazovce.

Sinclair Basic má přes 80 klíčových slov, která jsou přístupná ze všech 36 kláves (26 kláves písmenových a 10 číselných). S ohledem na to, že počítač používá tak vysoký rozsah instrukcí v jazyce Basic, mnoho kláves není určeno pouze pro jedno, ale pro několik klíčových slov, které počítač rozpoznává. Většina kláves udává klíčová slova stejně tak jako písmena, čísla, znaménka nebo formu (charakter grafiky), které všechny mohou být využity pro program.

### Volba klíčových slov a znaků

Na klávesnici počítače Spectrum jsou dva klíče, které budete nejčastěji používat. Tyto jsou EXTEND MODE a SYMBOL SHIFT - to jsou klíče, které umožní zvolit, které klíčové slovo

a znak na dané klávese chcete zobrazit na obrazovce z několika významů dané klávesy. Nyní jste získali povšechný úvod do rozpisu kláves a poté, co se důvěrně seznámíte s rozložením klávesnice, ukážou další dvě stránky přesně, jak zvolit, co se má objevit na klávesnici počítače. Jakmile budete toto umět, můžete začít psát vlastní programy.

Popis kláves na klávesnici

(str. 18-19 originálu)

#### GRAPH

graf - tento klíč se používá pro volbu tvarů nebo grafiky nebo typu u kláves 1-8. Stisknete-li tuto klávesu a pak číslo klávesy s CAPS nebo SHIFT, na obrazovce se objeví grafický typ.

#### TRUE VIDEO a INV VIDEO

- tyto klávesy vkládají řídicí kódy do programových řádků, čímž vyvíjí normální nebo inverzní barvy.

#### NEW

- tato klávesa čistí paměťovou oblast počítače v jazyce Basic, vymazává jakýkoli program, který se v paměti nachází.

#### DELETE

- tato klávesa je užívána při chybném stisknutí jakékoli klávesy - pokud chcete chybu odstranit, případně odstraňujete-li vadné klíčové slovo, písmeno, číslo nebo znaménko - viz str. 10 originálu

#### EDIT

- tato klávesa je používána pro změnu řádku v programu bez celkového přepisování - viz str. 21 originálu.



#### EXTEND MODE

- tato klávesa volí horní klíčové slovo nad zvýšenou částí klíče. Je-li následována klávesou SYMBOL SHIFT a klávesou, volí znaménko nebo klíčové slovo, které je bezprostředně nad zvýšenou částí klíče - viz str. 20-21 originálu

#### CAPS SHIFT

- při stisknutí této klávesy současně s klávesou velkého písmena dostanete velké písmeno. Požadujete-li větší počet písmen zobrazených jako velká, použijete klávesu CAPS LOCK

#### CAPS LOCK

- tato klávesa se užívá, chcete-li vypsát klávesy, které dávají velká písmena po celou dobu. Stisknete-li ji opět, získáte malá písmena.

#### BEEP

- tato klávesa udává klíčové slovo, které řídí zvukový syntezátor počítače Spectrum

#### Space bar

- tato klávesa udává mezery stejně tak, jako mezerník na psacím stroji

#### Směrové šipky

- stisknutím těchto klíčů získáte pohyb ve stejném směru, který šipky ukazují. Tyto klíče se používají často u programů, kdy řídíme pohyb forem na obrazovce. Rovněž se užívají při vydávání programu.

#### Klávesy pro určení barev

- 1-7 - těchto 6 kláves dává klíčová slova, která řídí barvy na obrazovce

### Číselník

- obdobně jako při dávání čísel mohou tyto klávesy udávat vstupní kódy do programů pro vykázané barvy - viz str. 33 originálu. Klíčová slova, která jsou bezprostředně nad klávesami od 4 do 0 s výjimkou klávesy 8, jsou užívána pouze pro Microdrive ZX.

### BREAK

- tato klávesa zastavuje program. Nemaže program z paměti počítače.

### ENTER

- tato klávesa udává počítači povel, že informace, která byla dodána, je kompletní a že se může uskutečnit.

### SYMBOL SHIFT

- při stisknutí této klávesy a současném stisknutí čísla nebo písmena zvolíte spodní klíčové slovo nebo znak na zvýšené části klávesy. Je-li užita tato klávesa po stisknutí klávesy EXTEND MODE, volí symbol nebo klíčové slovo bezprostředně nad zvýšenou částí - viz str. 20-21 originálu.

## Jak pracovat s klávesami

Z většiny kláves počítače ZX Spectrum+ můžete získat šest samostatných klíčových slov, písmen, číslic nebo znaků. Přesto však volba typu nebo klíčového slova na klávesnici není komplikovaná, jakmile se důvěrně seznámíte s jedním ze zvláštních rysů tohoto počítače. Stisknete-li klávesu, výsledek, který se objeví na obrazovce, závisí na "modu", ve kterém je počítač v dané chvíli. Odlišné mody vám umožní klíčovat různé typy i funkce, jako jsou klíčová slova, písmena nebo grafické typy. Výhoda této úpravy spočívá v tom, že jak stisknete klávesnici, Spectrum skutečně pomáhá při volbě modu kláves tak, že vkládáte instrukce a informace ve správném pořadí. Na těchto dvou stranách naleznete přesně to, co který typ modu dokáže a provádí.

### Mod klíčových slov

Zapněte nebo znovu nastavte počítač tak, aby na obrazovce byla zpráva copyright. Nyní stiskněte klávesu ENTER. Objeví se blikající K ve spodním levém rohu. Blikající čtverec je nazýván cursor. Ukazuje, kde bude další znak vypsán a K značí, že počítač je v modu klíčových slov. Stiskněte jakékoli písmeno a horní klíčové slovo na zvýšené části klávesnice se objeví na obrazovce. Zkuste např. Q - objeví se klíčové slovo PLOT. Stiskněte klávesu DELETE, čímž odstraníte klíčové slovo, a zkuste další klíče. Čísla klíčů vám udají čísla, avšak jakmile stisknete klávesu písmene, na obrazovce se opět objeví klíčové slovo ze zvýšené části. Použijte opět klávesu DELETE, takže se opět objeví K. Nyní stiskněte klávesu SYMBOL SHIFT, podržte ji a stiskněte jakékoli písmeno. Tentokrát se objeví klíčové slovo nebo znak právě nad písmenem ve zdvižené sekci. S klávesou číslice se objeví znak napravo ve zdvižené části. Modus klíčových slov se tedy vztahuje ke zvýšené části kláves.

### Modus písmen a velkých písmen

Jakmile bylo vydáno klíčové slovo nebo znak v klíčovém modu se počítač automaticky změní - cursor na L. Nyní je v písme- novém modu. Stiskněte jakoukoli klávesu a objeví se malé písmeno, stiskněte číslici a objeví se číslo. Chcete-li získat velké písmeno, stiskněte klávesu CAPS SHIFT a pak klávesu písmene.

Chcete-li psát všechna písmena velká, stiskněte CAPS LOCK. Cursor se změní na C. Nyní je počítač v modu velkých písmen, vždy když stisknete na klávesnici písmeno, vytisknete velké písmeno. Stále budete i v tomto modu získávat po stisknutí klávesy čísla číslici. Pro návrat do modu písmen opět stiskněte CAPS LOCK.

### Rozšířený modus

Další modus je nazýván rozšířený a vytváří se stisknutím klávesy EXTEND MODE. Cursor se nyní změní na E. Stiskněte jakékoliv písmeno a horní klíčové slovo z páru klíčových slov nad zvýšenou částí se objeví na obrazovce. Např. stisknete-li B, získáte BIN. Pro získání spodního klíčové- ho slova nebo znaku nad zvýšenou částí stiskněte nejprve klávesu SYMBOL SHIFT, podržte ji a poté stiskněte klávesu písmene. U klávesy B tímto získáte např. klíčové slovo BRIGHT. Takto pomocí rozšířeného modu získáte dvojici klí- čových slov nad zvýšenou částí kláves. Po stisknutí klíče (nebo EXTEND MODE) v rozšířeném modu se počítač automaticky vrací na modus písmen nebo velkých písmen.

### Grafický modus

Pátý modus je nazýván grafický a je dán stisknutím klávesy Graph. Cursor se změní na G. Stiskněte klávesu 1-8 a na obrazovce budou zobrazeny grafické typy, které jsou na těch- to klávesách zobrazeny. Nyní stiskněte CAPS SHIFT a jakékoli číslo 1-8. Nyní se objeví grafické typy, avšak tentokrát černobíle a převrácené. Při vypouštění grafického modu musíte vždy opět stisknout klávesu GRAPH, neboť počítač tento modus neopouští automaticky.

Strana 20 a 21 originálu, dole

Jak zvolit klíčové slovo, symbol nebo typ

Můžete zvolit libovolné klíčové slovo, písmeno, znaménko , typ nebo číslici. Při volbě funkce klávesy věnujte pozornost umístění daného znaku a dle toho rozhodněte, kterou ze dvou kláves (případně tří) a v jakém pořadí použijete, abyste se dostali do potřebného módu. Stále sledujte kurzor na obrazovce, abyste zaznamenali ve kterém módu počítač právě pracuje.

Opravy (editace) na počítači Spectrum

Vložíte-li povel nebo přistoupíte k výpisu programu, může se stát, že budete chtít opravit omyly v povelích nebo vypustit či změnit řádky v programu. Tuto "editaci" můžete snadno provádět pomocí výpisu.

Jak opravit omyl:

Podarí-li se vám dát počítači povel nebo napsat celý řádek, který je chybný, neodpovídá Basicu, Spectrum to ohlásí blikající ? před chybou. Pro opravu omylu podržte levou nebo pravou klávesu pro pohyb kurzoru tak dlouho až se kurzor přemísť k chybě. Poté klávesou DELETE nebo vložním správného klíčového slova, písmena, čísla nebo znaku opravte chybu. Nyní stiskněte ENTER.

Např.: Předpokládáme, že chcete, aby počítač násobil sedm osmi a **nestiskli** jste SYMBOL SHIFT, což způsobí vytištění písmene b místo \*. Ve skutečnosti je tedy na obrazovce příkaz PRINT 7b8 .

Spectrum tento příkaz nezná a po stisknutí ENTER se objeví ? před b, které je chybně. Vše, co je třeba udělat, spočívá v tom, že kurzor posunete na pozici vpravo vedle chyby. Stiskněte "delete", čímž odstraníte b. Nyní stiskněte SYMBOL SHIFT a klávesu B, vytiskne se \*. Povelem ENTER dáte počítači pokyn k opravě. Kursorem již nemusíte pohybovat zpět. Na obrazovce se zobrazí provedená oprava.

Strana 21 originálu, dole

Jak opravit chybu v programovém řádku

Píšete-li program, dáváte jednotlivé očíslované řádky instrukcí, kterým se říká listing. Vypíšete-li po vypsání programu jeho část stisknutím tlačítka K a ENTER, můžete vidět znaménko > proti jednomu z řádků programu. Ne-li, stiskněte a podržte buď nahoru nebo dolů, dokud nezískáte cursor. Stisknete-li pak klávesu EDIT, řádka je duplikování na spodní části obrazovky a může být změněna tak jako dříve pomocí cursoru a kláves DELETE. Stiskněte ENTER, čímž nahradíte nový řádek v programu. Chcete-li vypsát jinou řádku, pohybujte klávesou cursoru nahoru nebo dolů podle řádků na řádek, který si přejete změnit. Poté stiskněte EDIT. Trvá-li to příliš dlouho, stiskněte LIST, poté, co jste zaznamenali číslo řádku a poté stiskněte EDIT. V každém případě řádek, který požadujete, se objeví na spodní části obrazovky a může být změněn. Pro vymazání kompletního řádku z programu jednoduše ponechte samotné číslo řádku a poté stiskněte ENTER. Necháte-li běžet program, který obsahuje chybu, uvidíte obrazovou zprávu. Tyto jsou vysvětleny na str. 74 originálu.

## Televizní počítačka

ZX Spectrum+ může provádět výpočty mimořádně rychle a s velkou přesností. Vše, co potřebuje, je několik čísel pro zpracování dat a znamének jako je + a -, které řeknou, co s čísly dělat. Nejprve klávesu dané instrukce (naleznete znaménko + na klávesnici K):

PRINT 6+2

Toto je povel. Pak stiskněte ENTER. Povel se objeví a odpověď - číslice 8, se vytiskne na obrazovce. Počítač používá pěti znamének známých jako aritmetické operátory pro kalkulace. O tom, co dokáží, se můžete přesvědčit na spodní části této stránky. Můžete je všechny používat stejným způsobem pomocí klávesy PRINT.

Vstup povelů jako je PRINT 6+2 změní počítač Spectrum na počítačku. Avšak může provádět mnoho operací, které běžná kalkulačka nemůže. Např. může zobrazit výpočty a jejich výsledky pohromadě.

Zadejte následující povel:

PRINT "6+2=" ; 6+2

Počítač odpoví zobrazením 6+2=8

Co se stane, to je to, že PRINT vypíše vše mezi dvojitými uvozovkami, co má být zobrazeno na obrazovce, takže 6+2 se objeví. Typy mezi uvozovkami tvoří zadání. Polosloupec instruuje Spectrum, aby zobrazil výsledek neprodleně za znaménkem = .

Znaménka výpočtových operací Spectrum:

Následující znaménka neboli aritmetické operátory jsou používány počítačem Spectrum pro provádění matematických operací. Poznamenejte si, že počítač nepoužívá znaménka x nebo : .

Symboly jsou vypsány následovně:

Klávesa K - symbol + např. sečtete dvě čísla

Klávesa J - symbol - např. odečtete dvě čísla

Klávesa B - symbol x např. násobte dvě čísla

Klávesa V - symbol / např. dělte dvě čísla

Klávesa H - symbol na např. umocněte druhým číslem číslo první.

Váš první program

Poté, co byl vydán povel, a poté, co jej počítač provedl, zapomíná jej. Chcete-li, aby počítač kalkulaci opakoval, musíte ji napsat jako program. Zapište si danou instrukci a poté stiskněte ENTER.

```
10 PRINT 6+2
```

Tentokrát to není přijato přímo. Počítač zobrazí instrukci na obrazovce. Dále stiskněte R(RUN) a ENTER. Výsledek 8 se objeví teprve nyní. Celá instrukce je nyní počítačový program. Uvedením čísla na začátek udává počítači místo instrukce v paměti, avšak nevybírání ji, dokud mu není dán příkaz. Kdykoli necháte běžet program pomocí klávesy R a poté ESTER, instrukce je prováděna. Instrukce je nyní nazývána příkaz namísto povelu a tvoří očíslovaný řádek programu. Příkazy programu jsou vždy prováděny podle jejich řádkových čísel a tyto obvykle stoupají po desítkách, takže mezi ně mohou být vloženy v budoucnosti další řádky.

Nyní spusťte opravdu svůj počítač. Složte tento program. Zapamatujte si, že za zaklíčováním každé řádky stisknete ENTER. Poté, co jste dokončili, stiskněte R(RUN) a ENTER. Když necháte program běžet, uvidíte následovně:

viz program na str. 22 vpravo dole

Veškerá čísla od 1 do 203 jsou zobrazena na obrazovce -

- stiskněte nyní jakoukoli klávesu, vyjma N, pomlčky, STOP nebo BREAK. Objeví se celá nová sada čísel.

Tento program používá proměnné. V tomto případě je proměnná nazývána N. Jakékoli písmeno nebo slovo může být použito -  
- N jednoduše značí stav čísla. Proměnná je dána hodnotou, která se mění, jak program běží. V ř. 10 je klíčové slovo LET použito pro nařízení hodnoty na 1. Ř. 20 zobrazuje hodnotu, která je následována uvozovkami. Poté v ř. 30 je opět použito LET, tentokrát pro nárůst hodnoty o 1, takže N se stává 2. Ř. 40 používá jednoduchého klíčového slova GOTO, čímž posílá program zpět na ř. 20, který nyní zobrazí 2. Ten je opakován dále a toto se celé opakuje v cyklu, dokud celá čísla nenaplňují obrazovku.



Jak se dotážeme programu na číslo

Zastavte program stisknutím klávesy BREAK. Nyní vypište nový řádek

```
10 INPUT n
```

Tento řádek nahradí nově původní instrukci na řádku 10 programu. Pokud program spustíte, počítač nyní čeká, zda do něj vložíte číslo. Stiskněte jakékoliv číslo a stiskněte ENTER. Nyní se opět čísla začnou načítat od čísla, které jste vložili do paměti. Toto je proto, že vstup n udává hodnotu n rovnou číslu, které jste zapsali. Klávesa INPUT instruuje počítač na dotaz po informaci v průběhu programu.

Programování násobící tabulky

Tlačítkem RESET zrušte předchozí program a vložte následující. Tímto programem se ze Spektra stane násobička. Zapišete-li po aplikaci programu nějaké číslo, na obrazovce se objeví násobící tabulka. Stiskněte jakoukoli klávesu kromě N, BREAK, SPACE a program pokračuje. Klávesou BREAK navrátíte program na úvod a můžete zadat jiné číslo. Uvedený výpis programu (viz strana 23 originálu dole a nahoře) platí pro zadání čísel 3 a 146.

Proč je třeba používat závorky

Občas budete potřebovat využít závorek. Vložte do počítače následující dva povely a porovnejte výsledky.

```
PRINT 6 + 2 / 4
```

```
PRINT(6 + 2)/ 4
```

První výpočet vychází 6,5 a druhý 2. Důvod těchto odlišných výsledků je ten, že počítač má zabudovanou soustavu priorit, kterou se při výpočtech řídí. Dává přednost nejprve mocnině, poté násobení, dělení a konečně sčítání nebo odčítání. Vždy však provádí jako první operace uzavřené v závorkách. Takže v prvním výše uvedeném povelu je první dělení 2/4 a pak následuje přičítání výsledku 0,5 k 6. U druhého povelu počítač nejprve sečetl 6 + 2 a poté výsledek vydělil 4.

## Jak používat interpunkci u Spektra

Spektrum používá pro interpunkci určitý okruh znamének. Jsou velmi důležité, neboť mnoho z nich plní funkci jako instrukce počítače a ovlivňuje způsob, kterým počítač přijímá a chápe instrukce v programových řádcích nebo jakým způsobem tvoří výpis na display.

- ; Středník - při povelu PRINT udává počítači zobrazení dvou bodů na různých stranách obrazu
- : Dvojtečka - ukončuje příkaz, následuje-li v programovém řádku více příkazů za sebou
- " Uvozovky - jakýkoliv výraz uvnitř uvozovek je chápán jako text, nikoliv jako proměnná. Uvozovkami začínají a končí instrukce
- , Čárka - používá se u příkazu PRINT, kde udává počítači zobrazení následujících pozic buď ve středu řádku nebo na začátku dalšího řádku. Nepoužívá se pro oddělování tisíců nebo miliónů.
- . Tečka - používá se ve funkci desetinné tečky nebo pro ukončení vety

## Barvy a jak je využít

Počítač ZX Spektrum+ může vyrobit 8 různých barev, kdy každá z nich má rozdílné číslo barevného kódu. Každý barevný kód (každou barvu) můžete využívat třemi způsoby:

- barva okraje
- barva písma (inkoustu)
- barva podkladu (papíru)

### Barevné kódy ZX Spektrum+

Tabulka na str. 24 ukazuje, jaké barvy a jim odpovídající kódy Spektrum používá. Pamatovat kódy si nemusíte, jsou nadepsány nad odpovídajícími číslicemi příslušné barvy. Názvy barev nejsou klíčová slova.

| Číslo | Barva       |
|-------|-------------|
| 0     | černá       |
| 1     | modrá       |
| 2     | červená     |
| 3     | růžová      |
| 4     | zelená      |
| 5     | bledě modrá |
| 6     | žlutá       |
| 7     | bílá        |

Skutečné odstíny, které získáte na vlastním televizoru, závisí na přístroji a seřízení barvy, kontrastu a jasu. Na to potřebujete barevný televizní přijímač.

### Tři způsoby použití barev

Barvy můžete řídit třemi způsoby. Hraniční barva je barva okraje okolo středové plochy obrazovky. Barva tisku je barva, ve které jsou typy (písmena, čísla, znaménka a grafické formy) a body nebo linky zobrazovány. Barva podkladu je barva pozadí, a to buď po celé ploše obrazovky nebo ve čtvercích okolo každého typu.

Zapnete-li počítač Spektrum, užívá předem nastavenou barvu. Barva inkoustu je černá, barvy okrajové a barva papíru jsou bílé. Můžete změnit tyto barvy průběžně tím, že zadáváte přímé povely z klávesnice. Můžete si tuto práci prohlédnout na stránkách 6-7, kde povely pro okraje jsou použity pro kontrolu, že vaše televize a počítač Spektrum byly pro barvy vhodně nastaveny. Nyní stiskněte tlačítko RESET a klávesu pro následující jednoduchý program.

Barevná zkouška:

Zkouška je uvedena na str. 24 vpravo nahoře.

Vzor hvězdiček vznikl v černobílé. Nyní stiskněte BREAK a zadejte povel pro některou barvu. Udejte barvu pro okraj, podklad i tisk, každou s udaným číslem od 0 do 7, stisknutím ENTER po každé z nich a poté opět nechte proběhnout program. Zde ukazujeme dva displeje - první se vstupem BORDER 4, PAPER 2 a INK 7, druhý s BORDER 3, PAPER 5 a INK 1.  
(viz obr na straně 24 originálu - vpravo dole)

## Jak vypsát programy s barvou

Můžete použít BORDER, PAPER a INK klíčová slova, která použijete pro program pro vyvedení textů, tabulek, vzorů a vyobrazení, které mohou být zobrazeny ve všech druzích barev. Použitím hranice v programovém řádku udáváte změnu barvy okraje ve chvíli, kdy Spectrum dosáhne tohoto řádku. Je-li v řádku zápis INK, nová barva písma se objeví, objeví-li se jakékoli typy dané řádky a dále na obrazovce. Barva podkladu v řádku změní barvu podkladu, avšak pouze v okolí všech písmen (znamená to i tečky nebo čárky). Chcete-li celé pozadí displeje stejné barvy, musíte za údaj PAPER zapsat klíčové slovo CLS.

Rovněž lze použít INK a PAPER po PRINT. V tomto případě mají dané barvy tisku i podkladu pouze hlavní typy, které jsou zobrazeny jako tisk. Další program představuje celou okrajovou část, podklad i tisk. Rovněž ukazuje, jak používat barvu pro tisk INK a PAPER za klíčovým slovem PRINT.

### Barevné kombinace:

(str. 25 originálu, program v levém sloupci)

Poté, co jste nechali proběhnout tento program, uvidíte, že kombinace okrajů, papíru a inkoustu jsou takové, jaké zobrazuje výše uvedené vyobrazení. Program má tři proměnné, b pro číslo okraje, i pro číslo inkoustu a p pro číslo papíru. Tlačítko BEEP udává zvuk a řádky začínající slovy FOR a NEXT značí začátek a konec programové smyčky, která mění veškerá barevná čísla z 0 do 7 v pořadí. O využití klíčových slov FOR a NEXT v programových smyčkách se dovíte více v oddíle o programových smyčkách na str. 27.

Poznamenejte si, že tisk a podklad mohou mít obe hodnotu 9. Tím je barva papíru i tisku buď černá nebo bílá - podle toho, jak se značí pozadí nebo typ.

## Programování barevných sloupcových grafů

Další program používá barev Spektra pro výrobu grafu. Ukazuje 12 denních teplot jako žluté sloupce s čísly. V ř. 60 vložte dvě mezery mezi uvozovky. viz program sloupcového grafu na str. 25 - vpravo nahoře Nyní přidejte následující řádky a vypište nový ř. 110 podle níže uvedeného. Graf nyní bude mít dvě barvy. Na str. 33 naleznete informaci o klíčových slovech BREAK a DATA.

### Jednoduché grafiky DIY

Počítač ZX Spectrum+ může zobrazovat grafiky s vysokým a nízkým rozkladem. Oba druhy se mohou objevovat na obrazovce současně. Nízkorozkladové grafiky jsou vyráběny z bloků barev. Na těchto dvou stranách vám ukážeme, jak vyrábět bloky pomocí klávesnice a jak je polohovat na obrazovce.

Obrazovka s nízkým rozkladem:

Obrazovka má při grafice s nízkým rozkladem 32 poloh typů na podélné straně a 22 poloh směrem dolů. Každá poloha na obrazovce má dvojici čísel pro identifikaci. Nejprve je zařazeno číslo řádku, ve kterém je číslo řádku ve směru dolů po obrazovce pro docílení polohy. Horní řádek je nulový a spodní 21. Dále přichází číslo sloupce, které udává číslo sloupce napříč obrazovkou, pro dosažení polohy. Levý krajní sloupec je sloupec 0 a pravý krajní sloupec je sloupec 31. Na str. 80 lze pročíst, jakým způsobem je mříž grafiky s nízkým rozkladem kladena. Další program naplní dané polohy barvami. Klíčová slova RND(na klávese R) volí náhodnou barvu písma.

viz program na str. 26 - vpravo nahoře

Na obrazovce se po celé ploše objeví čtverce. Proto, aby se objevily typy na hlavních polohách, potřebujete klávesu PRINT spolu s klávesou AT. AT je umístěno za PRINT a je následováno číslem řádku, a číslem sloupce. Např. Povel PRINT AT 11,16;"\*" zobrazí hvězdičku na ř. 11 ve sloupci 16, což je střed obrazovky.

Program zobrazující barevné diagonály

Dobrým způsobem, jak vytvářet barevné vzory, je použití smyček kláves FOR NEXT v barevných programech. FOR NEXT jsou smyčky (cykly), které jsou součástí programu, jež se opakují po určitý počet cyklů. Do řádku, který spouští cyklus, můžete dát povel počítači, kolik cyklů má mít smyčka. Pokud toto provede, může být použito pro umístění typů např. na obrazovku.

Nejste omezeni možnostmi počítače, můžete programovat současně více cyklů (smyček). Cykly se mohou prolínat často s velmi užitečnými výsledky. Program vám ukazuje, jak dvě smyčky FOR NEXT (jedna umístěna uvnitř druhé) mohou být použity pro změnu barev a poloh vyráběných pomocí podkladu a tisku. Uvidíte, jak programovat tyto smyčky podle rozpisu na konci této stránky.

### Programování obrazů

U grafik s nízkým rozkladem můžete malovat obrazy s tím, že vyrábíte polohy a barvy podle typů grafiky. Můžete vyrobit vlastní obraz použitím mříže nízkého rozkladu na str. 80. Poté volbou typů grafiky a způsobem, který je uveden na str. 26 dole v modrém zarámování, můžete zaprogramovat řádky jeden za druhým pro vystavení obrazu. Další program vám ukazuje druh výsledku, který můžete získat. Veškeré tvary, které jsou v něm uvedeny, mohou být získány na číselných klávesách. Můžete buď čekat, dokud nezaklíčujete veškeré řádky před rozběhem programu, avšak necháte-li je proběhnout vždy po vstupu každé řádky, uvidíte, jak jednotlivé odlišné části robota se skládají dohromady (zapamatujte si, že jestliže zapíšete typy pro vadnou grafiku, můžete ji vypsat stejně, jako můžete vypsat nepřesné číslo nebo písmeno pomocí příkazu EDIT, a tak jej opravit).

Klíčové slovo TAB, které se objeví na PRINT v ř. 70 je použito pro polohu typů podél řádku, na kterém právě počítač pracuje. TAB je následováno jedním z čísel od 0 do 31, které určují polohu sloupce.

### Jak zvolit typy grafiky

(viz str. 26 originálu vlevo dole - modrá)

Počítač ZX Spectrum+ má soupravu typů grafických klávesnic, které tvoří nízkorozkladovou grafiku, snadnou pro programování. Vidíte je na klávesách 1-8.



Pro volání grafických typů na obrazovku stiskněte klávesu GRAPH, poté klávesu 1-8 s použitím pomlček mezi každou. Typy grafik se objevují na spodní části obrazovky. Bílá část každého typu na klávese je barva inkoustu a černá část barva papíru. Nyní stiskněte opět klávesu, podržte současně CAPS SHIFT. Tentokrát se typy objeví s inkoustem a barvami podkladu inverzně. Tím je přesně umožněno vkládat grafické typy do programových řádku. Pro ponechání grafické volby a návrat na normální klávesnici se jednoduše stiskne opět GRAPH.

GRAPH - tato klávesa se užívá pro přepínání Spectra na grafický modus

Klávesa 8 - tato klávesa se používá spolu s GRAPH a CAPS SHIFT pro modulaci jednobarevné barevné plochy

Jak používat příkazy cyklu FOR NEXT

Smyčky FOR NEXT vždy začínají s řádkem obsahujícím klíčové slovo FOR a TO spolu s proměnnou a její počáteční a konečnou hodnotou, např.

```
30 FOR c=1 TO 6
```

kde proměnná je c. Takto začínající smyčka bude obsahovat řádek, který provádí opakující se operaci. Může být užita i vlastní proměnná cyklu c. Smyčky FOR NEXT vždy končí klíčovým slovem NEXT a proměnnou např.

```
50 NEXT c
```

Je-li program v chodu, celá smyčka se opakuje od FOR do NEXT po soubor čísel, pro který je programována.

Proměnná začíná u první hodnoty před TO a vzrůstá po jedné v každém cyklu, dokud nedosáhne horního limitu. V našem případě se cyklus opakuje 6x, c mění hodnoty od 1 přes 2, 3, 4, 5 a konečně 6.

V prvním programu na str. 25 byly použity tři cykly současně. Tzn., že pro každý cyklus "vnější" smyčky, "střední" smyčka proběhla všemi cykly. "Vnitřní" smyčka proběhla všemi svými cykly mnohem častěji, pokaždé, kdy "střední" smyčka proběhla jednou.

## Nákresy na obrazovce

Grafika počítače Spectrum+ není omezena pouze na vzory s nízkým rozkladem a obrázky. Se svojí schopností vysokého rozkladu (jemné grafiky) může Spectrum vytvořit podrobné představy s ostře ohraničenými obrysy a rovnými nebo zakřivenými čarami a okraji.

Jemná grafika je tvořena body umístěnými jeden vedle druhého tak, aby tvořily čáru nebo vyplnily tvar obrazce v plnou plochu. Každý bod je šedesátičtvrtina čtverce, který je použit jako nejjemnější prvek hrubé (nízkorozkladové) grafiky. Můžete vydat následující povel:

```
PLOT 128,87
```

Uvidíte jeden bod uprostřed obrazu.

Body jemné grafiky se nazývají "pixly", což je zkrácený výraz pro obrazovou buňku. Stejně jako hrubá nízkorozkladová grafika je u jemné grafiky používán na určení polohy údaj dvou parametrů. Tato čísla nejsou stejná jako čísla používaná u hrubé grafiky.

### Rastr jemného rozkladu

Síť je tvořena maticí 256 bodů vodorovně x 176 bodů svisle. Stejně jako u hrubého rozkladu je první parametr souřadnice vodorovná (v intervalu 0 - 255), zleva doprava. Druhé číslo je koordinátor vertikální, avšak čísla běží od 0 dole do 175 v horním vrcholu obrazu. Poloha 0,0 je levý dolní roh. Nikoli levý horní jako u hrubé grafiky (viz strana 80 - výčet hrubého rastru).

### Vytyčování a kreslení

Potřebujete pouze tři klíčová slova pro výrobu jemné grafiky.

Jsou to slova:

```
PLOT, DRAW, CIRCLE
```

Po příkazu PLOT následují dva parametry oddělené čárkou, příkaz umísťuje obrazovou buňku v obrazovém poli. Příkaz DRAW má podobně jako PLOT dva parametry, které udávají vzdálenost jedné polohy od druhé (souřadnice koncového bodu úsečky). Příkaz DRAW nakreslí spojnici mezi těmito polohami. Pokud příkazu DRAW nepředcházela povel PLOT, je výchozím bodem pro kreslení bod 0,0. Pokud tento povel byl již dříve použit, jedná se i o příkaz DRAW vložený dříve, je výchozí poloha dána posledním postavením PLOT nebo koncovým bodem poslední spojnice DRAW. Příkaz DRAW poté nakreslí čáru do nového postavení. Má-li čára procházet obrazovkou doleva nebo dolů, musí být horizontální nebo vertikální vzdálenost negativní. Vyzkoušejte následující program.

PLOT má výchozí polohu na vrcholu obrazovky. Pět příkazů DRAW pak nakreslí 5 červených čar.

Nyní do programu přidejte tyto řádky:

```
4 BORDER 1:PAPER 6:INK 1:CLS
```

```
5 CIRCLE 128,87,87
```

Spusťte program a červená hvězda se objeví v kruhu na zabarveném podkladu. (viz obr. na str. 28 vpravo dole) CIRCLE potřebuje 3 hodnoty. První dvě udávají polohu středu kruhu, třetí velikost jeho průměru. Můžete rovněž přidat třetí hodnotu pro příkaz DRAW. Vyzkoušejte hodnoty mezi 2 a -2 u programu a sledujte, co se stane.

## Jak vytvářet plochy

S jemnou grafikou můžete snadno vytvářet barevné plochy. Plocha vznikne vyšrafováním čarami v těsné blízkosti. To můžete provádět pomocí cyklu FOR NEXT, který mění polohy DRAW tak, že vzrůstají o 1.

Zajímavý efekt získáte, jestliže tyto cáry vyšrafujete ve větších vzdálenostech. Můžete to provést přidáním klíčového slova STEP a čísla do příkazu FOR. Tato technika je užita u Východu slunce na str. 11. Stejně bude účinná i na programu Trojúhelníku. Zadejte jinou řádku 20 a nechte opět proběhnout program.

```
20 FOR x=100 TO 100 STEP 4
```

Tentokrát se vytvoří obraz podobný ventilátoru, který je na obr. na str. 29 vlevo dole. Důvodem tohoto obrazu je příkaz STEP, který provádí vzrůst ve skocích po 4 místo po 1 při každé další cáre.

## Náčrtky na obrazovce

Počítač Spectrum je velmi univerzální. Nepotřebujete psát program pokaždé, kdy chcete vytvořit obraz nebo vzor. Místo toho můžete použít program, který umožňuje vytvořit obraz přímo na obrazovce. Předkládáme zde velice jednoduchý program, který vám toto umožní. Začíná užitím klíčového slova INPUT, kterým se vás zeptá na číslo barvy písma (INK). Další příkaz INPUT má jako parametr textovou proměnnou (označenou kř), kterou zadáte směr kreslení čáry nahoru, dolů, vlevo a vpravo. Jedná se o písmena u, d, l a r.

## Podmíněné příkazy, rozhodování - IF a THEN

Řádky 60 a 90 v předchozím programu obsahují příkazy IF THEN. Tyto příkazy umožňují provádět rozhodování. V tomto případě počítač rozhoduje, kterou klávesu jste stlačili. Zda se jedná o klávesu příkazující pohyb nahoru, dolů, vlevo či vpravo.

Jestliže je některá z těchto kláves stisknuta, nakreslí počítač čáru. Nebude kreslit čáru v případě, že zapíšete velké písmeno.

Klíčové slovo IF je vždy následováno něčím, co počítač Spectrum zkoumá, ověřuje pravdivost nebo jestli se stane - například určení klávesy, zda byly stlačeny. Je-li toto pravdivé nebo stalo-li se to, poté je prováděna činnost THEN. Jestliže ne, program pokračuje do dalšího řádku. Každé následující THEN je v řádku předmětem rozhodování. V tomto řádku

```
110 IF b=5 THEN PRINT "x" :GOTO 200
```

počítač se pouze přesune do řádku 200 v tom případě, že  $b = 5$ .

## Konstrukce vlastních vzorů a obrazů

S počítačem ZX Spectrum+ můžete vytvářet všechny druhy vzorů a obrazů za použití nízko nebo vysokorozkladových grafik nebo obou dohromady. Nejlepším způsobem, jak vytvářet grafiky, je nejprve vykreslit vlastní tvar na kopii mřížky ze str. 80. Pak provádět program, který vytvoří tyto čáry a tvary a uvede je do správných poloh.

Pro vykreslení vzorů a obrazů můžete často užívat klíčová slova smyček FOR NEXT, která opakují část programu několikrát. Pokaždé se poloha a barva typu nebo čáry může měnit, obvykle pravidelně. Předkládáme program, který tuto techniku využívá: Tento program obsahuje tři smyčky FOR NEXT. Smyčka x mění barvu a rovněž rozměr velkého čtverce, smyčka l a c mění řádek a sloupec - polohy malých čtverců pokaždé, když jsou tištěny. Vyzkoušejte změnu čtverce v ř. 60 na hvězdu nebo jiný typ na klávesnici.

### Libovolné efekty a podprogramy

Za použití smyček nepotřebujete udávat identické vzory pokaždé při běhu grafického programu. Použitím klíčového slova RND (pro zkratku RaNDom) - generátor nahodilých čísel - ve smyčkách můžete vytvářet barvy, polohy a další rysy obrazu pokaždé odlišně. Podívejte se na mozaikový program na str. 10. Pracuje, neboť barva grafiky je RND x 7 - tzn., že jakékoli desetinné číslo od 0 do 7 mění barvu podle nejbližšího celého čísla. Tzn., že pokaždé, když je zobrazen čtverec, je jeho barva některá z barev 0-7.

Další program kreslí symetrické vzory grafických typů na obrazovce. Používá RND pro změnu těchto typů a pro změnu jejich poloh. Proměnné i a p udávají barvy "inkoustů a papíru" - označují, kolik vzorů je nakresleno (v tomto případě 4). Proměnné n udávají počet typů každého vzoru, zatímco x je libovolné číslo od 129 do 142.

Příkaz GOSUB 1000 v ř. 50 vysílá počítač do podprogramu. Podprogram je skupina řádků, která funguje jako program v programu. V tomto programu je podprogram na ř. 1000. Zobrazuje grafický typ ve čtyřech čtvrtinách obrazovky tak, že každý je ve stejné vzdálenosti od středu (poloha 11, 16). Tato vzdálenost je dána řádkem 1000 a 1010, i udává vzdálenost v řádcích a c vzdálenost ve sloupcích. INT mění libovolné číslo na celé číslo tak, že i je jakékoli číslo od 0 do 100 a c jakékoli celé číslo od 0 do 15. Poté řádky 1030 až 1060 zobrazují typy grafiky, jejíž kód je x (viz tabulku seřazení typů na str. 51). Klíčové slovo BEEP vytváří zvuk, tento příkaz najdete na posledním řádku podprogramu (1070). Pak RETURN na ř. 1080 vysílá program zpět do hlavního programu za příkaz GOSUB na ř. 50.

Řádek 60 mění barvy podkladu a tisku, příkaz PAUSE 100 na ř. 70 zdrží program na 2 sekundy před jeho sklouznutím do cyklu a opětovného začátku. STOP je potřebné v ř. 90, aby se zastavil program po čtvrtém cyklu.

Tento program můžete změnit nahražením č. 4 v ř. 30 a č. 40 v ř. 50 jinými čísly. Provedete-li rozsah x širší v ř. 40, získáte jiné typy, které se objeví na obrazovce. Proměnné i a p nesmí být větší než 7.

## Použití FOR NEXT v grafikách

Smyčky FOR NEXT mohou být použity velice účinně u grafik s vysokým rozkladem pro vytváření obrazů prováděných v pravidelných tvarech a čarách. Spusťte následující program a nechte jej běžet. Použití pouze dvou klíčových slov PLOT a DRAW ve dvou smyčkách FOR NEXT dokáže nakreslit na obrazovce pět trojúhelníků nebo pyramid.

viz program PYRAMIDY na str. 31 originálu v levém sloupci  
Nyní přidejte řádky v dalším sloupci. Necháte-li program opět proběhnout, shledáte, že laserový paprsek nepřetržitě střílí do nočního nebe a působí vybuchování hvězd.

Je nakreslen z rohu obrazovky do polohy  $x, y$ , kde proměnné  $x$  a  $y$  jsou libovolná náhodná čísla. Toto jsou tedy převrácená čísla poloh hvězd s nízkým rozkladem.

viz program na str 31 nahoře

Příkaz OVER 1 v ř. 160 a 180 umožňuje prvnímu řádku nakreslit laserový paprsek a druhému řádku odstranit jej bez vlivu na ostatní obraz. Uložte tento program (viz str. 38), neboť jej budete později potřebovat.

## Blikání, jas a inverse

Klávesy FLASH, BRIGHT, INVERSE mohou skutečně vytvořit nové barvy, ve kterých bude váš počítač pracovat. Každé klíčové slovo je následováno parametrem 0 nebo 1; můžete je vložit pomocí příkazu PRINT tak, že za 0 nebo 1 uvedete středník. Klíčové slovo FLASH 1 provádí blikání ploch písmen a podkladu, zatímco klíčové slovo BRIGHT 1 zesvětluje barvy. Příkaz INVERSE 1 mění barvu "inkoustu" na barvu podkladu a naopak. Parametrem 0 za těmito klíčovými slovy vracíte obraz do původního stavu.

Vyzkoušejte následující změny:

V programu "ČTVERCE" (str. 30) změňte čtverce na hvězdy v ř. 60 a pak doplňte:

```
15 INVERSE 1
```

Nyní se hvězda jeví černě (podklad barevně) proti barevným pásmům (změna barvy tisku). Příkaz zrušte pomocí parametru 0. Do programu symetrických vzorů přidejte následující povely:



15 BRIGHT 1

16 FLASH 1

Všimněte si, jak FLASH hýbe vzorem. Vložte FLASH 0 : CLS ,  
tak zastavíte blikání.

Tyto změny mění obraz u všech programů. Pomocí slov FLASH,  
BRIGHT a INVERSE v rámci příkazu PRINT omezíte účinnost pove-  
lu na tisk daného řádku.

## Jak vytvářet vaše vlastní počítačové typy

Váš počítač ZX Spectrum+ není omezen pouze na typy grafik, které můžete vkládat klávesnicí, ale ve zvláštních částech své paměti může uschovat jiné typy, které můžete konstruovat sami. Tato grafika je nazývána "uživatelská grafika" a každý program může používat maximálně 21 těchto typů.

Každý typ je vyroben ze 64 malých bodů v barvě dané povelom INK. Rastr bodů má tvar čtverce 8 x 8.

### Konstrukce vlastního typu

Nejprve si načrtněte rastr 8 x 8. Nyní vyplňte některé čtverce, čímž vytvoříte typ. Tyto plné čtverce jsou do počítače vkládány jako 1, prázdné čtverce jako 0. Na obrázku na straně 32 dole (originálu) je ukázka konstrukce "pavouka".

Každá uživatelská grafika je pojmenována písmenem od A do U. Zde není rozdíl, zda pro proměnnou A až U uijeme malé či velké písmeno. V následujícím příkladu výpisu programu je osm příkazů POKE USR, končících BIN následovaným binárním číslem vyjadřujícím pomocí osmi bitů obsazení dané řádky zvolenými body. Tento obrázek pavouka nazveme "s".

Spusťte tento program a stiskněte klávesu "GRAPH" a "s". Na obrazovce se objeví pavouk. Nyní doplňte následující řádky, čímž získáte pavouka v řádku 30. Nechte běžet program. Pavouk se postupně objeví po celé obrazovce.

Navrhujete-li vlastní typy, nezapomeňte, že nebudete schopni tyto typy rozpoznat, dokud nenecháte proběhnout program, který je definuje. Do té doby se budou objevovat ve výpisu jako písmena.

### Jak míchat barvy pomocí "skvrnitých čtverců"

Míchané barvy můžete snadno vyrobit. Musíte sestavit plochu pomocí grafiky tak, aby byla tvořena např. 50% zelenými a 50% modrými body rovnoměrně se střídajícími. Ukázka programu je na straně 32 originálu vpravo dole.

Vše, co musíte v této chvíli udělat, je stanovení dvou řádek "pixlů" a poté instrukce pro počítač, aby je použil místo písmen.

Běží-li váš program, uvidíte, že čtverce jsou skvrnité. Použijete-li stejnou techniku v programu, takovou, která obsahuje klíčová slova barev, skvrnité čtverce budou mít barvu, která je směsí barev PAPER a INK.

Zjednodušení grafiky pomocí příkazů READ a DATA

Existuje snadnější cesta, jak vytvořit na počítači vlastní grafiku, která spočívá v použití desetinných čísel pomocí klíčových slov READ a DATA. Nejprve změňte osmimístné binární číslo složené z nul a jedniček na desetinné číslo. Toto proveďte vysláním klíčového slova PRINT BIN, po kterém následuje číslo, například takto:

```
PRINT BIN 00111100
```

Na obrazovce se zobrazí číslo 60, tj. desetinný ekvivalent binárního čísla 00111100. V případě pavouka je osm desetinných čísel následujících:

60, 126, 219, 255, 189, 165, 165, 36.

Nyní můžete použít klíčových slov READ a DATA. Tato dvě klíčová slova umožňují snadný způsob zakládání většího souboru vstupních hodnot, jako jsou čísla do proměnných v programu. Klíčové slovo READ je doplněno parametrem - jedním nebo více písmeny, manipulujete-li s číselnými proměnnými. Vkládáte-li pomocí příkazu READ textové řetězce, je parametr pouze ještě doplněn symbolem " \$ " (dolar). Když chcete číst desetinná čísla, potřebujete číselnou proměnnou. Nazveme ji y.

Zavedete-li do počítače klíčové slovo READ, vrací se ke čtení prvního příkazu DATA, který obsahuje výčet hodnot, oddělených čárkami. Počítač vybírá první hodnotu z výčtu a přiřadí ji do proměnné READ. Další údaj, který počítač přibírá do READ, je druhá hodnota následující za první proměnnou. Tak jsou načtena všechna data.

Předkládáme vám ukázkou programu, který zobrazí pavouka (viz program na straně 33 originálu vlevo).

Program uschová všech osm desetinných čísel do paměti, na jejich základě je posléze vytvořen typ. Změňte pouze s v řádku 30 na libovolné písmeno, za klíčovým slovem DATA v řádku 50 uveďte osm čísel - každé oddělené čárkou. Stiskněte příkaz GRAPH a písmeno, které vybírá typ z programu.

## Nakreslení šachovnice

Předkládáme vám program, který zobrazí šachovnici a vypisuje části hry. Můžete vytvářet barvy, které řídíte použitím kódů (viz rozpis níže).

Jednotlivé části hry jsou dány programem, který začíná řádkem 500.

### Užívání kódů pro řízení barvy:

Místo užití klíčových slov jako je INK a PAPER, můžete vložit kódy do příkazu PRINT po první uvozovce. Typ, který má být zobrazen, změní barvu výpisu a rovněž se zobrazí v této barvě na obrazovce.

Pro získání kódu nejdříve stiskněte klíčové slovo EXTEND MODE a pak číslo klávesy buď s nebo bez užití klíčového slova CAPS SHIFT. Následující tabulka ukazuje, jak volit jakékoli barvy.

Tabulka na str. 33 originálu - fialové dole

## Animace

Grafiky počítače vyhlíží nejlépe, když se typy nebo řádky pohybují po obrazovce. Tím je vytvářena animace, která na počítači Spectrum není obtížná. Vše, co pro to musíte udělat je změnit polohu, na které je typ vytištěn, nebo změnit vypsanou řádku. Nejlepší způsob je využití jedné nebo více smyček FOR NEXT.

### Svislý a vodorovný pohyb

Zaklíčujte a nechte běžet tento program. Nemáte-li znovu nastaven nebo vypnut váš počítač od výroby typů grafiky pavouka ze str. 32 originálu, nevsímejte si vstupných řádků 10-50. Typy grafiky stále zůstanou v paměti pod písmenem "s". Uvidíte padat pavouka dolů po obrazovce - bude klesat na svém vlákně pokaždé, pokud necháte proběhnout program. V programu vytváří řádky 60-100 další grafický typ ("t") pro vlákno. Oživení se objeví v řádcích 110-140, které vytváří smyčku FOR NEXT, ve které l (č. řádku) se mění od 0 do 20. Pokaždé se smyčka opakuje, délka vlákna je tištěna v jedné poloze a pavouk je tištěn o polohu níže. Při dalším okruhu je pavouk nahrazen delší délkou vlákna a opět se objeví níže. Tímto způsobem pavouk rychle klesá po vláknech, dokud nedosáhne ř. 21, který tvoří spodní část obrazové plochy.

Počítač Spectrum může propočítat novou polohu velmi rychle, takže pavouk klesá rychle. Pro zpomalení akce vložte následující řádek:

```
135 PAUSE 10
```

Tím donutíte počítač čekat na dobu **pětinu vteřiny** pokaždé, než vytiskne pavouka na další pozici. Vyzkoušejte změnit 10 na jinou hodnotu a sledujte, jak se mění rychlost.

Nyní přiveďte zvláštní řádek do programu a nechte jej opět proběhnout. Uvidíte oživení v jiném směru.

Ubíhající pavouk - str. 34 vpravo nahoře.

Nyní pavouk utíká do jedné strany, jakmile se dotkne spodní části. Zvláštní řádek z další smyčky FOR NEXT, který řídí sloupec polohy c - poznamenejte si, že prostor je vytištěn dříve a poté je pavouk zobrazován v další poloze sloupce. Tím je způsobeno, že pavouk mizí z jedné pozice a objevuje se v další. Vždy je výhodnější zrušit typ před vytištěním na další poloze, než vytisknout nejprve typ a poté zrušit jeho staré umístění. Tím je pomáháno k vyrušení nebo snížení komíhání u pohyblivých grafik.

### Procvičování s terčem

U mnoha počítačových her se činnost často zakládá na dvou pohyblivých formách, jejich srážkách nebo na předmětu, který je trefován paprskem. Jak počítač ví, kdy dochází ke srážce nebo explozi?

Detekce srážky není obtížná. Jestliže dva typy vytištěné na pozici l,c (pro řádek a sloupec) a polohu v,h (pro svislé a vodorovné), poté jestliže l=v a c=h musí být ve stejné poloze. Napište toto jako příkaz, např.

```
160 IF l=v AND c=h THEN PRINT "CRASH!"
```

Dalším způsobem kontroly srážek je použití barev. Pomocí zavedení příkazu klíčového slova NEW zrušte program pavouka. Poté vložte program Pyramidy ze str. 31 nebo založte program, máte-li ho uschován na pásce. Nyní můžete stupňovat tento program a kombinovat jej vlastním pavoukem (je stále v paměti, pokud jste nestiskli klíčové slovo RESET nebo nevypnuli počítač), který vyrábí další program.

Nejprve přidejte následující řádky, které tvoří explozi grafiky nazývanou "e".

Nyní zrušte ř. 190 a přidejte nebo změňte následující řádky.

Pavouci a pyramidy - program na str. 35 vlevo uprostřed. Zážeh hvězd se již neobjeví. Místo toho padají oblohou pavouci a ujídají pyramidy a zem. Co musíte provést - přidat smyčku FOR NEXT, ve které v a h udávají polohu pavouka. Pro proměnnou h náhodnou, kdy pavouk zahajuje svislý pád na různých místech napříč obrazovkou. Dále přidejte následující řádky:

Vybuchující pavouci - program na str. 35 vlevo dole.

Poté co laserový paprsek trefí pavouky, celé se změní na žluté. Když čára - paprsek, vyvíjená pomocí DRAW dosáhne typu pavouka - jeho polohy, barva inkoustu se změní na stejnou barvu, jako je barva řádku, který je žlutý. V ř. 190 rozlišuje klíčové slovo ATTR zežloutne-li pavouk a vysílá počítači údaj o explozi do podprogramu na ř. 500.

#### Odráž míče

Mnoho grafických programů spočívá v odražení tvarů od okrajů obrazovky. Následující program ukazuje, jak toto provádět. Proměnné v a h pracují stejným způsobem, jako u programu vybuchujících pavouků, avšak je dodáno +1 nebo -1 k v nebo h, čímž dosáhneme klesání nebo stoupání míče a pravého nebo levého posuvu. **SCREENS** kontroluje, kde je X na pozici v,h.

#### Použití atributů

Klíčové slovo ATTR vyhledává "atributy" na hlavní poloze obrazovky. Atributy jsou v barvě inkoustu a papíru, a to i tehdy, když poloha bliká nebo je světlá. V programu explozujících pavouků zajišťují ATTR, že je pavouk zničen, zežloutne-li. To je po barvě jeho inkoustu (č. 6).



Barva papíru je modrá (č. 1) a pavouk není světlý nebo blikající. To znamená, že jeho atribut je celkem 14. Vstup ATTR v programátorské referenční příručce ukáže, odkud tato čísla pocházejí.

## Jak vyrábět hudební a zvukové efekty

ZX Spectrum+ může sloužit za zvukový syntezátor, který může vytvářet vaše programy s velkou škálou hudebních zvuků a zvláštní škálou zvukových efektů. Je jednoduchý pro použití i při malé nebo žádné znalosti hudby.

Syntezátor vydává zvukový signál, který vstupuje do zabudovaného reproduktoru v počítači.

### Programování zvuků

Pro výrobu zvuků na počítači použijte pouze jednu klávesu - BEEP. Je následováno dvěma čísly nebo proměnnými, která představují čísla. První udává počítači, jak dlouhý (v sekundách) má být zvuk, druhý informuje, jak vysoký nebo nízký má zvuk **špičkový kmitočet**. Špička je měřena v půltónech. Hodnoty špičky jsou 0 pro střední C, 1 pro C#, -1 pro B (Cb) atd.

Abyste slyšeli celý rozsah zvuků, které může počítač vydávat, ponechte běžet následující program: (str. 36 vlevo dole).

Počítač Spectrum probírá svůj celý rozsah tónů od nejvyššího tónu (69) do nejnižšího (-60). Zjistíte, že nejvyšší noty jsou téměř neslyšitelné, a nejnižší zní jako praskot. To je způsobeno tím, že tyto tóny přesahují za rozsah slyšitelnosti ucha.

Graf na spodní části str. 36 originálu ukazuje, které tóny v rozmezí not můžete transformovat do hudebních záznamů a zakódovat do programů Spectra.

Hodnoty tónů pro tvorbu hudby: udáváme hodnoty Spectra pro spodní až nejvyšší basy a výšky. Přidejte 1 k špičkové hodnotě pro ostrý tón, uberte 1 pro basovou notu.

### Zvukové efekty

Můžete získat veškeré druhy zvukových efektů, obvykle umístěním signálu BEEP do smyčky, která rychle mění špičkovou hodnotu tónů. Vyzkoušejte následující programy a vyzkoušejte pomocí těchto programů vytvářet vlastní zvuk. Uvědomte si, že trvání hodnot je velmi krátké, tvoří setiny sekundy. Stiskněte klíčové slovo BREAK, čímž ukončíte smyčkové programy.

### Bublání

Tento program hraje skupinu tří tónů podle náhodné volby. Rozsah je široký, avšak můžete měnit hodnoty v ř. 10, čímž rozsah měníte.

### Stroj

Tento program dává dva zvuky - jeden stoupající a druhý klesající. To je způsobeno tím, že jsou vydány dva příkazy BEEP, které udávají dva tónové zvuky proti sobě po dobu pouze setiny sekundy na odlišných špičkách.

### Stoupání

Tento program je podobný strojovému programu, avšak nyní dva tóny stoupají spolu po šesti půltónech. Navíc špičková hodnota se mění o 0,2 - pětina půltónu. Tím se zvuk zdvihá pouze pomalu. Vyzkoušejte malou změnu špičky pomocí hodnoty klíčového slova STEP.

### Převaděč klávesnice

Tento program čeká, pokud nestlačíte další klávesu. Poté, co klávesu stlačíte, vydává různé zvuky. Poznamenejte-si, že stisknutí tlačítka CAPS SHIFT, kdy držíte jinou klávesu, snižší zvuk. Tento program pracuje na principu toho, že klíčové slovo CODE INKEY\$ udává rozlišenou hodnotu pro pokaždé, kdy je stisknuta nová klávesa. Druhá řádka nastavuje počítač, tento dál již zvuky netvoří, pokud není stisknuta klávesa. Můžete získat hodnoty pro CODE, kdy toto se vrací na typy, uvedené v tabulce na str. 51.

### Zvuk a obraz

Zvukové efekty, které může vydávat počítač Spectrum, se nejlépe sladí se spoluprací zobrazování na obrazovce. Pro představu, jak můžete přidat zvuk do programu, se vraťte ke kompletnímu programu Ubíhajícího pavouka na str. 34.

Zapamatujte si, že jste vložili příkaz PAUSE na ř. 135, čímž jste zpomalili chod. Namísto zrušení programu tímto způsobem můžete naprogramovat pauzu, která vydává zvuk. Změňte ř. 135 na nový řádek

```
135 GOSUB 500
```

Nyní přidejte následující řádky programu - str. 37 vpravo  
uprostřed

Nechte proběhnout tento program, pavouk nyní vydává při pádu chvějivý zvuk. Tento podprogram hraje tři tóny velice rychle, tóny klesají stále níže, jak se pavouk blíží do protilehlé polohy obrazovky. Zkuste přidat více not

změnou ř. 500 a zrychlením nebo zpomalením not změnou 0,02 v ř. 510.

### Jak zesílit počítačové zvuky

Aby byl zvuk počítače Spectrum silnější, můžete připojit buď do zdířky EAR sluchátka nebo zesilovač a reproduktor. Zesilovač bude mít úroveň řízení hlasitosti, kterou půjde hlasitost zvuku nastavit podle vašeho přání.

Nejjednodušším způsobem je použití přívodního kabelu pro zvuk - propojení zdířky EAR do zdířky MIC kazetového přehrávače. V případě nezbytnosti vyjměte kazetu, zapněte kazetový magnetofon a pak stiskněte klávesu PLAY, REWIND(REVERSE) nebo FAST FORWARD(CUE). Seřídte úroveň hlasitosti kazetového přehrávače a můžete poslouchat zvuk z počítače, který vám přichází z reproduktoru magnetofonu. Alternativně můžete připojit sluchátka - je možno je připojit s kazetovým přehrávačem.

Rovněž můžete propojit počítač Spectrum s hi-fi sestavou nebo hudebním centrem, abyste docílili skutečně plného zvuku. Budete potřebovat zvláštní kabel s 3,5 mm zástrčkou pro stranu Spectra a **zástrčkou** pro vnitřní zásuvku v hi-fi zesilovači nebo hudebním centru. Počítač Spectrum vydává lineární signál obdobný výstupu z kazetového magnetofonu a páskového magnetofonu, takže pro tuto činnost je vhodné zapojit do zdířky LINE IN nebo REPLAY. Máte-li jakékoli obtíže, spojte se s obchodem, který vám dané zařízení prodal.

### Jak uschovávat vlastní programy

Budete-li **chtít** uschovat před vypnutím přístroje vlastní programy na kazetový pásek, propojte kazetový přehrávač s počítačem Spectrum a uložte program, který je v počítači. Spectrum vysílá program na kazetový magnetofon ve formě, která může být zaznamenána na pásek. Poté kdykoli potřebujete program použít, naplníte jej z kazetového magnetofonu zpět do počítače použitím plnicího postupu popsaného na str. 14-15 originálu. Na těchto dvou stránkách můžete sledovat, jak ukládat programy a dále jak kontrolovat, že programy byly správně uloženy.

#### Nahrávání vlastních programů:

1. Nejprve propojte počítač s vhodným kazetovým magnetofonem použitím přívoďního kabelu podle popisu na str. 14, avšak přesvědčte se, že pouze zdířka Spectrum MIC je zapojena do kazetového magnetofonu.
2. Má-li kazetový magnetofon nahrávací úroveň nebo řízení hlasitosti, nastavte jej asi na 2/3 maxima. Nemá-li, nemějte obavy, neboť úroveň záznamu bude nastavena automaticky.
3. Stiskněte klávesu SAVE, po které vypíšete název programu v uvozovkách, např.  
SAVE "prog2"  
Jakákoli kombinace až do deseti čísel může být využita pro tuto potřebu. Nyní stiskněte klávesu ENTER. Zmizí řádek SAVE a poté uvidíte na obrazovce příslušné instrukce, jak manipulovat s kazetovým magnetofonem.
4. Nastavte kazetový magnetofon na záznam, obvykle stisknutím tlačítka RECORD a PLAY. Poté stiskněte jakoukoli klávesu na počítači.
5. Nyní počkejte po dobu, kdy počítač ukládá program. Nejprve uvidíte modré a červené pásy pohybující se pomalu po obrazovce. Poté bude následovat krátký záblesk modrých a žlutých pruhů. To je v tom případě, kdy Spectrum vysílá jméno programu na pásek.

6. Dále následuje krátká mezera a poté více modrých a červených pásů. Toto je následováno modrými a žlutými pruhy opět tehdy, kdy Spectrum vysílá program na pásku. Dlouhý program může zabrat poměrně dlouhou dobu.
7. Když byl celý program odvysílán na pásek, objeví se na obrazovce zpráva O OK,0:1.  
Zastavte pásek. Program byl nyní uložen. Máte-li zájem, můžete nyní kontrolovat nebo "ověřit" toto uložení.

Nyní ověřte svůj program

Ačkoli vysílal počítač program na kazetový přehrávač, nemůžete mít jistotu, že program byl účinně nahrán na pásek. Samozřejmě, že počítač Spectrum je schopen toto pro vás zkontrolovat.

Tento postup se nazývá ověření (verifikace). Nejprve převiňte pásek na začátek programu, poté propojte zdířku Spectrum EAR do kazetového EAR, můžete ponechat zdířku MIC propojenou. Nyní zakličujte slovo VERIFY, po kterém uvedete jméno programu v uvozovkách. Poté stiskněte klávesu ENTER a spusíte pásek. Bude vidět stejný sled modrých a červených pásů a modrých a žlutých pruhů. Program bude vytištěn a zůstane, dokud nedojde k ověření. Konec ověření je vyznačen modrými a žlutými pruhy ve druhé části

0 OK,0:1

bude zobrazen na obrazovce. To znamená, že počítač Spectrum zkontroloval program na pásku s programem, který je uložen v jeho paměti, a shledal je totožnými. Program byl kladně ověřen.

Nyní **si** můžete být jisti, že váš program je bezpečně na pásku.

Typy pro úschovu software

1. Vypište jméno programu na nálepku kazety nebo kartičku, a to ihned, kdy jej uschováváte. Používejte stejná velká nebo malá písmena, která jsou zobrazována na obrazovce. Má-li kazetový magnetofon počítadlo, užíjte jej, abyste lokalizovali program na kazetě a zapište si číslo počítadla k názvu.
2. Před uschováváním programu umístěte jméno programu do programu použití příkazu REM, např.  
5 REM SPIDER program Version 3  
Počítač nebere na zřetel žádné příkazy REM při běhu programu a můžete použít klíčové slovo REM pro uvedení uvozovek a poznámek do programu, kdekoli potřebujete.



Nebyla-li vysílána na obrazovce zpráva 0 OK,0:1, poté se stalo něco špatného. Nejprve zkontrolujte rozpis - blokové schéma - na str. 16 - Odstraňování závad při zakládání software - závada může být v tom, že program je sice uložen na pásce, avšak nebyl vložen zpět do počítače pro ověření. Shledáte-li zde jakoukoli závadu, napravte ji, převiňte pásek a ověřte program opět. Jestliže počítač stále neověřuje program, obraťte na následující stránku 40, kde je blokové schéma k odstraňování závad při úschově programu - software. Nestiskávejte klávesu NEW, RESET nebo turn off, neboť poté byste ztratili program z paměti, aniž byste měli ověřenou kopii na pásku.

#### Automatické spuštění programu

Po klíčovém slově SAVE můžete uvést jméno programu a pak číslo řádku od kterého se má program spustit (např. číslo 1).  
SAVE "SPIDER" LINE 1

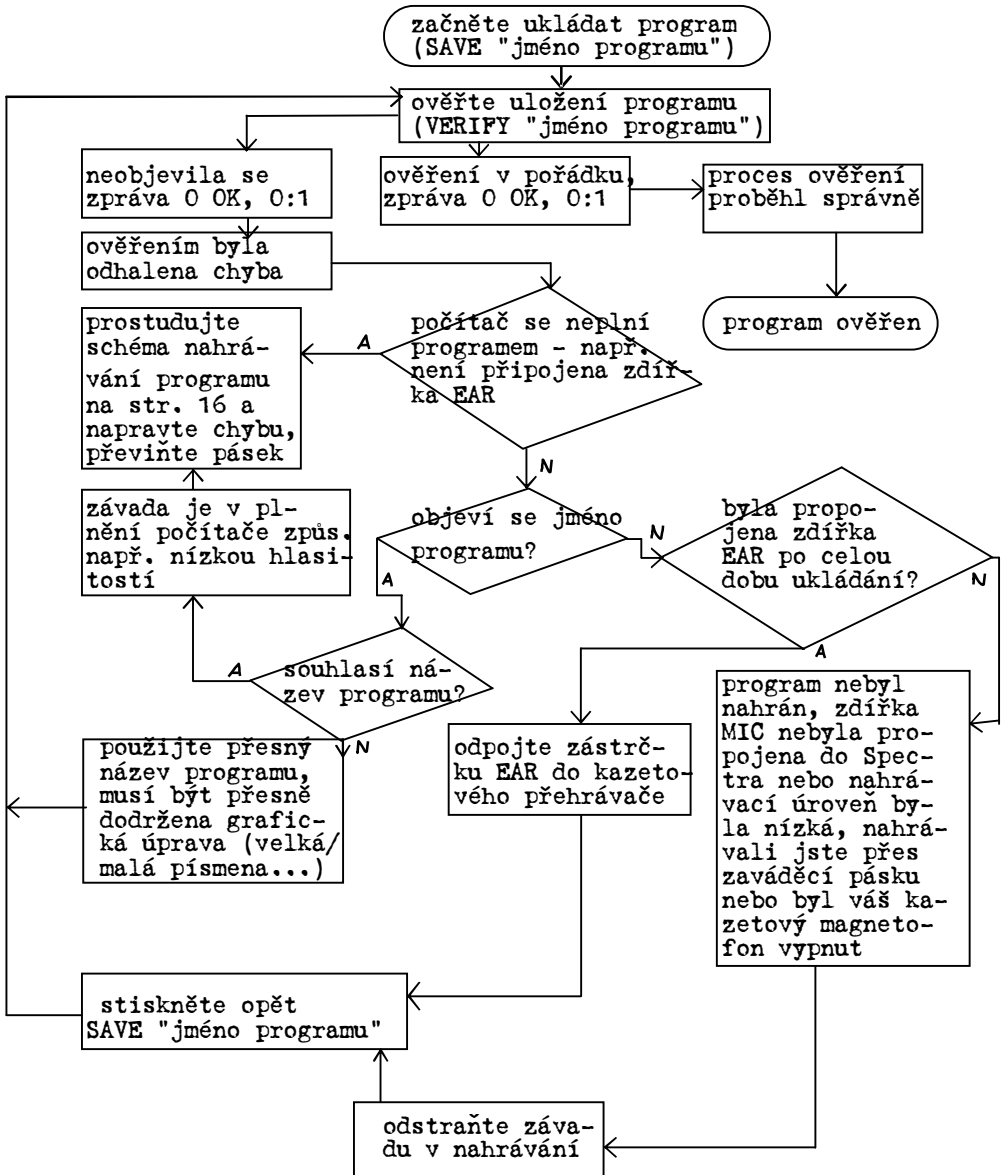
Úschova programu je stejná jako v předchozím případě, avšak při ověřování se nekontroluje řádek 1 a název programu. Programy takto uložené začínají automaticky. Není potřeba stisknout klávesu RUN, ale nesmíte zapomenout zastavit kazetový magnetofon.

Nemá-li program v sobě řádek s číslem uvedeným v příkazu automatického startu, přeskočí počítač na další následující řádek a od tohoto řádku se spustí program.

#### Úschova CODE, SCREEN § a DATA

Klávesa SAVE může být rovněž použita ve spojitosti s klávesou CODE nebo SCREEN § nebo DATA. Způsob, jak uchovávat data zadaná tímto způsobem naleznete v příručce programovacího jazyka "Programmer's Reference Guide", která tvoří kapitolu 4 tohoto manuálu.

### Odstraňování závad při ukládání software



### 3. Naučte se znát váš ZX Spectrum+

Tato kapitola vás zavádí dovnitř počítače, vysvětluje, jak pracují komponenty pod klávesnicí a jak jsou spolu propojeny, aby počítač fungoval. Rovněž ukazuje, jak užít "periferie" - přídatná zařízení, která vám umožní zvýšit Spectrum na úplný počítačový systém. Naleznete zde i více informací o technické otázce vašeho počítače, včetně způsobu, jak je organizována paměť a technickou specifikaci počítače Spectrum.

Co je uvnitř?

Abyste to zjistili, přečtete si následující a nepokoušejte se počítač rozebírat, abyste zjistili, jak pracuje. Připravili byste se o záruku a přitom můžete přivodit vážné poškození přístroje.

Uvnitř pouzdra jsou pružné páskové spoje, které propojují klávesnici se zbytkem počítače. Tento je umístěn na jedné desce plošného spoje, jsou zde standartní elektrické díly jako jsou odpory a kondenzátory. Nejdůležitější součástí jsou černé pravouhlé integrované obvody, uspořádané do skupin.

Uvnitř integrovaného obvodu

Funkční část integrovaného obvodu je ve skutečnosti mnohem menší než plastický obal, který má funkci pouzdra. Pouzdro nese veškeré vývody, které integrovaný obvod potřebuje pro komunikaci s okolím. Vlastní "čip" je tenký křemíkový plátek, který obsahuje mnoho tisíc elektrických spojů. Každý spoj účinkuje jako spínač pro zastavení, přechod nebo uložení elektrického signálu, který jej dosáhne. Ačkoli to je jednoduchý postup, je zde tolik spojů navzájem propojených, že mohou vydávat signál, který ukládá nebo utváří informaci a zpracovává ji s nepředstavitelnou rychlostí a přesností. ZX Spectrum+ má velký počet odlišných čipů, z nichž každý je konstruován pro vlastní podíl na chodu počítače.

Jak jsou čipy zapojeny

Váš počítač je elektrický obvod neobvykle rozsáhlé komplexnosti. Kódové signály, které se skládají z elektrických impulsů, které stále blikají podél vnitřních obvodů a mezi čipy a komponenty, čímž dochází k vlastní činnosti počítače.

Jak je vše udrženo v pořádku, tak, že správný signál se dostane na správné místo ve správnou dobu? Ukryt uvnitř jednoho ze čipů je řídicí obvod - krystalové hodiny počítače.

Jeho tikání pomocí vzdávání elektronických impulsů je 3,5 milionů každou sekundu. Tyto pulsy se pohybují obvody, čímž vytváří kódové signály, které řídí akci každé části a vše udržují ve vzájemném souladu.

Popis počítače ze str. 42-43:

Vnitřek vašeho ZX Spectrum+ - následuje pohled na obvody na desce Spectrum, dvě pružná propojení s klávesnicí byla odstraněna.

Je-li Spectrum používán, stisknutí klávesy propojí dvojici kablíků pod klávesnicí. Tím je vyslán kódový signál do CPU.

ULA - neřízený logický obvod - tento čip generuje display podle informací, které jsou v RAM a rovněž je činný jako řidič systému.

TV output - televizní výstup - tento vyrábí signál, který prochází do televizního přístroje.

Keyboard connector point - přípojka pro klávesnici - jeden z gumových kablíků do klávesnice je připojen v těchto místech.

TV encoder - televizní kodér - mění signály, které vydává počítač, do signálů barevného televizního vysílání.

RAM - paměť s libovolným vstupem - tyto čipy obsahují program, který je zakládán do počítače v jakékoli dílčí informaci, která je programem vyžádána, jako jsou např. hodnoty, které jsou proměnné. Obsah 48kB RAM může být z klávesnice měněn a může být úplně vymazán stisknutím RESET nebo vypnutím počítače.

Cassette sockets - zdiřky pro kazetový přístroj - jsou užívány pro vysílání informací a programů z paměti do magnetofonového přístroje a na pásek a pro jejich přijímání zpět do paměti z pásku.

Logické čipy - tyto čipy jsou činné jako interference při výměně informací CPU a RAM.

CPU - mozek počítače - CPU je Z80 mikroprocesor - provádí veškeré výpočetní kalkulace a řídí celkovou činnost počítače.

9 Voltová zdiřka - připojuje zdroj k počítači.

Edge connector - **břítový konektor** - tento propojuje Spectrum s vnějšími zařízeními, jako je tiskárna apod.

ROM - paměť, kterou je možno pouze celou pročitat bez možnosti vstupu do libovolné části uspořádání.

16K část paměti, která udržuje stálé pracovní instrukce potřebné obvodem CPU. Mezi jinými tyto instrukce převádějí instrukce jazyka BASIC do formy, které může CPU porozumět. Obsah těchto paměťových čipů nemůže být klávesnicí ovlivňován a měněn.

Keyboard connector point - přípojně místo pro konektor klávesnice - jeden z pružných přípojných konektorů z klávesnice je připojen v tomto místě.

Loudspeaker - reproduktor - vydává podle požadavků zvuk.

Voltage regulator - regulátor napětí - tento díl zabraňuje změnám napětí, které by mohly poškodit počítač.

Jak vlastně pracuje váš počítač ZX Spectrum+ ?

ZX Spectrum+ zpracovává, stejně tak jako ostatní počítačové systémy, informace ve čtyřech hlavních částech.

Tyto jsou vstupní jednotka, jako je klávesnice, která ukládá informace nebo program do počítače, dočasné nebo trvalé paměti, které uskladňují informace, programy a operační instrukce, CPU - ústřední řídicí jednotka, která provádí dané instrukce na základě informací, a výstupní jednotky, které udávají výsledky.

Zakládání a spouštění programu

Co se stane uvnitř Spectra, když založíte a necháte běžet velice jednoduchý program?

Zde např. jednořádkový příklad

```
10 PRINT 6+2
```

Nejprve pracujete s klávesnicí. Pod klávesami je mříž překřížení kabelů. Pokaždé, když stisknete klávesu, se dvojice kabelů propojí a vysílá kódový signál do CPU. CPU jej obratem vysílá v kódu do RAM, kde je uložen.

Když program běží, CPU vybírá uložené kódy z RAM jeden po druhém podle příkazu programu. Nejprve obdrží kód pro PRINT, který udává, aby získala dílčí pracovní kódy z ROM. Tyto operační kódy dosahují CPU a poté je CPU připravena pro provádění činností, které zobrazí hodnoty na obrazovce. CPU potom bere hodnotu 6 z RAM. I tato je ve formě kódu a CPU ji ukládá v malé vnitřní paměti nazývané registr. Dále přichází kód pro sčítání a CPU opět přijímá nezbytné operační kódy z ROM. Nakonec CPU přijímá kód pro 2 z RAM. Přidává tento kód k hodnotě v registru, čímž získá výsledek 8. CPU poté převádí výsledek do další soustavy kódů a tyto vysílá do displayového uspořádání. Toto je část RAM, která udržuje kódy pro veškeré informace, které vidíte na obrazovce a na obrazovce se objeví číslo 8.

## Ukládání programu

Dotazujete-li se počítače Spectrum na úschovu programu na pásek, CPU opět vybírá kódy z RAM. Avšak namísto toho, aby je zpracovával, vysílá kódy do převáděcí jednotky, která je mění na zvukové signály. Tyto signály jsou vysílány do kazetového magnetofonu a zaznamenávány na pásek. Když plníte program později, zvukové signály z kazetového magnetofonu jsou měněny zpět na počítačové kódy pomocí konvertoru. CPU je vysílá zpět do RAM, kde jsou až do chvíle vyžádání uschovány.

## Binární kódy:

Veškeré kódy, které zpracovává váš počítač jsou v binární formě. Nazývají se binární - dvojkové - protože jsou všechny složeny pouze ze dvou typů signálů. Mohou být představeny jako dvojková čísla, která jsou čísla obsahující pouze dvě číslice - 0 a 1. Binární číslo pro 6 např. je 00000110.

Uvnitř počítače se kódy sestávají ze sekvencí rychlých elektrických pulsů. Dosáhne-li tento puls kteréhokoli bodu, představuje ve dvojkové soustavě 1. Nepřichází-li, představuje 0. V počítačovém kódu je proto 6 :

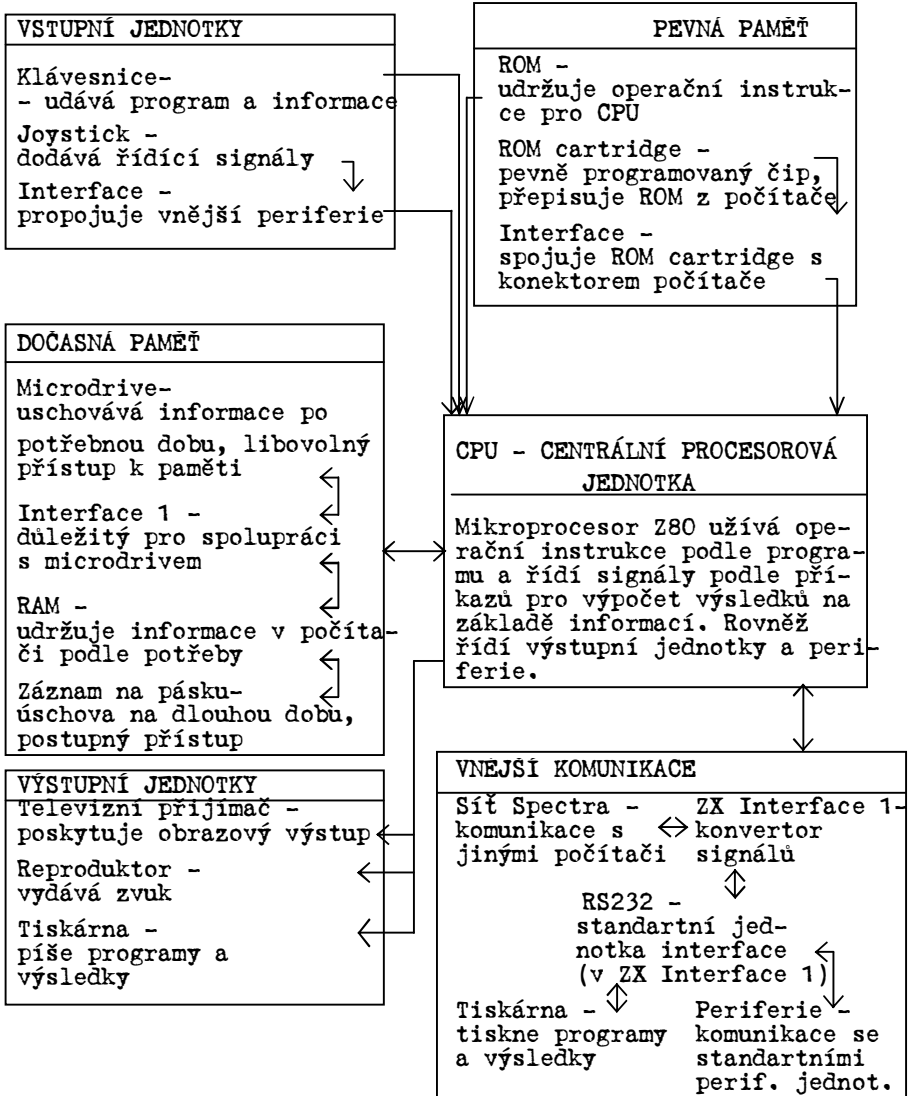
vypnuto-vypnuto-vypnuto-vypnuto-vypnuto-zapnuto-zapnuto-vypnuto. Spectrum manipuluje současně osmi kódy.

Na diagramu na str. 14 je zřejmé, jak různé typy binárního kódování, používané počítačem, přenáší informace z jednoho místa na druhé.



## Vstupní a výstupní pochody v počítači

Následující diagram ukazuje, jak kódové informace procházejí ze vstupních jednotek (klávesnice) do počítačového procesoru a pak do výstupních jednotek (televizor). Jednosměrné šipky značí směry, které jsou pouze jednosměrné, šipky dvousměrné označují trasy obousměrné.



## Jak připojit periferie

Můžete umocnit váš ZX Spectrum+ na komplexní a výkonnou výpočetní soustavu pomocí slučitelných Sinclair a dalších Spectrum-periferií. Středem této soustavy je ZX Interface 1 - která umožňuje propojení mikrodrivů pro rychlou a jednoduchou manipulaci s programy a údaji a která rovněž propojuje široký rozsah odlišných periferií včetně druhých počítačových jednotek Spectrum. S tímto interfacem můžete připojit počítač Spectrum na standartní tiskárny, zapojit počítač do modemů umožňujících přenos a příjem programů nebo dat přes telefonní linky. Další interface mohou propojovat zástrčkami náboje ROM do počítače pro přímé zakládání programu. Navíc jimi můžete rovněž připojit ruční ovládací páčky pro snadnější řízení her.

### Tiskárny Spectrum

Některé tiskárny se zapojují přímo do rovného konektoru Spectrum. Máte-li již tiskárnu Sinclair ZX, můžete ji připojit do počítače bez použití interface. Tento druh tiskárny se rovněž může zastrčit do zadní ZX Interface 1. Použití tiskáren, které vyžadují výstup RS232 musíte použít zdířku D na ZX Interface 1.

### ZX Interface 1

Jednotka ZX Interface 1 je připojena na zadní části spodní Spectra. Propojuje Spectrum s více než 8 microdrivy, s až do 63 dalšími počítači Spectrum a přes standartní jednotku interface RS232 s širokým rangem standartních periferií. Microdrivy a microdrivové náboje nahrazují kazetový přehrávač a pásky pro uschovávání programů a údajů. Vkládáním nábojů microdrive můžete uložit, ověřit a založit programy v sekundách. Každý náboj může uložit až do 85kB údajů a při použití maxima 8 microdrivů váš počítač bude mít kapacitu paměti 680kB! Veškeré programy jsou vyhledávány automaticky s běžným přístupovým časem 3,5 sekundy.

Použitím síťových vodičů dodávaných s jednotkou interface můžete propojit váš počítač s dalším - buď ZX Spectrum nebo jiným ZX Spectrum+. Tato síť může být poté propojena na maximálně 63 dalších počítačů Spectrum. Informace jsou vyměňovány rychlostí 10 000 typů za sekundu.

Jednotka ZX Interface 1 rovněž zahrnuje interface RS232 s devíticestným D-konektorem pro tiskárnu, další standartní periferie, modemy a další počítače, propojitelné se Spectrem.

Standartní interface kabel je rovněž možno objednat.

Náboje ROM a páčky ručního ovládání her

Interface jako ZX Interface 2 umožňuje propojit náboje ROM a ruční ovládací páčky. Náboje ROM plní bezprostředně po **energizování** programy, které by běžně při přehrávání z pásky zabraly dlouhou dobu.

Popis obr. na str. 46-47 originálu

Plnění microdrive - microdrive jsou vkládány do otvoru v přední části drivu

Zapojení náboje ROM - tyto náplně jsou vkládány do zdičky v interface. Poté, co je počítač zapnut, program je automaticky zakládán přechodem přes vnitřní ROM počítače.

Přípojně periferie - v průběhu činnosti je zastrčeno ZX Interface 1 do rovného konektoru, takže je za a pod počítačem. Vyobrazení ukazuje soustavu před propojením s počítačem.

Jednotky microdrivů - až 8 těchto paměťových jednotek může být připojeno do jednoho počítače Spectrum.

Pružný plochý kabel - propojuje microdrivy s počítačem přes ZX Interface 1.

ROM cartridge/joystick interface - interface pro připojení náplní ROM a páček pro přímé řízení her.

Rovný konektor - periferie jsou vedeny do počítače tímto konektorem.

Připomínka - ZX Spectrum+ má dvě vestavěné nožky, které mohou být použity pro náklon klávesnice. Tyto nožky nemusí být použity, je-li připojeno ZX Interface 1.

Varování - periferie musí být vždy připojeny před zapnutím počítače do sítě.

## Uspořádání paměti ZX Spectrum+

Pohlédnete-li na fotografii vnitřku Spectra na str. 42-43, uvidíte, že je zde jeden čip ROM a šestnáct menších čipů RAM. Tyto čipy vytvářejí paměť počítače Spectrum. Paměť sestává z 65 536 paměťových jednotek, z nichž každá obsahuje jeden byte (číslo od 0 do 255). Každá jednotka je identifikována pomocí čísla, nazývaného adresa.

ROM znamená paměť s postupným čtením. Táto část paměti obsahuje operační instrukce pro CPU. Má rozsah 16kB, což znamená, že obsahuje 16x1024 (16384) bytů nebo adres. Byty mohou být z paměti pouze čteny, takže nemohou být měněny. Kdyby byly měněny, počítač by přestal pracovat. Můžete získat byte na jakékoli adrese použitím PEEK.

RAM znamená Random Access Memory - paměť s libovolným přístupem a obsahuje programy a informace, kterými je počítač plněn. Spectrum má 48kB RAM, tzn., že obsahuje 48x1024 (49152) bytů nebo adres. Libovolný přístup znamená, že byt na jakékoli adrese může být měněn, a to užitím klávesy POKE.

Adresy paměti jsou rozšířeny od 0 do 65535, první čtvrtina je ROM a zbytek RAM.

## Systém proměnných

Sloupec na str. 48 originálu ukazuje, jak jsou organizovány paměti počítače Spectrum. Zde můžete vidět, kde jsou odlišné různé části, které počítač řídí, umístěny. Několik jich může měnit polohu a jejich hranice jsou dány soustavou proměnných.

Systémové proměnné počítače nejsou takové proměnné, jako ty, které jsou použity v Basicu. Jsou to jednoduše jména jistých užitečných hodnot, které jsou umístěny na dílčích adresách nebo umístěních v paměti. Účelem těchto jmen je pomoci zapamatovat si význačnost hlavních hodnot uložených na místech.

Například systémová proměnná RAMTOP je adresa RAM. Tato oblast paměti udržuje program Basic a hodnoty jeho proměnných. Adresa RAMTOP je 23730.

Uspořádání paměti:

Grafiky tvořené uživatelem

Štosování GOSUB

Rezerva

Výpočtové štosování

Dočasný prostor pro práci

Vstupní údaje

Povely nebo vydaná vedení

Proměnné

Program Basic

Kanálové informace

Rozvržení microdrive

Systémové proměnné

Spoj pro tiskárnu

Atributy

Rozvržení displaye

Názvy v zeleném poli značí jména adres daných hranic.

#### 4. Naučte se programovací jazyk Sinclair Basic

Tato kapitola popisuje celý programovací jazyk Sinclair Basic. Naleznete v ní souhrn veškerých užitých klíčových slov a dále podrobnosti, jak jazyk Sinclair Basic pracuje. Informace udávají rozmezí potřebné pro nejjednodušší programování až do nejpokročilejšího programování v jazyce. Toto není kapitola, kterou je třeba číst od začátku do konce. Místo toho je to programovací slovník, který vám umožní nejširší využití schopností počítače Spectrum.

## Programovací referenční příručka pro klíčová slova jazyka Sinclair Basic

### Skupiny klíčových slov

Klíčová slova patří do jedné nebo více ze čtyř skupin.

#### Povely:

Klíčové slovo, které vyvolá činnost a které může být použito pro tvorbu přímého povelu. Je prováděno tím, že je založeno.

Př. - RUN, LOAD

#### Příkazy:

Klíčové slovo, které vyvolá činnost a které může být použito v programovém řádku. Je prováděno pouze tehdy, kdy program je v chodu.

Př. - DRAW, INPUT

#### Funkce:

Klíčové slovo, které vydává hodnotu určitého druhu.

Tvoří část povelu nebo příkazu.

Př. - RND, INT

#### Logické operátory:

Klíčové slovo, které je používáno pro vyjádření logiky v příkazu nebo povelu. Stanoví nebo mění pravdivost určitých podmínek. Spectrum má tři logické operátory - AND, OR a NOT.

Tato příručka obsahuje úplný popis všech klíčových slov v jazyce Basic vhodných pro ZX Spectrum+. Každé představení rysů obsahuje:

- umístění klíčového slova
- třídu klíčového slova
- účel klíčového slova
- užití klíčového slova
- programovací formát



Podrobnosti udané pro umístění, účely a užití jsou samovyvětlující. Třída a formát jsou komplexnější a pro nejefektivnější užití příručky musíte nejprve pročíst informace na str. 50-51 (tyto informace).

Čísla a proměnné:

Čísla - uložené s přesností 9 nebo 10 míst.  
Operace s čísly v rozsahu přibližně od  $10^{38}$  do  $4 \times 10^{-39}$ .

Přijímané proměnné - číslo jakékoli délky počínající písmenem. Mezery jsou vypouštěny a všechna písmena převáděna na malá písmena. Velká písmena a malá písmena nejsou rozlišena.

Jakékoli samostatné písmeno následované  $\$$  - string.  
Velká a malá písmena nejsou rozlišena.

Array - pro proměnné plochy nebo popisy viz vysvětlení DIM.

Formát klíčových slov:

Formát klíčových slov vyjadřuje syntaxi každého klíčového slova. Tzn. přesnou kombinaci klíčového slova a dalších faktorů, jako jsou hodnoty a proměnné. Následující zkratky jsou užívány při udávání formátu:

| Zkratka       | Vysvětlení   | Příklad           |
|---------------|--|-------------------|
| num-const     | Číselná konstanta - číslo  | 24,5              |
| num-var       | Číselná proměnná - proměnná, která může obsahovat číslo  | sum               |
| num-expr      | Číselná vyjádření - jakákoli platná kombinace číselných konstant, proměnných a klíčových slov, které udávají číslo | sumx24.5<br>RNDx7 |
| int-num-const | Číselná konstanta, proměnná  |                   |
| int-num-var   | nebo vyjádření, jehož hodnota  |                   |
| int-num-expr  | je zaokrouhlena na nejbližší celé číslo  |                   |

|              |   |                                 |
|--------------|---|---------------------------------|
| string-const | Konstanta textová nebo text (jakákoli kombinace typů, písmen uvnitř uvozovek) | "ZX Spectrum+"                  |
| string-var   | Textová proměnná, která může obsahovat text                                   | aš                              |
| string-expr  | Jakákoli platná kombinace textové konstanty, textu a klíčového slova          | aš+"ZX Spectrum+"<br>aš (6T08)  |
| letter       | Jakékoli velké nebo malé písmeno  | Y x                             |
| letterš      | Jakékoli velké nebo malé písmeno, za kterým následuje š                       | Bš aš                           |
| cond         | Jakákoli podmínka nebo podmínka v rámci podmínky                              | x=10ANDt<10                     |
| statement    | Jakýkoli příkaz v Basicu, který je platný společně s jiným příkazem           | IFT>10THEN STOP<br>PRINT INK2;x |
| []           | Volitelná položka, která může být opakována                                   |                                 |

Poznámka - termíny "číselná hodnota" a "hodnota textového řetězce" jsou užity v textu pro číselné nebo textové body.

Znaménka v Sinclair Basicu:

| znaménko | umístění        | činnost/užití  |
|----------|-----------------|--|
| §        | 4               | textová proměnná   |
|          | 7               | začátek nové řádky   |
| (        | 8               | otevření závorky   |
| )        | 9               | uzavření závorky   |
| <=       | Q               | je menší nebo rovno  |
| <>       | W               | není rovno   |
| >=       | E               | je větší nebo rovno  |
| <        | R               | je menší než   |
| >        | T               | je větší než   |
| ↑        | H               | umocněno   |
| -        | J               | odčítání/záporný   |
| +        | K               | sčítání/kladný/textové řetězce                                     |
| =        | L               | je rovno   |
| :        | Z               | odděluje příkazy v programovém ř.                                  |
| /        | V               | dělení   |
| ×        | B               | násobení   |
| .        | vlastní klávesa | desetinná tečka  |
| ;        | vlastní klávesa | zobrazí další sloupec, oddělí příkazy v rámci programového příkazu |
| "        | vlastní klávesa | otevívají a uzavírají textové řetězce                              |
| ,        | vlastní klávesa | zobrazí sloupce 0 nebo 16, odděluje hodnoty za klíčovými slovy     |

ZX Spectrum+ - sestava typů:

viz matice na str. 51

## ABECEDNÍ PŘEHLED KLÍČOVÝCH SLOV JAZYKA SINCLAIR BASIC

- ABS (absolutní hodnota)

Umístění klávesy: **EXTEND MODE G**

Funkce: Dává absolutní velikost číselné hodnoty, která je rovna dané číselné hodnotě bez kladného nebo záporného znaménka.

Užití: ABS musí být následováno číselnou hodnotou uzavřenou v závorkách, např.

50 LET x = ABS (y - z)

ABS vrací absolutní hodnotu číslu (y - z)

Příklad: Povel:

PRINT ABS - 34.2

Zobrazí se:

34.2

Formát: ABS num-const

ABS num-var

ABS (num-expr)

- ACS (arcus cosinus)

Umístění klávesy: **EXTEND MODE, SYMBOL SHIFT W**

Funkce: ACS vypočítá hodnotu úhlu z jeho cosinu.

Užití: ACS následováno číselnou hodnotou uzavřenou v závorkách, např.

60 LET x=ACS (y \* z)

Hodnota v závorce může být v rozsahu -1 až 1.

ACS poté převrací hodnotu úhlu v radiánech. Převod radiánů na stupně provedeme násobením výsledku číslem 180/PI.

Příklad: Povel:

PRINT 180/PI \* ACS 0.5

Zobrazí se:

60 - úhel ve stupních, který má cosinus=0.5

Formát: ACS num-const

ACS num-var

ACS (num-expr)

- AND ( logický součet )

Umístění klávesy: SYMBOL SHIFT Y

Logický operátor/funkce

AND je logický operátor testující pravdivost podmínek. Pouze tehdy, jsou-li všechny podmínky pravdivé, je celková kombinace pravdivá. AND rovněž převádí binární činnosti dvou číselných nebo textových proměnných.

Užití: Jako logický operátor, AND má dvě podmínky, kde je testována pravdivost, např.

```
90 IF x=z AND time<10 THEN PRINT "CORRECT"
```

Pouze jsou-li obě podmínky pravdivé, zobrazí se CORRECT. Jestliže některá z podmínek není splněna, celá kombinace neplatí a program v tomto případě pokračuje dalším řádkem.

Jako funkce může AND operovat se dvěma číselnými hodnotami, např.

```
50 LET x=y AND z
```

Zde vrací AND proměnné x hodnotu proměnné y v případě, že hodnota z není nulová a nebo přiřazuje do proměnné x hodnotu 0, jestliže druhá hodnota (z) je rovna nule.

AND může rovněž operovat s textovými řetězci. Číselná hodnota musí být zařazena vždy za AND, např.

```
50 LET a$=b$ AND z
```

AND přiřadí proměnné a\$ hodnotu b\$ v případě, že z není rovno nule. Je-li tomu tak, bude a\$=" "(prázdný text).

Zapamatujte si, že ZX Spectrum přiřazuje hodnotu 1 pravdivé podmínce a 0 jako nepravdu.

Příklady:

```
60 LET correct=(x=y+z) and time <10
```

```
70 LET score=score+10*(1 AND correct)
```

```
80 LET a$=("Out OF TIME OR NOT" AND NOT correct)+"Correct"
```

Jestliže dvě podmínky v řádku 60 jsou pravdivé, pak

číselná proměnná correct je označena jako hodnota 1. Poté je načítáno score o 10 a textová proměnná aš má tvar "Correct". Jestliže některá z podmínek neplatí, má correct hodnotu 0 a aš obsahuje zprávu "Out Of Time Or Not Correct".

Formát: cond AND cond  
num-expr AND num-expr  
string-expr AND num-expr

- ASN (arcus sinus)

Umístění klávesy: EXTEND MODE, SYMBOL SHIFT Q

Funkce

ASN vypočítává velikost úhlu z jeho sinu.

Užití: ASN je doprovázeno číselnou hodnotou. Vyjádření musí být uzavřeno v závorkách, např.

60 LET x=ASN (y\*xz)

Hodnota ASN je sinus požadovaného úhlu a může být v rozsahu -1 až 1. Výsledek je v radiánech. Převod radiánů na stupně-použijte koeficient 180/PI, kterým výsledek vynásobíte, např.

Povel:

PRINT 180/PI \* ASN 0.5

Zobrazí se:

30, tj. úhel odpovídající sinu 0,5

Formát: ASN num-const  
ASN num-var  
ASN (num-expr)

- AT

Umístění klávesy: SYMBOL SHIFT I

Viz INPUT; LPRINT; PRINT

- ATN (arcus tangens)

Umístění klávesy: EXTEND MODE, SYMBOL SHIFT E

### Funkce

ATN vypočítává velikost úhlu z jeho tangenty.

Užití: ATN je doprovázeno číselnou hodnotou. Vyjádření musí být uzavřeno v závorkách, např.

60 LET x=ATN (y \* z)

Hodnota parametru je tangenta požadovaného úhlu. ATN vrací hodnotu úhlu v radiánech. Pro převod radiánů na stupně násobte hodnotu ATN konstantou 180/PI.

Příklad: Povel:

```
PRINT 180/PI * ATN 1
```

Zobrazí se:

45, tj. úhel ve stupních, který má tangentu rovnu 1.

Formát: ATN num-const

ATN num-var

ATN (num-expr)

- ATTR (rysy, vlastnosti)

Umístění klávesy: EXTEND MODE, SYMBOL SHIFT L

### Funkce

ATTR udává kvalitu zadaného typu na obrazovce; barvu INK a PAPER, jas, blikání typu v dané poloze.

Užití: ATTR je doplněno dvěma parametry oddělenými čárkou a uzavřenými do závorek, např.

```
150 IF ATTR (v,h)=115 THEN GOSUB 2000
```

První hodnota v závorce může nabývat hodnoty od 0 do 23, odpovídající číslu řádku obrazového rastru. Druhý parametr (h) může být v rozmezí od 0 do 31 a udává číslo sloupce dané polohy. ATTR poté vrací číslo od 0 do 255. Toto číslo je součet rysů dané pozice, je tvořeno následovně: barevný kód inkoustu (0 až 7)

barva podkladu (osminásobek barevného kódu)

jas (64)

blikání (128)

Příklad: Jestliže typ v poloze 11,16 je zobrazen v barvě INK= 3 (ružová), barva podkladu je 6 (žlutá) a je světlá, nebliká, pak povel

```
PRINT ATTR (11,16)
```

zobrazí 115(3+8 \* 6 + 64 + 0).

ATTR ve dvojkové formě

ATTR vrací jeden byte, ve kterém bit č. 7 (nejvýznamnější) je jedna pro blikání nebo 0 pro běžné, bit 6 je 1 pro světlé a 0 pro běžné, bity 5-3 jsou barvy papíru (ve dvojkové) a bity 2-0 jsou barvy inkoustu.

Formát: ATTR (num-expr, num-expr)

- BEEP

Umístění klávesy: EXTEND MODE, SYMBOL SHIFT Z

Příkaz/povel

BEEP vydává tón z reproduktoru - daný čas a daný tón.

Užití: BEEP může být užíváno pro tvoření příkazů v programu nebo pro přímý povel. Je následováno dvěma číselnými hodnotami oddělenými čárkou, např.

80 BEEP x,y

První proměnná x může být v rozsahu od 0 do 10 a stanoví trvání tónu v sekundách. Druhá hodnota y může být v rozsahu od -60 do 69 a stanoví výšku tónu dané noty po pultónech pod středním C při záporném nebo nad středním C při kladném.

Příklad: Povel:

BEEP 0.5,1

Způsobí, že tón C# nad středním C zní půl sekundy.

Formát: BEEP num-expr, num-expr

- BIN (dvojkové číslo)

Umístění klávesy: EXTEND MODE B

BIN převrací binární číslo do desetinných čísel.

Užití: BIN je následováno dvojkovým číslem, složeným z 16 1 a 0, např.

50 POKE USR"a",BIN 1010

1010

BIN vrací desetinou hodnotu dvojkového čísla. Je používán společně s POKE a USR, jak je uvedeno výše pro tvoření uživatelem definovaných grafických typů s 1 značící přítomnost bodu grafiky v barvě inkoustu a 0 v barvě papíru.



Příklad: Povel:

```
PRINT BIN 11111110
```

zobrazí 254, dekadickou hodnotu binárního čísla.

Formát: BIN [1][0]

- BORDER

Umístění klávesy: B

Příkaz/povel

BORDER stanoví barvu okraje televizní obrazovky okolo displaye.

Užití: BORDER může být užíván jako přímý povel nebo jako příkaz v programu. Je následován číselnou hodnotou, např.

```
30 BORDER RND*7
```

Hodnota následující za BORDER je zaokrouhlena na nejbližší celé číslo a stanoví barvu okraje obrazu:

0 - černý

1 - modrý

2 - červený

3 - růžový

4 - zelený

5 - azurový

6 - žlutý

7 - bílý

BORDER rovněž seřizuje barvu papíru spodní poloviny obrazovky. Na rozdíl od klíčových slov INK a PAPER nemůže být příkaz BORDER založen do příkazu PRINT.

Formát: BORDER int-num-expr

- BRIGHT

Umístění klávesy: EXTEND MODE, SYMBOL SHIFT B

Příkaz/povel

BRIGHT způsobí, že typy, které jsou zobrazovány, mají světlejší barvu než normálně.

Užití: BRIGHT může být užíván jako přímý povel, avšak běžně je používán ve formě příkazu v programu. Je následován číselnou hodnotou, např.

80 BRIGHT 1

Hodnota následující za BRIGHT je zaokrouhlené nejvyšší celé číslo podle nutnosti a může být buď 0, 1 nebo 8. Hodnota 1 způsobí, že veškeré následně zobrazované typy pomocí PRINT nebo INPUT se objeví ve světlejší barvě inkoustu a papíru a hodnota 8 způsobí, že světlé polohy typů zůstanou světlé a normální polohy typů zůstanou normální při tom, co jsou do nich vytištěny nové typy. BRIGHT následováno 0 ruší oba BRIGHT 1 a BRIGHT 8, takže veškeré následně zobrazované typy jsou normální.

BRIGHT může být rovněž vloženo do příkazu displaye, který je tvořen pomocí klíčových slov PRINT, INPUT, PLOT, DRAW a CIRCLE. BRIGHT následuje klíčové slovo, avšak předchází údajům nebo parametrům displaye. Je následováno stejnými hodnotami a středníkem, např.  
50 PRINT BRIGHT 1;"WARNING"

Efekt klíčového slova BRIGHT je v tom, že jeho působení je místní a vztahuje se pouze na zobrazené typy, vytečkované body nebo vytažené čáry podle příkazu na displayi. BRIGHT 1 zesvětluje barvu papíru celého typového postavení 8x8 bodů, jestliže jakýkoli bod v dané poloze je vyznačen barvou inkoustu.

Formát: BRIGHT int-num-expr[;]

- CAT (katalog)

Povel při manipulaci s microdrivy, viz Microdrive a Interface 1 příručky.

- CHR $\mathcal{S}$

Umístění klávesy: EXTEND MODE, U

Funkce:

Typy a klíčová slova, která jsou dostupná na klávesnici a jakékoli uživatelem definované grafické typy, prováděné pomocí typů Spectra. Užitím CHR $\mathcal{S}$  a čísla kódu může být jedno každé jako string. Sestavení typů rovněž zahrnuje několik řídicích kódů, které ovlivňují zobrazení typů. Tyto kódy mohou být vnášeny do činnosti a do zobrazených typů použitím klíčového slova PRINT před CHR $\mathcal{S}$ . Kompletní souprava typů a čísel kódů může být nalezena na str. 51.

Užití: CHR $\mathcal{S}$  je následováno číselnou hodnotou, např.

```
80 PRINT CHR $\mathcal{S}$ x
```

Vyjádření musí být uzavřeno v závorkách. Hodnota následujících CHR $\mathcal{S}$  (x výše) je zaokrouhlené nejbližší celé číslo. Jestliže je v rozsahu 32-255, CHR $\mathcal{S}$  vrací typy klávesnice, uživatelsky definované grafické typy nebo klíčová slova jako text.

Spectrum používá kód ASCII pro hodnoty od 32 do 95 97 do 126. Je-li x označeno jako hodnota 65, výše uvedený příkaz označí např. A.

Řízení kódů: Hodnoty od 1 do 31 buď vrací řídicí kódy nebo nejsou tyto použity. CHR $\mathcal{S}$  6(PRINT comma), 8 (back-space) a 13(nový řádek nebo ENTER) narušují zobrazení na obrazovce, jsou-li začleněny do příkazu PRINT. CHR $\mathcal{S}$  může být následováno hodnotou kódu a středníku, např.

```
60 PRINT "A"; CHR $\mathcal{S}$ 6; "B"
```

Tento příkaz zobrazí       A

B

Další příklad použití CHR\$ jako řídicích kódů je tvoření složených textů, obsahujících tyto:

```
60 PRINT "A" + CHR$ 6 + "B" ,
```

má přesně stejný efekt, jako předchozí příklad. Kódy 16 až 23 ovlivňují barvu a polohu. Každá může být použita ve složeném stringu spolu s CHR\$ následovaným barevným kódem od 0 do 7 CHR\$16 (řízení barvy "inkoustu") a CHR\$17 (řízení barvy papíru), nebo 0 až 1 pro CHR\$ 18 až CHR\$21 (blikání, světlé nebo inverzní typy, OVER).

Povel:

```
PRINT CHR$ 16 + CHR$ 3 + CHR$ 17 + CHR$ 6 + CHR$ 18 +  
+ CHR$ 1 + "ZX Spectrum+"
```

zobrazí na obrazovce ZX SPECTRUM+ blikající v červené a žluté. Toto může být změněno s tím, že každé znaménko + může být nahrazeno středníkem.

CHR\$ 22 (AT řízení) je následován dvěma hodnotami CHR\$, které značí řádek a sloupec čísla. Povel

```
PRINT CHR$ 22 + CHR$ 11 + CHR$ 16 + CHR$ 42
```

zobrazí hvězdičky ve středu obrazovky.

CHR\$ 23 (řízení TAB) je rovněž následován dvěma hodnotami stejným způsobem. Druhá hodnota je běžně 0 a první hodnota dává polohu TAB. Povel

```
PRINT CHR$ 23 + CHR$ 16 + CHR$ 0 + CHR$ 42
```

zobrazí hvězdu napůl napříč obrazovkou.

Pouze tyto ovládací prvky jsou použitelné. Užití PRINT CHR\$ s klíčovým slovem o hodnotě vyšší než 164 jednoduše zobrazí dané klíčové slovo a nezanášejí jej do činnosti.

Formát: CHR\$ int-num-const[;] [+]

CHR\$ int-num-var[;] [+]

CHR\$ (int-num-expr) [;] [+]

- CIRCLE

Umístění klávesy: EXTEND MODE, SYMBOL SHIFT H

Povel/příkaz

Klíčové slovo CIRCLE umožňuje kreslit kruhy na obrazovce.

Užití: CIRCLE má tři parametry, oddělené čárkami, např.

80 CIRCLE x,y,z

Každý parametr se zaokrouhlí na nejbližší celé číslo CIRCLE pak kreslí kruh po grafické mříži s vysokým rozkladem v barvě INK. Prvé dva parametry x a y stanoví svislou a vodorovnou polohu, třetí (z) udává délku poloměru. Rozměry musejí být takové, aby kruh nevybočil přes okraje obrazu. Kruh je ovlivňován příkazy o barvě a může zahrnovat tyto příkazy stejně jako PLOT a DRAW.

Příklad: Povel:

CIRCLE 128,88,87

Objeví se kruh, který zabírá většinu obrazové plochy.

Formát: CIRCLE[příkaz;]int-num-expr, int-num-expr, int-num-expr

- CLEAR

Umístění klávesy: X

Příkaz/povel

CLEAR vymazává hodnoty všech proměnných, uvolňuje prostor v paměti, takže hodnoty zabírající místa na RAMTOP, horní adresy systémové oblasti BASIC. CLEAR může být použit pro znovunastavení RAMTOP.

Užití: CLEAR může být užíván jako přímý povel nebo může mít podobu programového příkazu. Nevyžaduje parametry, např.  
50 CLEAR

CLEAR vymaže hodnoty které jsou běžně přiřazeny všem proměnným, včetně polí. Rovněž vykonává CLS a RESTORE, čímž je "čištěna" obrazovka a je připravena pro příjem dat na první místo. Poloha PLOT je znovu nastavena do spodního levého rohu a všechny podprogramy smazány.

Klíčové slovo CLEAR není vyžadováno před předimenzováním oblastí, neboť DIM ruší stávající oblasti stejného jména. Slovo RUN provádí současně i CLEAR.

CLEAR a RAMTOP: Příkaz CLEAR může mít parametr. Například:  
CLEAR 65267

CLEAR pak provádí funkci mazání jako dříve a rovněž seřizuje RAMTOP, nejvyšší adresu systémové oblasti Basicu na zadanou hodnotu, tj. na adresu 65267. RAMTOP je nastaveno na adresu 65367 ( ZX Spectrum+) a leží pod oblastí vyhrazenou pro uživatelsky definované grafiky. Povel NEW maže paměť včetně RAMTOP, takže užití CLEAR pro nižší RAMTOP (o 100 bytů v předchozím příkladě) zahrnuje větší část paměti, která je imunní vůči slovu NEW. Zvýšení RAMTOP dává více prostoru pro Basic na úkor uživatelské grafiky. Podprogramy jsou pak umístěny na RAMTOP, s tím, že RAMTOP zůstává nad zásobníkem kalkulátoru.

Pro vyhledání RAMTOP vložte  
PRINT PEEK 23730+256\*PEEK 23731

Formát: CLEAR[num-expr]

- CLOSE #

Povel pro manipulaci s microdrivy - viz příručku pro  
užití Microdrive a Interface 1.

- CLS (Smazání obrazovky)

Umístění klávesy: V

Povel/příkaz

CLS čistí veškerý text a grafiku z oblasti displaye, pone-  
chává ho slepý v původní barvě papíru (pozadí).

Užití: CLS lze využít jako přímý povel nebo může mít formu  
příkazu v programu. Nevyžaduje parametry, např.

250 IF a\$= "NO" THEN CLS

Oblast displaye, avšak ne okraj, je pak vymazán  
na zvolenou barvu podle předešlého příkazu PAPER  
nebo povelu nebo na barvu papíru bílou.

CLS musí být užito po PAPER a před PRINT nebo ji-  
ným příkazem výpisu, aby bylo vytvořeno barevné  
pozadí po celé ploše obrazovky.

Formát: CLS

- CODE

Umístění klávesy: EXTEND MODE I

Funkce

CODE udává číslo kódu typu ve Spectru (viz str. 51).

Užití: CODE je následováno hodnotou textové proměnné,  
např.

90 IF CODE a\$<65 OR CODE a\$>90 THEN GOTO 80

Vyjádření textu musí být uvedeno v závorkách.

CODE vrací číslo kódu prvního typu v textu.

Je-li toto mezera (" "), pak CODE nabývá hodnoty 0.

### SAVE/LOAD/VERIFY CODE

CODE je používáno rozličnými způsoby - pomocí SAVE, LOAD a VERIFY.

Formát: CODE string-const  
CODE string-var  
CODE(string-expr)

### - CONTINUE (pokračování)

Umístění klávesy: C

Povel

Zastaví-li program, může být použito klíčového slova CONTINUE pro opětovné spuštění programu od místa, na kterém byl zastaven. Jestliže způsobila zastavení programu chyba, musí být nejprve napravena než klíčové slovo CONTINUE umožní pokračovat v programu.

Užití: CONTINUE je používáno jako přímý povel při zastavení programu. Nevyžaduje parametry. Poté, co bylo CONTINUE stisknuto, program normálně pokračuje na stejném příkazu, na kterém byl zastaven. Byl-li důvodem zastavení chybný údaj (chyba), je možno uložit opravu chyby a CONTINUE umožní pokračování programu od daného příkazu. Jestliže program zastavil na příkazu STOP, dávající zprávu 9, nebo byl-li zdržen z důvodu stisknutí klíčového slova BREAK, udání zprávy L, poté klíčové slovo CONTINUE pokračuje od dalšího příkazu. Je možno začlenit napravující povel podle potřeby.

Jestliže je použito klíčového slova CONTINUE pro obnovu přímého povelu, pak dojde ke smyčce, jestliže povel nezastaví první příkaz povelu. Display zmizí, avšak ovládání může být obnoveno stisknutím BREAK. CONTINUE udává zprávu 0, jestliže zastavil povel na druhý příkaz a zprávu N při třetím nebo následném příkaze.

Formát: CONTINUE



- COPY (kopie)

Umístění klávesy: Z

Povel

COPY dává povel tiskárně Sinclair vyrobit kopii displaye obrazovky.

Užití: COPY je užíváno jako přímý povel při dokončení programu nebo jeho zastavení. Nevyžaduje parametry.

Po povelu COPY za předpokladu, že je připojena tiskárna, bude vytištěno 22 prvních řádků obrazovky.

Veškeré barvy inkoustu (tisku) jsou tištěny černě; barvy papíru (pozadí) nejsou tištěny. Tiskárna může být zastavena stisknutím BREAK.

Jestliže se na obrazovce objeví výpis programu, může být tištěn použitím COPY s tím, že byl vyvolán pomocí povelu LIST.

Výpis se objeví na obrazovce po stisknutí ENTER poté, co byl program dokončen nebo zastaven, avšak tento "automatický" výpis nemůže být tištěn pomocí příkazu COPY.

Formát: COPY

- COS (cosinus)

Umístění klávesy: EXTEND MODE W

Funkce

COS udává cosinus úhlu.

Užití: COS je následován číselnou hodnotou, např.

140 LET x= COS y

Vyjádření musí být uzavřeno v závorkách. Hodnota následující COS je úhel v radiánech. COS se pak vrací do cosinu úhlu. Stupně mohou být převáděny do radiánů násobením  $\text{PI}/180$ .

COS vrací zápornou hodnotu úhlů od 90 do 270 stupňů a kladnou pro úhly od 0 do 90 stupňů a 270-360 stupňů.

Příklad: Povel

PRINT COS (60\*PI/180)

zobrazí 0.5 cosinus 60 stupňů

Formát: COS num-const  
COS num-var  
COS(num-expr)

- DATA

Umístění klávesy: EXTEND MODE D

Příkaz

DATA zajišťuje výpis pozic údajů programu. Tyto položky mohou být hodnoty proměnných nebo textů, které mají být zobrazeny. Každá hodnota je označena jako proměnná pomocí příkazu **READ**.

Označení je prováděno podle toho, v jakém pořadí se jednotlivé pozice údajů objevují v programu, avšak může být použito RESTORE pro začátek označování na první pozici daného příkazu DATA.

Užití: DATA může být použito pouze pro vytvoření příkazu v programu. Normálně je následováno výčtem číselných nebo textových konstant, každá oddělená čárkou, např.

```
50 DATA 31,"JAN", 28,"FEB"
```

Každá konstanta je pak označena jako proměnná pomocí příkazu READ, který čte DATA. Příkaz DATA může být umístěn kdekoli v programu. Číslo, druh (číselná nebo textová) a pořadí konstant musí odpovídat počtu provádění příkazu READ a druh a postup - pořadí proměnných v příkazu READ. Výpis dat může být rozdělen do několika příkazů DATA, je-li zde příliš mnoho pozic pro naplnění jednoho příkazu.

Příklad: viz program na str. 56 pravý sloupec nahoře

Tento program zobrazí, že leden má 31 dnů a únor 28 dnů.

Použití proměnných s klíčovým slovem DATA

Údaje DATA v příkazu DATA mohou sestávat z číselných nebo textových proměnných nebo vyjádření, udávající proměnná, mají předem označené hodnoty. U výše uvedeného příkladu příkaz DATA může být změněn na

```
50 DATA d,m$ ,d-3,"FEB"
```

Je-li předem označeno  $\bar{d}$  jako hodnota 31 a  $\bar{m}$  hodnota "JAN", je dán stejný display.

LOAD DATA, SAVE DATA a VERIFY DATA:

Klíčového slova DATA lze rovněž použít s klíčovými slovy LOAD, SAVE a VERIFY pro uschování oblastí na pásek.

Viz LOAD DATA, SAVE DATA a VERIFY.

Formát: DATA num-expr [,num-expr][,string-expr]

DATA string-expr [,num-expr][,string-expr]

- DEF FN (definice funkce)

Umístění klávesy: EXTEND MODE, SYMBOL SHIFT 1

Příkaz

DEF FN umožňuje uživateli stanovit funkci, která není dostupná jako klíčové slovo. Velké množství parametrů může být zahrnuto do funkce jako FN příkaz, který vyvolává funkci a může vrátit buď číselnou nebo textovou hodnotu jako výsledek.

Užití: DEF FN lze použít pouze jako příkaz v programu.

Má-li být definována číselná funkce, je DEF FN následováno jakýmkoli samostatným písmenem a poté jednou nebo více číselnými proměnnými, každé oddělené čárkou a uzavřené závorkami, např. DEF FN r(x,y). Potom následuje znaménko = a poté číselné vyjádření obsahující proměnné, např.

1000 DEF FN r(x,y)=SQR(x $\uparrow$ 2+y $\uparrow$ 2)

Písmeno následující DEF FN (r) je jméno, které identifikuje funkci. Proměnné mohou být rovněž jednoduchá písmena. V obou případech Spectrum nerozlišuje mezi malými a velkými písmeny.

Vyjádření, které následuje za znaménkem = používá proměnné (x,y) na definice funkce.

Příkaz definice funkce může být umístěn kdekoli v programu. Pro vyvolání funkce, která jej definuje, je použito příkazu FN. Poté následuje písmeno označující funkci a výpis číselných hodnot, každé oddělené čárkou a uzavřené do závorek, např.

```
50 PRINT FN r(3,4)
```

Hodnoty uvnitř závorek prochází do funkce ve stejném pořadí, jako proměnné v příkazu DEF FN. Tedy v tomto případě je x označeno hodnotou 3 a y hodnotou 4. FN vyhodnocuje vyjádření a vrací hodnotu.

DEF FN může být rovněž následováno písmenem a dvojicí samostatných závorek, např.

```
1000 DEF FN r( )=INT(x+0.5)
```

Hodnota přesně označená jako proměnná (x) prochází funkcí, je-li vyvolána FN. V tomto případě FN r( ) vrací hodnotu přesně označenou jako x zaokrouhlenou na nejbližší celé číslo.

DEF FN a textové řetězce:

DEF FN a FN lze rovněž použít stejným způsobem pro definici a vyvolání funkcí textů. V tomto případě jméno funkce tvoří jednoduché písmeno, za kterým je uvedeno znaménko § a jedna nebo více proměnných v příkaze je písmeno následováno znaménkem §. Odpovídající vyjádření textu tvoří definici, např.

```
1000 DEF FN a§(b§,x,y)=b§(xT0y)
```

Vyjádření textu následující za znaménkem = - v tomto případě je § přiřazení a x a y jsou první a poslední písmena oddílu b§. FN musí být následováno jménem funkce a v závorkách hodnotou textové proměnné spolu s veškerými dalšími parametry, které budou procházet funkcí. V tomto případě je povel:

```
PRINT FN a§ ("FUNDAMENTAL",1,3)
```

Zobrazí se: FUN

a povel:

```
PRINT FN a§ ("FUNDAMENTAL",5,8)
```

Zobrazí se AMEN

Formát: DEF FN písmeno([písmeno][,písmeno])= num-expr  
DEF FN písmeno §([písmeno §][písmeno][,písmeno]  
[,písmeno §]) = string-expr  
FN písmeno([num-expr][,num-expr])  
FN písmeno §([string-expr][num-expr][,num-expr][,string-  
expr])

- DIM (rozměr)

Umístění klávesy: D

Příkaz

DIM je užíváno na dimenzování (nastavení) daného počtu číselných nebo textových proměnných. Oblast je výpis proměnných stejného jména, které jsou vyjádřeny pomocí subskripcí (hodnot, které označují každou proměnnou nebo prvek oblasti).

Užití: DIM je užíváno pro vytváření příkazů v programu.

Je následováno jednoduchým písmenem, které pojmenovává pole a jedním nebo více číselnými hodnotami oddělenými čárkou a uzavřeno do závorek, např.

20 DIM x(10)

80 DIM z(20,5)

V prvním případě je jednorozměrové číselné pole vytvářené deseti prvky s označením od jedné do deseti. Pole má název x a označené proměnné jsou x(1) až x(10) včetně. Jakákoli existující pole stejného jména jsou vymazána a proměnné jsou každá označena hodnotou 0. V určování velikosti pole počítač nerozlišuje mezi nápisy velkých a malých písmen - proměnná x(2) je stejná jako X(2). Avšak jednoduché číselné proměnné, které mají stejné písmeno, jako název oblasti (x nebo X) spolu mohou koexistovat a mohou být samostatně podle potřeby použity. Počet hodnot v závorkách je rovný počtu rozměrů vytvořených v číselném poli. Druhý příklad uvádí dvourozměrové pole o 100 prvcích s 20 prvky v prvním rozměru a s 5 ve druhém. Tyto prvky jsou číslovány z(1,1) až z(20,5).

Pole může být vytvořeno z jakýchkoli čísel a rozměrů. Prvky číselných polí mohou následně být stanoveny názvem pole, za kterým je hodnota v závorkách, např.

70 PRINT x(a)

160 PRINT z(7,b)

DIM a textové pole:

DIM je používáno stejným způsobem jako u číselných polí, s výjimkou toho, že samostatné písmeno je následováno znakem  $\$$  pro název pole. Navíc je třeba udat zvláštní hodnoty pro rozměr hodnot v závorkách, aby bylo možno určit délku každého textu. Např.

30 DIM a\$(20,5)

90 DIM b\$(20,5,10)

První příkaz vytváří pole o 20 prvcích, každý obsahuje text o 5 typech. Poznačené proměnné jsou pojmenovány a\$(1) až a\$(20) včetně a jsou počátečně označeny nulou (prázdný) text (""). Veškeré stávající oblasti stejného jména jsou vymazány a navíc na rozdíl od číselných polí spolu jednoduché proměnné textu nemohou koexistovat.

Druhý příklad vytváří dvourozměrné textové pole o 100 prvcích s 20 prvky v prvním rozměru a 5 ve druhém.

Veškeré prvky mají délku 10 typů.

Když jsou textové proměnné postupně označeny do textového pole, jsou zvětšeny volnými místy na konci textu nebo označeny definovanou délkou podle potřeby.

Prvky textového pole jsou označeny jménem pole, které je následováno (jedním nebo více) údaji o číselných hodnotách, udávajících popisná čísla. Např. prvek a\$(2) může být "SMITH" a prvek b\$(12,4) může být "DERBYSHIRE".

Pro definici hlavního typu v textu mohou být přidány zvláštní hodnoty. V těchto případech a\$(2,2) bude "M" (druhý typ v "SMITH") a b\$(12,4,5) bude "Y".

Nulová textová pole:

Je možné vytvářet nulová textová pole použitím pouze jedné hodnoty v závorkách, např.

10 DIM c\$(15)

Toto pole má pouze jeden prvek, který je c\$, a jehož délka je fixována na stanovenou hodnotu (15 typů).

Formát: DIM letter(num-expr[,num-expr])

DIM letter\$( num-expr[,num-expr])

- DRAW

Umístění klávesy: W

Příkaz/povel

DRAW je užíváno pro kreslení přímých čar a křivek na obrazovce.

Užití: DRAW je normálně užíváno pro vytváření příkazů v programu. Jestliže je vyžadována přímá čára, je následováno dvěma číselnými hodnotami oddělenými čárkou, např.

40 DRAW x,y

Přímá čára je poté vytažena na vysokorozkladové grafice z polohy stanovené předešlým příkazem PLOT nebo polohou dosaženou předešlým příkazem DRAW, podle toho, která je poslední. Obě hodnoty následující po povelu DRAW jsou zaokrouhleny na nejbližší celé číslo, je-li nezbytné. První hodnota (x výše) stanoví horizontální vzdálenost od této polohy, druhá hodnota (y) vertikální vzdálenost. Tyto hodnoty jsou záporné, běží-li čáry nahoru doleva nebo dolů a docílená poloha musí být uvnitř oblasti displaye.

Není-li předchozí příkaz PLOT nebo DRAW, povel DRAW vychází z polohy 0,0 (spodní levý roh obrazovky).

DRAW je ovlivňováno příkazy nebo povely barevných kódů a může zahrnovat příkazy se stejnými efekty jako s PLOT a CIRCLE.

Kreslení zakřivených čar:

Za klíčovým slovem DRAW může být uvedena třetí hodnota, která kreslí křivku, která je částí kruhu, např.

40 DRAW x,y,z

Třetí hodnota (z výše) stanoví úhel (v radiánech), pod kterým se čára křiví podle toho, jak je vytahována.

Čára zatáčí doleva, je-li hodnota kladná, a doprava, je-li záporná. Hodnoty PI nebo -PI vytváří kruh.

Příklad: Následující program nakreslí trojúhelník:

```
10 PLOT 127,150
20 DRAW 70,-100
30 DRAW -140,0
40 DRAW 70,100
```

Navíc přidání 1 nebo -1 k DRAW způsobí, že se strany zakříví vně nebo dovnitř.

Formát: DRAW [statement;]int-num-expr[,int-num-expr[,int-num-expr]

#### - ERASE

Pro použití při microdrivech, viz. Microdrive a Interface 1 manuálu.

#### - EXP (exponent)

Umístění klávesy: EXTEND MODE X

Funkce

EXP je matematická funkce, která povyšuje exponent e na danou mocninu.

Užití: EXP je následován číselnou hodnotou, např.

```
60 LET y= EXP x
```

Vyjádření musí být uzavřeno do závorek. EXP poté vrátí exponent e povýšený na argument (x uvedený výše).

Příklad: Povel

```
PRINT EXP 1
Zobrazí 2.7182818, hodnota přirozeného čísla e.
```

Formát: EXP num-const

EXP num-var

EXP (num-expr)

#### - FLASH

Umístění klávesy: EXTEND MODE, SYMBOL SHIFT V

Příkaz/povel

FLASH způsobí, že poloha typu bliká a to tím, že barva inkoustu a papíru se mění v konstantním poměru.



**Užití:** FLASH může být používáno jako přímý povel, avšak je běžně používáno pro tvoření příkazů v programu. Je následováno číselnou hodnotou, např.

50 FLASH 1

Hodnota následující za FLASH je zaokrouhlené nejbližší celé číslo podle nutnosti, a může být tedy 0, 1 nebo 8. Hodnota 1 způsobí, že veškeré následně zobrazené typy pomocí PRINT nebo INPUT blikají. Hodnota 8 vyvolává blikání poloh typů poté, co normální typy zůstanou normální, když jsou do těchto míst vytištěny. Uvedením 0 za povel FLASH ruší FLASH 1 a FLASH 8, takže veškeré typy jsou obyčejné. FLASH může být rovněž vloženo do příkazu po utváření displaye pomocí PRINT, INPUT, PLOT, DRAW a CIRCLE. FLASH následuje klíčové slovo, avšak předchází údajům nebo parametrům displaye; je následováno stejnými hodnotami a případně středníky, např.

120 PRINT FLASH 1;INK 2; PAPER 6; "WARNING"

Účinek blikání je poté místní a vztahuje se pouze na typy, které jsou zobrazeny, vytyčená místa nebo vytažené čáry podle příkazu displaye. FLASH 1 ovlivňuje blikání celého pole 8x8 a to **tehdy**, když je jakýkoli vytyčený bod v barvě inkoustu.

**Formát:** FLASH int-num-expr[;]

- FN (funkce)

Umístění klávesy: EXTEND MODE, SYMBOL SHIFT 2

Funkce

FN vyvolává funkce stanovené uživatelem. Je vždy používáno spolu s DEF FN, které definuje vyvolávanou funkci.

Užití: Má-li být vyvolána číselná funkce, je za FN uvedeno písmeno a poté pár závorek. Má-li být do funkce vložen jakýkoli parametr, je každý oddělen čárkou a uzavřen do závorek, např.

```
170 LET x=FN r(3,4)
```

Parametr (3,4 výše) poté prochází do funkce nazývané r. FN poté vrací výsledek. Nejsou-li žádné parametry zadány, dvojice závorek musí být přesto přítomna, např.

```
70 PRINT FN r()
```

V tomto případě používá funkce hodnot přesně označených k proměnným.

FN vyvolává funkci textu stejným způsobem s výjimkou toho, že vždy musí být přidáno znaménko § po písmenu označujícím funkci. (viz DEF FN - podrobnosti)

Formát: FN letter([num-expr][,num-expr])

```
FN letter§([string-expr][num-expr][,num-expr][,string-expr])
```

- FOR

Umístění klávesy: F

Příkaz/povel

FOR je vždy užíváno s klíčovými slovy TO a NEXT pro vytvoření smyčky FOR NEXT. Tato struktura umožňuje opakovat část programu po daný počet obrátek.

Užití: FOR vždy tvoří příkaz spolu s TO. FOR je následováno písmeny, rovnítkem a poté dvěma numerickými hodnotami oddělenými pomocí TO, např.

```
60 FOR a=1 TO 9
```

Písmeno a tvoří řídicí proměnnou. Příkazy, které mají být opakovány, následují a jeden nebo více jich běžně tvoří řídicí proměnnou.

Smyčka potom končí příkazem NEXT, kdy NEXT je následováno řídicí proměnnou, např.

90 NEXT a

Při provádění klíčové slovo FOR ruší veškeré proměnné stejného jména jako je řídicí proměnná a označuje ji počáteční hodnotou rovnou předchozí hodnotě klíčového slova TO (1 výše). Příkazy jsou poté prováděny s řídicí proměnnou, která má tuto hodnotu. Při dosažení klíčového slova NEXT je hodnota řídicí proměnné povýšena o 1. Je-li tato hodnota nižší nebo rovna hodnotě po klíčovém slově TO (omezující hodnota 9 výše), program se vrací na příkaz FOR a poté je opakována smyčka FOR NEXT. Je-li řídicí proměnná vyšší než omezující hodnota, poté smyčka končí a program pokračuje s dalším příkazem následujícím za FOR NEXT.

V předchozím případě je smyčka opakována 9x s řídicí proměnnou vzrůstající od 1 do 9. Při opouštění smyčky má hodnotu 10.

Spectrum nerozlišuje při pojmenování proměnných mezi velkými a malými písmeny.

Použití klíčového slova STEP a smyček FOR NEXT:

Klíčové slovo STEP je klíčové slovo, které může být začleněno do příkazu FOR, jestliže má řídicí proměnná být zvyšována o jinou hodnotu než 1 nebo má-li klesat. STEP následuje za omezující hodnotou a je následováno číselnou hodnotou, např.

60 FOR a=1 TO 9 STEP 2

Řídicí proměnná se zvyšuje po krocích o hodnotě dané (2 výše), pokud není vyšší než limitní hodnota. Řídicí proměnná má tedy v tomto případě hodnoty 1, 3, 5, 7 a 9 a smyčku opouští s hodnotou 11.

Záporný krok způsobí, že hodnota řídicí proměnné klesá.

V tomto případě musí být počáteční hodnota vyšší než konečná hodnota, smyčka končí, když hodnota řídicí proměnné poklesne pod omezující hodnotu, např.

60 FOR a=9 TO 1 STEP -1

Hodnota klesá od 9 do 1 a opouští smyčku s hodnotou 0.

Prolínající se smyčky:

Jedna nebo více smyček FOR NEXT mohou být umístěny navzájem do druhé postupem nazývaným "zahnížděné" smyčky. Postup řídicích proměnných v příkazech NEXT musí být reverzní proti postupu řídicích proměnných u příkazů FOR. Smyčky FOR NEXT mohou být začleněny jedna do druhé do jakéhokoli počtu podle toho, kolik smyček je požadováno pro daný úkol.

Formát: FOR letter= num-expr TO num-expr[STEP num-expr]  
NEXT letter

- FORMAT

Povel při užití microdrivů - viz příručka pro Microdrive a Interface 1.

- GOSUB

Umístění klávesy: H

Příkaz/povel

GOSUB dává programu povel nebo příkaz pro provádění podprogramu, který je samostatnou oddělenou částí programu. Toto je užitečné, je-li podprogram vyžadován v jednom programu několikrát.

Užití: GOSUB může být užíván jako příkaz nebo přímý povel, je následován číselnou hodnotou, např.

GOSUB 1000

Při vykonávání se hodnota následující GOSUB (1000) zaokrouhlí na nejbližší celé číslo a program vykonává podřádku po této hodnotě. Použití proměnné nebo vyjádření umožňuje programu vyhledat podprogram na propočítané řádce. Jestliže neexistuje číslo řádku, program stále vyhledává a pokračuje s prvním příkazem, který je **zadán**.

Podprogram končí slovy RETURN, a program se poté vrátí zpět do příkazu, který následuje za příkazem GOSUB.

Podprogramy mohou být uloženy jeden do druhého, takže jeden je dosažitelný z druhého, kdy v tomto případě klíčové slovo RETURN zasílá program zpět do příkazu následujícího za posledním prováděným příkazem GOSUB.

#### Zásobník GOSUB:

Kdykoli je prováděn příkaz GOSUB, jeho číslo řádku je umístěno do zásobníku GOSUB v paměti. Jsou-li prováděny dva nebo více podprogramů před povelem RETURN, jejich číselná označení se ukládají tak, že poslední číslo je na horní části zásobníku. Klíčové slovo RETURN vždy vybírá horní řádek čísla ze zásobníku a vrací se na tento řádek k pokračování programu.

Chyba 4 (mimo paměť) se může objevit tehdy, není-li dostatek příkazů pro RETURN.

Formát: GOSUB int-num-expr

#### - GOTO

Umístění klávesy: G

Příkaz/povel

Klíčové slovo GOTO dává programu pokyn pro vyhledání daného řádku.

Užití: GOTO může být užíváno jako přímý povel pro chod programu z daného řádkového čísla bez předchozího vymazání obrazovky. Rovněž může být používáno pro vytvoření příkazu v programu. GOTO je následováno číselnou hodnotou, např.

```
60 GOTO 350
```

Při provádění je hodnota následující za GOTO zaokrouhlena na nejbližší celé číslo a program se větví k číslu řádku, které má tuto hodnotu.

Užití proměnné nebo vyjádření umožňuje programu, aby se větvil na vypočítaný řádek. Neexistuje-li řádek, program stále pokračuje v rozvětvení

a začíná s prvním příkazem, který je do něho uložen.

Formát: GOTO int-num-expr

- IF

Umístění klávesy: U

Příkaz/povel

IF je vždy užíváno s klíčovým slovem THEN pro rychlé rozhodnutí, které ovlivňuje následnou činnost. Aby toto provedl, počítač zkouší nalézt něco, zda je to nebo není pravdivé. Je-li to pravdivé, poté následuje jeden směr činnosti. Není-li, nastupuje druhý.

Užití: IF běžně tvoří příkaz spolu s THEN. IF je nejprve následováno běžnou hodnotou, či podmínkou a za druhé buď THEN a jedním nebo více platnými příkazy Basic, např.

```
80 IF x THEN GOTO 250
```

```
240 IF a$="NO" THEN PRINT "THE END":STOP
```

Konstanty, proměnné nebo vyjádření (jako x výše) jsou považovány za pravdivé, mají-li nenulovou hodnotu. V tomto případě je příkaz následující za THEN a veškeré další příkazy ve stejném řádku proveden. Program poté pokračuje na dalším řádku. Jestliže je hodnota nulová, je konstanta, proměnná nebo vyjádření považováno za nepravdivé. Následující příkazy proto nejsou provedeny a program přeskočí na další řádek. V příkladu nepůjde program do ř. 250, bude-li x=0. Je-li podmínka (a\$="NO") po IF pravdivá, příkazy následující za THEN budou provedeny. Je-li podmínka nepravdivá, program se pohybuje na další řádek. V tomto případě, jestliže a\$ má hodnotu "NO", poté je zobrazeno "THE END" a program se zastaví. Jestliže a\$ má jinou hodnotu, program pokračuje od dalšího řádku.

Spectrum dává rozhodovací podmínky 1- pravdivá a 0-nepravdivá. Rozlišuje jakékoli nenulové číslo jako pravdivé a 0 jako nepravdivé. Proměnné mohou být označeny hodnotou podmínky pomocí příkazu jako:

```
70 LET x=a&="NO"
```

Na rozdíl od jiných programovacích jazyků Basic nemůže být THEN před GOTO vypuštěno.

Formát: IF num-expr THEN statement[:statement]  
IF cond THEN statement[:statement]

- IN

Umístění klávesy: EXTEND MODE, SYMBOL SHIFT I

Funkce

IN kontroluje stav klávesnice a další vstupní a výstupní zařízení. Čte byty z daných adres, které označují stav zařízení připojeného k portům.

Užití: IN je následováno číselnou hodnotou, např.

```
150 LET x=IN y
```

Hodnota následující za IN může být v rozsahu od 0 do 65535 a určuje adresu portu, který má být čten. IN poté udává byt čtený z tohoto portu.

Klávesnicové adresy:

Klávesnice má 8 adres, každá může obsahovat 1 nebo 5 různých bytů v závislosti na klíči, který je stisknut. Adresy jsou 65278, 65022, 64510, 63486, 61438, 57342, 49150 a 32766.

Hodnoty bytů těchto adres mohou být 175, 183, 187, 189 nebo 190.

Formát: IN num-const  
IN num-var  
IN (num-expr)

- INK

Umístění klávesy: EXTEND MODE, SYMBOL SHIFT X

Povel/příkaz

INK určuje barvu tisku, ve které je typ zobrazován, vytyčovány body a vedený čáry a křivky.

Užití: INK může být používán jako přímý povel, avšak běžně je používán ve formě příkazu v programu. Je následován číselnou hodnotou, např.

70 INK x

Hodnota následující za INK je zaokrouhlena na nejbližší celé číslo a může být v rozsahu od 0 do 9.

Následující barvy jsou udány pro barvu tisku:

0 - černá

1 - modrá

2 - červená

3 - růžová

4 - zelená

5 - zeleno-modrá

6 - žlutá

7 - bílá

8 - transparentní

9 - kontrastní černá nebo bílá

INK 8 určuje, že stávající barva zůstává nezměněna v jakékoli pozici na obrazovce, kde je použito klíčového slova 8. INK 9 způsobí, že barva bude buď černá nebo bílá tak, aby se objevila v protipoloze barvy papíru (pozadí).

Globální a lokální barvy :

Jestliže INK tvoří samostatný příkaz jak je uvedeno výše, barva je celková a veškeré dílčí displye se objevují v této barvě. INK může být rovněž vložen do displye ve formě příkazu pomocí PRINT, INPUT, PLOT, DRAW a CIRCLE. INK následuje klíčové slovo, avšak předchází údaj nebo parametry displye; je následován stejnými hodnotami a středníky, např.

60 CIRCLE INK 4; 128,88,87



Vliv INK je potom místní a vztahuje se pouze na zobrazený typ, vytyčený bod nebo vytaženou čáru podle příkazu displaye, v tomto případě namalování zeleného kruhu.

Poté se barva INK **převrací** na všeobecnou barvu nebo na **opak černé barvy**.

Formát: INK int-num-expr[;]

- INKEY\$ (vstupní klíč pro string)

Umístění klávesy: EXTEND MODE N

Funkce

INKEY\$ je používán pro stanovení stisknutí klávesy na klávesnici.

Užití: Nevyžaduje argumenty a je všeobecně užíván pro **označování** typů pro textové proměnné nebo pro zkoušení dílčích typů, např.

```
70 LET a$=INKEY$
```

```
130 IF INKEY$="N" THEN STOP
```

Při provádění vrací INKEY\$ typ udaný klíčem, který byl stisknut. Nebyl-li stisknut žádný klíč, **INKEY\$ vrací prázdný text** (""). Klíčové slovo INKEY\$ rozlišuje mezi velkými a malými písmeny a jejich posunutými a neposunutými typy. (Používejte IN pro stanovení veškerých kláves bez rozlišování typů.)

Na rozdíl od INPUT **nečeká INKEY\$, avšak** ihned pokračuje na další příkaz. Proto je běžně umísťován uvnitř smyčky, která se opakuje, dokud není požadovaný klíč stisknut.

Příklad: **Tato linka** podmiňuje operaci, pokud Y není stisknuto (bez CAPS SHIFT nebo CAPS LOCK).

```
60 IF INKEY$<>"y" THEN GOTO 60
```

Formát: INKEY\$

- INPUT

Umístění klávesy: I

INPUT umožňuje, aby byly údaje zahrnuty do programu v průběhu jeho chodu.

**Užití:** INPUT tvoří příkaz v programu a je používán velice obdobně jako PRINT. U jeho nejjednodušší formy je následován číselnou nebo textovou proměnnou, např.

```
60 INPUT x
```

```
90 INPUT a$
```

Počítač poté čeká, dokud číslo nebo text není zadán. Hodnota je zobrazena na začátku spodní řádky podle zaklíčování. Po stisknutí ENTER je hodnota **označena pojmenovanou proměnnou** a program pokračuje.

Příkaz INPUT může zahrnovat více než jednu proměnnou a zobrazí utvořené typy okamžitě. To je prováděno přesně stejným způsobem, jako u klíčového slova PRINT užitím uvozovek pro uzavření typů a středníků nebo čárek podle nutnosti pro oddělení pozic. Příkazy display jako INK, FLASH a PAPER mohou být rovněž použity, např.

```
80 INPUT INK 2; "What is your name?";n$, ("How old are you,"+n$+"?");age
```

(jak se jmenuješ, kolik ti je let, věk)

Odlišnosti od klíčového slova PRINT: INPUT čeká, když dojde k proměnné, takže proměnné a výrazy (viz výše uvedené n\$), které musí být začleněny **přímé** musí být uzavřeny v závorkách. Display začíná na začátku spodního řádku a poté se posouvá vzhůru, když je použito více než jedné řádky. AT může být použito při příkazu INPUT stejným způsobem jako s PRINT. AT 0,0 zobrazí na začátku řádku nad **spodní čárkou** a display se posouvá vzhůru, je-li použito více než dvou řádek.

**Zastavení INPUT:** Jestliže je INPUT následován číselnou proměnnou a je vloženo STOP, poté se program zastaví. S textovou proměnnou je

první uvozovka, která se objeví, **uvažována jako** vymazaná a poté vložené STOP zastaví program.

**Užití INPUT s LINE:** INPUT LINE lze použít pouze s textovými proměnnými. Běžně INPUT spolu s proměnnou textovou vyvolá zobrazení dvojice uvozovek. Jakmile je text zakódován, objeví se mezi uvozovkami. Pro odstranění těchto uvozovek použijte INPUT LINE následované textovou proměnnou. Jestliže je použito přímého dotazu, je umístěn mezi INPUT a LINE, např.

```
70 INPUT "Jak se jmenuješ?"; LINE a$
```

**Formát:** INPUT prompt [;][,][']num-var  
INPUT prompt [;][,][']string-var  
INPUT prompt [;][,][']LINE string-var[prompt]=[string-const][string-expr][AT int-num-expr,int-num-expr] [statement][;][,][']

- INT (Integer)

**Umístění klávesy:** EXTEND MODE R

**Funkce**

INT mění necelá čísla (čísla, která nejsou celá) na celá čísla nebo celistvá čísla.

**Užití:** INT je následováno číselnou hodnotou, např.

```
70 LET x=INT y
```

Výraz musí být uzavřen v závorkách. INT se poté vrací s hodnotou zaokrouhlenou dolů na celé číslo.

**Příklad: Povel**

```
PRINT INT 45.67,INT -7.66  
Zobrazí 45 -8
```

**Formát:** INT num-const  
INT num-var  
INT (num-expr)

- INVERSE

Umístění klávesy: EXTEND MODE, SYMBOL SHIFT M

Povel/příkaz

INVERSE převrací barvy na místech typů tak, že barva tisku se stává barvou papíru a naopak.

Užití: INVERSE se běžně užívá pro tvoření příkazů v programu. Je následován číselnou hodnotou, např.

```
70 INVERSE 1
```

Hodnota následující INVERSE je zaokrouhlena na nejbližší celé číslo a může být tedy 0 nebo 1. INVERSE 1 dává příkaz ke změně všech následných zobrazení prováděných klíčovým slovem PRINT a INPUT v inverzních barvách. INVERSE 0 znovu převádí převrácené barvy na původní.

INVERSE může být založen do příkazu displaye stejnou cestou jako příkaz INK. Je-li však použit s CIRCLE, PLOT nebo DRAW, INVERSE 1 tvoří čáru nebo bod, který je vytyčen na obrazovce v barvě podkladu, takže tento mizí.

Formát: INVERSE int-num-expr

- LEN (délka textu)

Umístění klávesy: EXTEND MODE K

Funkce

LEN udává délku textu

Užití: LEN je následováno **hodnotou délky textu**, např.

```
50 LET x=LEN a$
```

Výraz musí být uzavřen v závorkách. LEN vrací počet typů v textu.

Příklad: následující řádek

```
120 INPUT a$:IF LEN a$>9 THEN GOTO 120
```

Prochází pouze testy obsahující více než 9 typů.

Formát: LEN string-const  
LEN string-var  
LEN (string-expr)

- LET

Umístění klávesy: L

Povel/příkaz

LET se používá jako přiřazení hodnoty k proměnné.

V Basicu Sinclair nemůže být LET vypuštěno v přiřazovacím příkaze.

Užití: LET běžně tvoří příkaz v programu, avšak může být použito jako přímý povel. Je následováno číselnou nebo textovou proměnnou, rovnítkem a poté hodnotou. Hodnota může být číselná nebo textová v závislosti na proměnné, která předchází LET, např.

```
60 LET x=x+1
```

```
80 LET a$="Correct"
```

Hodnota je přiřazena k proměnné.

Jednoduché proměnné jsou nedefinovány, pokud nemají přiřazené hodnoty pomocí klíčového slova LET, READ nebo INPUT. Pole proměnných je počítáno od 0 nebo nulového textu (viz DIM).

Formát: LET num-var=num-expr  
LET string-var=string-expr

- LINE

Umístění klávesy: EXTEND MODE, SYMBOL SHIFT 3

Viz INPUT, SAVE

- LIST

Umístění klávesy: K

Povel/příkaz

LIST vydává výpis z programu, který je v paměti.

**Užití:** LIST je běžně používán jako přímý povel, avšak může tvořit příkaz v programu. Pro výpis kompletního programu je používán sám. Po přímém povelu

LIST

je první strana výpisu vytištěna na obrazovku a následující strany se budou posouvat nahoru obrazovkou poté, co stlačíte jakoukoli klávesu kromě N, pomlčky, STOP nebo BREAK.

LIST může být rovněž následován číslem řádku ve formě numerické hodnoty, např.

LIST 100

Hodnota následující LIST je zaokrouhlena na nejbližší celé číslo podle nutnosti a výpis pokračuje na této řádce. Jestliže není řádka tohoto čísla, výpis pokračuje na další řádce.

**Formát:** LIST [int-num-expr]

- LLIST (řádková tiskárna LIST)

**Umístění klávesy:** EXTEND MODE V

**Povel/příkaz**

LLIST dává povel tiskárnám typu Sinclair vydávat a tisknout výpis programu obsažený v paměti.

**Užití:** LLIST se užívá stejným způsobem jako LIST (viz LIST) v průběhu tištění výpisu se nemění display na obrazovce.

**Formát:** LLIST [int-num-expr]

- LN (přirozený logaritmus)

**Umístění klávesy:** EXTEND MODE V

**Funkce**

LN udává přirozený logaritmus (logaritmus na bázi přirozeného čísla e) dané hodnoty. Je činný jako inverzní číslo k EXP.

Užití: LN je následováno číselnou hodnotou, např.

60 LET x=LN y

Výraz musí být uzavřen v závorkách. Hodnota následující za LN musí být vyšší než 0. LN poté uvádí přirozené logaritmy daných hodnot.

Formát: LN num-const

LN num-var

LN (num-expr)

- LOAD

Umístění klávesy: J

Povel/příkaz

LOAD vkládá celý program do paměti z pásku.

Užití: LOAD je běžně užíván jako přímý povel, avšak může být ve formě příkazu v programu v případě, že se nový program zakládá postupně. LOAD je následován jménem programu, který má hodnotu textové proměnné až do 10 písmen dlouhou, např.

LOAD "filename"

Při vykonávání tohoto příkazu je program, který je v dané chvíli v paměti a veškeré hodnoty jeho proměnných, vymazán. Spectrum poté hledá jmenovaný program a zakládá jej po nalezení do paměti.

Počítač rozlišuje mezi velkými a malými písmeny u názvů programů.

Je-li za příkazem LOAD **uvedeno " "**, např.

LOAD ""

poté Spectrum zakládá první úplný program, který nalezne. Klíčové slovo LOAD je užito jiným způsobem, je-li připojen microdrive, viz podrobnosti v příručce Microdrive a Interface 1.

Formát: LOAD string-expr

- LOAD CODE

Umístění klávesy: J, EXTEND MODE, I

Povel/příkaz

LOAD CODE se užívá pro zakládání části paměti s informací, která má být uložena do paměti. Informace je složena ze soustavy bytů, které jsou vysílány do soustavy adres v paměti. LOAD CODE může být použito pro zakládání displaye nebo zakládání informace pro grafiky definované uživatelem.

Užití: LOAD CODE se užívá jako přímý povel nebo může mít formu příkazu v programu. LOAD je následováno názvem uspořádání, které má textovou hodnotu, a poté CODE, např.

LOAD "data" CODE

Název uspořádání, který následuje po LOAD, je název informace, která má být založena, a je podřízen stejným omezením jako jména programu (viz LOAD).

LOAD CODE poté hledá dané informace a po jejich nalezení zobrazí **Byty: podle názvů**. Spectrum poté zakládá byty do paměti na adresy, ve kterých byly zachovány. Veškeré stávající informace jsou přepsány.

CODE může být rovněž následováno jednou nebo dvěma číselnými hodnotami oddělenými čárkou, např.

LOAD "picture" CODE 16384,6912

Hodnoty následující za CODE jsou zaokrouhleny s přesností na **nejvyšší** celé číslo a poté definují počáteční adresu (16384 výše), na které má být uložena jmenovaná informace a **číslo bytu** (6912), které **má být zasláno do bloků** (míst) začínajících touto adresou. Je-li číslo špatné, je dána zpráva o chybě při zakládání **pásky**. Jestli následuje za CODE jen jedna hodnota, definuje pouze počáteční adresu, od které mají být umístěny veškeré byty.



Výše uvedený příklad může být rovněž prováděn pomocí klíčových slov LOAD SCREEN§.

Detaily o ukládání bytů jsou uvedeny pod SAVE CODE.

Formát: LOAD string-expr CODE[int-num-expr][,int-num-expr]

- LOAD DATA

Umístění klávesy: J, EXTEND MODE, D

Příkaz/povel

LOAD DATA se užívá pro zakládání polí z pásku. Pole jsou nahrávána použitím SAVE DATA.

Užití: LOAD DATA se může používat ve formě příkazu v programu nebo jako přímý povel. LOAD je nejdříve následováno názvem souboru, který je textové povahy, následuje klíčové slovo DATA a písmeno nebo písmeno a § a konečně dvojicí prázdných závorek, např.

270 LOAD "numbers" DATA n()

300 LOAD "names" DATA n§()

Název souboru následující za LOAD je název, který je dán v poli na pásku a je podřizen stejným omezením jako název programu užitý s LOAD. Písmeno nebo písmeno § následující za klíčovým slovem DATA je název, který má být dán poli v programu, je-li založeno a použito.

Při vykonávání hledá Spectrum pojmenované pole na pásku. Po jejich nalezení vydá zprávu o číselném poli: nebo typovém poli: následovaném jménem a pole se zakládá. Jakékoli pole, které je v dané chvíli v paměti a má stejné písmeno názvu (n nebo n§) je vymazáno, a nové pole, které má toto jméno - písmeno a hodnoty uskladněné na pásku, je vytvořeno na jeho místě. S typovými poli je i jakákoli **proměnná textová** v dané chvíli uložená v paměti, která má stejný název, rovněž vymazána.

Formát: LOAD string-expr DATA letter[§]()

- LOAD SCREEN\$

Umístění klávesy: J, EXTEND MODE, SYMBOL SHIFT K

Příkaz/povel

LOAD SCREEN\$ umožňuje založení displaye na obrazovce přímo z pásku. Vysílá informace z pásku do části paměti, která ovládá a řídí display obrazovky v postupu, jak je tvořen obraz.

Užití: LOAD SCREEN\$ může být použit ve formě příkazu v programu nebo jako přímý povel. LOAD je následováno názvem souboru, který je textové povahy a poté SCREEN\$ např.

LOAD "picture" SCREEN\$

Název souboru, který následuje LOAD, je název, který je dán obrazovce informací na pásku a je podřízen stejným omezením jako programová jména použitá s LOAD. Spectrum poté hledá pojmenovanou informaci a po nalezení ji nejdříve plní do displayového uspořádání a poté atributní části paměti. Obraz se postupně skládá z barev, které má v dané chvíli tisk i podklad obrazovky. Poté jsou přidány **vlastnosti** (barvy atd.).

Podrobnosti o úschově informací z obrazovky - viz SAVE SCREEN\$.

Formát: LOAD string-expr SCREEN\$

- LPRINT (řádkový tisk)

Umístění klávesy: EXTEND MODE C

Příkaz/povel

LPRINT dává tiskárnám Sinclair příkaz PRINT danou pozici dat stejným způsobem, jak se PRINT objevuje na obrazovce.

Užití: LPRINT může tvořit příkaz v programu nebo přímý povel. Je následován pozicemi údajů, které mohou být odděleny středníky, čárkami nebo apostrofy,

např.

```
60 LPRINT "Number";x' "Name";n$, "Age";a
```

Při výstupu do tiskárny jsou pozice vytištěny ve stejném formátu jako se budou zobrazovat na obrazovce. Příkaz LPRINT nebo povel LPRINT může rovněž zahrnovat příkazy TAB, CHR\$, INVERSE a OVER a řídicí kódy se stejným účinkem, jako je povel PRINT. Lze rovněž začlenit příkaz AT, avšak tehdy je vypuštěno číslo řádku a pozice údajů, které jsou tištěny v daných sloupcích ve stejném řádku.

Formát: viz. str. 64 originálu sloupec uprostřed

#### - MERGE

Umístění klávesy: EXTEND MODE, SYMBOL SHIFT **T**

Povel/příkaz

MERGE umožňuje sloučení dvou programů.

Užití: MERGE může být použito pro tvoření příkazu v programu nebo jaké přímý povel. Je následováno názvem souboru ve formě textové hodnoty, např.

```
500 MERGE "prog2"
```

Název souboru následující za klíčovým slovem MERGE je název programu, který má být sloučen s programem, který je v daném okamžiku v paměti. Toto jméno je podřízeno stejným omezením jako programová jména použitá s LOAD. MERGE poté založí nový program bez předchozího vymazávání stávajícího programu. Nový program přepisuje veškeré řádky ve stávajícím programu, které mají stejná řádková čísla jako řádky nového programu, a proměnné se stejnými názvy jsou rovněž přepsány.

Formát: MERGE string-expr

- MOVE

Pro připojení microdrivu - povel viz příručka Microdrive a Interface 1.

- NEW

Umístění klávesy: A

Příkaz/povel

NEW čistí paměť Basicu (oblast **na** RAMTOP), odstraňuje veškeré programy, které jsou v této části paměti.

Užití: NEW je běžně užíváno jako přímý povel, avšak může mít tvar příkazu v programu. Je užíváno samostatně. Po výkonu je program a proměnné smazán. Paměť je vyčištěna až k RAMTOP, takže grafiky, definované uživatelem, které jsou uskladněny na RAMTOP, nejsou narušeny.

Formát: NEW

- NEXT

Umístění klávesy: N

Příkaz/povel

NEXT se vždy užívá spolu s FOR pro vytvoření smyčky FOR NEXT.

Užití: NEXT je běžně užíván pro tvoření příkazu v programu pro dokončení smyčky FOR NEXT. Je následován písmenem, které je řídicí proměnnou smyčky, např. 90 NEXT a

V jazyce Sinclair Basic musí být přidána řídicí proměnná. Podrobnější údaje o smyčkách FOR NEXT jsou uvedeny pod tímto heslem.

Formát: NEXT letter

- NOT

Umístění klávesy: SYMBOL SHIFT S

Logický operátor/funkce

NOT se užívá pro převrácení pravdivosti podmínky, takže záporná podmínka se stává pravdivou a naopak.

Užití: NOT je následováno podmínkou nebo číselnou hodnotou, např.

```
90 IF NOT x=y+z THEN PRINT "Wrong"
```

```
90 LET correct =x=y+z:IF NOT correct THEN PRINT "Wrong"
```

**Je-li** NOT následováno podmínkou ( $x=y+z$  výše), počítač nejdříve označí hodnotu 1 podmínky, je-li pravdivá, a 0, je-li nepravdivá. NOT poté účinkuje jako funkce, převrací vydanou hodnotu, takže může být zkoumána převrácená podmínka. Podmínka musí být uzavřena v závorkách, obsahuje-li povely AND nebo OR.

Jestliže je klíčové slovo NOT následováno číselnou hodnotou, vrací 0, je-li hodnota za ním nenulová, a 1, je-li hodnota následující za ním 0. Tzn. že ve výše uvedených příkladech vytiskne Spectrum "Wrong", je-li  $x < y+z$  nebo je-li hodnoty 0.

Formát: NOT cond

NOT num-expr

- OPEN#

Klíčové slovo pro užití s microdrivem - viz manuál Microdrive a interface 1.

- OR

Umístění klávesy: SYMBOL SHIFT U

Logický operátor/funkce

OR je činné jako logický operátor pro zkoumání pravdivosti kombinací podmínek. Jestliže je jedna nebo více podmínek pravdivých, poté jsou pravdivé všechny kombinace. OR rovněž funguje jako funkce, která převádí binární operace na dvě číselné hodnoty.

Užití: Jako logický operátor vede OR v příkazu dvě podmínky kde je zkoumá na pravdivost celé věty, např.

```
70 IF INKEY$="N" OR INKEY$ = "n" THEN STOP
```

Jestliže jakákoli nebo obě z podmínek jsou pravdivé, poté je celá kombinace pravdivá. Ve výše uvedeném řádku je jedna z podmínek (INKEY\$ = "N") a (INKEY\$ = "n") pravdivá tehdy, jakmile je stisknuto tlačítko n bez ohledu na to, je-li v činnosti CAPS SHIFT nebo CAPS LOCK. Celá kombinace je tedy pravdivá a program se zastaví.

OR jako funkce:

ZX Spectrum+ označí číselnou hodnotu 1 pravdivou podmínku a 0 nepravdivou podmínku. Rozliší jakoukoli nenulovou hodnotu jako pravdivou a 0 jako nepravdivou. OR může proto být následováno nebo předcházeno číselnou hodnotou, např.

```
40 LET x=y OR z
```

Proměnná x je poté označena jako hodnota 1, je-li z nenulová nebo pravdivostní podmínka, nebo hodnota y, je-li z nulová nebo jako nepravdivá podmínka.

To je užitečné v matematice, na následujícím příkladu je jízdné rozpůleno, je-li věk nižší než 14:

```
60 PRINT fare*(0.5 OR age<13)
```

Je-li věk menší než 14, podmínka věk je menší než 13 je nepravdivá, takže jízdné je násobeno 0,5. Jestliže věk je větší než 13, poté je jízdné násobeno 1.

Spectrum nevyhodnocuje kombinace číselných hodnot v souladu se standardními pravdivostními podmínkami.

Formát: cond OR cond|num-expr OR num-expr

- OUT

Umístění klávesy: EXTEND MODE, SYMBOL SHIFT O

Příkaz/povel

OUT vysílá byty na daný vstupní/výstupní port adresy v postupu, který řídí výstupní zařízení.

Užití: OUT může být užíváno ve formě příkazu v programu nebo jako přímý povel. Je následováno dvěma číselnými hodnotami, oddělenými čárkou, např.

40 OUT 254,3

Obě hodnoty jsou zaokrouhleny na nejbližší celé číslo, první hodnota (254) může být poté v rozsahu od 0 do 65535 a tvoří adresu portu. Druhá hodnota (3) může mít rozsah od 0 do 255 a je bytem, který je vysílán do této adresy.

Bity 0 až 2 bytu výstupů na adrese portu 254 seřizují barvu okraje; výše uvedený příklad proto převrací barvu na purpurovou. Bit 3 v této adrese řídí zdířku MIC a bit 4 reproduktor.

Adresa portu 251 řídí tiskárnu a 254, 247 a 239 jsou užívány s jinými periferiemi.

Formát: OUT int-num-expr,int-num-expr

- OVER

Umístění klávesy: EXTEND MODE, SYMBOL SHIFT N

Příkaz/povel

Klíčové slovo OVER je užíváno pro přetisknutí jednoho typu na druhý. Může být rovněž užíváno pro vytyčení bodů nebo vytažení čar nebo křivek v barvě papíru namísto v barvě inkoustu.

**Užití:** OVER je běžně užíváno pro tvoření příkazů programů. Je následováno číselnou hodnotou, např.

```
80 OVER 1
```

Hodnota následující za OVER je zaokrouhlena na nejbližší celé číslo a může být buď 0 nebo 1. OVER 0, které je předem nastaveno, působí, že veškeré typy nahradí předchozí typy na stejné pozici.

OVER 1 způsobí, že se zobrazí 2 typy na stejné charakterové poloze a ty jsou kombinovány.

OVER může být uloženo do příkazu PRINT nebo INPUT stejným způsobem, jako INK, takže narušuje pouze typy, které jsou zobrazeny v příkaze. Tento příkaz např. podtrhuje slovo

```
60 PRINT AT 11,15,"YES";  
   OVER 1; AT 11,15;"_____"
```

Typy jsou kombinovány tak, že barva papíru je dána tam, kde barva inkoustu překrývá.

OVER u grafiky s vysokým rozkladem:

OVER může být užito s příkazy PLOT, DRAW a CIRCLE. Bez příkazu OVER se křivky, přímky mohou překrývat navzájem, avšak musí mít stejnou barvu inkoustu, jinak je barva inkoustu měněná v jejich průsečících. Je-li užito OVER 1, linky nebo křivky se dostávají na svých překrývajících se částech na barvu podkladu nebo dosahují typů. Vytyčené body nebo vytažené čáry nebo křivky opět přesně ve stejné pozici s OVER 1 zmizí.

**Formát:** OVER int-num-expr



## - PAPER

Umístění klávesy: EXTEND MODE, SYMBOL SHIFT C

Příkaz/povel

Klíčové slovo OVER je užíváno pro volbu papíru nebo barvy pozadí použité pro obrazovku displaye. Může být buďto barvou pozadí po celé displayové ploše nebo barvou za jednotlivými typy, body nebo čarami, které se objevují na polohách jednotlivých typů.

Užití: PAPER může být použit pro tvoření příkazu v programu nebo jako přímý povel. Je následován číselnou hodnotou, např.

80 PAPER x

Hodnota následující za PAPER je zaokrouhlené nejbližší celé číslo a může dosáhnout rozmezí 0-9. Barvy papíru, které jsou tím dány, jsou stejné jako barvy dané pomocí klíčového slova INK. Barvy papíru mohou být rovněž globální nebo mohou tvořit místní zabarvení pomocí jejich vkládání do příkazu displaye v přesně stejném způsobu jako při vkládání barev tisku. Viz INK pro podrobnější popis.

Kdykoli jsou tištěny typy následující příkaz PAPER, ať již globální nebo místní, pozadí za celým místem typu se mění na zvolenou barvu. Toto je rovněž pravdivé u vytyčených bodů nebo vytažených čar nebo kruhů s uvedením příkazu PAPER, avšak **bez následného celkového** povelu nebo příkazu.

Pro vytvoření zabarveného pozadí po celé ploše displaye je nezbytné použít CLS po klíčovém slově PAPER. Vlastní display je vyčištěn na tuto barvu, která zůstává celkovou barvou pozadí.

Formát: PAPER int-num-expr[;]

- PAUSE

Umístění klávesy: M

Příkaz/povel

Klíčové slovo PAUSE může být použito pro zadržení programu na stanovený nebo neurčitý čas.

Užití: PAUSE je běžně užíváno pro tvoření příkazu v programu. Je následováno číselnou hodnotou, např.

130 PAUSE 100

Hodnota následující po PAUSE je zaokrouhlená na nejbližší celé číslo a může dosáhnout velikosti 0-65535. Definuje zdržení, které se projevuje u tohoto čísla řádku televizního obrazu, takže hodnota 50 vydává pauzu 1 sekundy ve Velké Británii a Evropě, kde je síťová frekvence 50 Hz. Veškeré prodlevy mohou být zkráceny stisknutím jakéhokoli klíče a PAUSE 0 dává neomezenou prodlevu, která trvá, dokud není klávesa stlačena.

Formát: PAUSE int-num-expr

- PEEK

Umístění klávesy: EXTEND MODE 0

Funkce

Klíčové slovo PEEK udává hodnotu bytu uloženého na hlavní adrese v paměti.

Užití: PEEK je následováno číselnou hodnotou, např.

80 LET x=PEEK(256\*Y)

Výrazy musí být uzavřeny v závorkách. Hodnota následující za klíčovým slovem PEEK je zaokrouhlená na nejbližší celé číslo, je-li nezbytné, a může být v rozsahu 0-65535, aby udala adresy v paměti.

PEEK poté vrací hodnotu bytu (číslo od 0 do 255) a udanou adresu.

Příklad: Počet uspořádání televizního displaye, který se objevil při posledním zapnutí Spectra, byl uložen

na adresy 23672-23674. Jelikož jsou tyto struktury vydávány v **pravidelném poměru**, **špičkování** jejich umístění udává metodu měření času. Následující řádek zobrazí čas v sekundách, **neboť Spectrum byl** naposledy zapnut (**bez ohledu na jakoukoli** dobu strávenou vydáváním zvuku a činností periférií, jako jsou kazetový přehrávač nebo tiskárna).

```
10 PRINT (PEEK 23672+256*PEEK 23673+ 65536*PEEK  
23674)/50
```

Poznámka: Je-li síťová frekvence 60 a ne 50 Hz (standart UK), změňte 50 na 60.

Formát: PEEK int-num-const  
PEEK int-num-var  
PEEK (int-num-expr)

- PI

Umístění klávesy: EXTEND MODE M

Funkce

PI udává hodnotu Ludolfova čísla (pí) ve výpočtech. Pí je poměr obvodu kruhu/průměr.

Užití: PI nevyžaduje hodnoty nebo proměnné při použití v příkaze nebo povelu, např.

```
DRAW 255,0,-PI
```

PI vrací hodnotu 3,1415927, takže výše uvedený povel vytahuje velký půlkruh na obrazovce.

Formát: PI

- PLOT

Umístění klávesy: Q

Příkaz/povel

PLOT je užíváno u grafik s vysokým rozkladem, vytyčuje body nebo tečky barev na dílčích polohách na obrazovce.

**Užití:** PLOT je užíváno ve formě příkazu v programu nebo jako povel. Normálně je následováno dvěma číselnými hodnotami oddělenými čárkou, např.

50 PLOT 128,87

Obě hodnoty následující klíčové slovo PLOT jsou zaokrouhleny na celá čísla, je-li potřeba. První hodnota může sahat od 0 do 255 a definuje horizontální souřadnice na obrazovce. Druhá hodnota může být v rozmezí od 0 do 175 a stanoví svislé souřadnice. Bod je tím normálně vytyčen v **správné** barvě tisku na stanovené poloze - výše uvedený příklad - ve středu obrazovky.

Vlivy příkazů nebo povelů pro barvu při PLOT:

Po OVER 1 stávající tečka na stejné poloze je změněna na barvu podkladu. Použitím INVERSE 1 je tato tečka vytyčena v **přesné** barvě papíru. Po BRIGHT 1 nebo FLASH 1 bude celá poloha typu u obrazovky s nízkým rozkladem v místě vytyčeného bodu světlá nebo blikající. Tato čtyři klíčová slova a INK mohou rovněž být přidána uvnitř příkazu PLOT stejným způsobem jako s PRINT, např.

160 PLOT INK 2;x,y

Jejich efekt je stejný, avšak je místní a omezený na body vytyčené příkazem. Jestliže klíčové slovo PAPER doplněno do příkazu PLOT, barva papíru celého typu a jeho polohy okolo bodů se změní na danou barvu.

PLOT rovněž stanoví počáteční polohu dalšího příkazu DRAW.

**Formát:** PLOT[statement;] int-num-expr,int-num-expr

## - POINT

Umístění klávesy: EXTEND MODE, SYMBOL SHIFT 8

Funkce

POINT je používání, která barva na hlavní poloze obrazovky s vysokým rozkladem je buď barvou tisku nebo barvou podkladu. POINT nekontroluje skutečnou barvu.

Užití: POINT je následován dvěma číselnými hodnotami, oddělenými čárkou a uzavřenými v závorkách, např.  
240 IF POINT(x,y)=1 THEN GOSUB 600

Dvě hodnoty následující klíčové slovo POINT jsou zaokrouhleny na celá čísla podle nezbytnosti. První hodnota může mít rozsah od 0 do 255 a stanoví vodorovnou souřadnici bodu na obrazovce. Druhá hodnota může mít rozsah od 0 do 175 a stanoví svislou souřadnici. POINT poté vrátí 1, jestliže bod na stanovené poloze je barva tisku, nebo 0, je-li barvy papíru.

Formát: POINT(int-num-expr,int-num-expr)

## - POKE

Umístění klávesy: O

Příkaz/povel

POKE je užíváno pro změnu hodnoty bytu na hlavní adrese v paměti. Hodnoty jsou běžně zasílány na paměťová místa v pořadí, jak mají udávat činnosti nezadané klíčovými slovy Basic.

Užití: POKE je používáno pro tvoření příkazů v programu nebo jako povel. Je následováno dvěma číselnými hodnotami oddělenými čárkou, např.

POKE 23609,255

Dvě hodnoty následující POKE jsou zaokrouhleny na nejbližší celé číslo podle nezbytnosti. První hodnota může být v rozsahu od 16384 do 65535 a udává adresu v RAM. Druhá hodnota může být v rozsahu

od 0 do 255 a tvoří byte, který má být zapsán do stanovené adresy.

Ve výše uvedeném příkladě je 255 vloženo do adresy 23609, která řídí zvuk vyráběný při stisknutí klávesy. Hodnota 255 dává dlouhý pípavý zvuk namísto krátkého, jiné hodnoty vydávají kratší pípnutí.

Formát: POKE int-num-expr,int-num-expr

- PRINT

Umístění klávesy: P

Příkaz/povel

PRINT zobrazuje údaje na obrazovce. Data mohou být jakýkoli jednoduchý typ nebo sousled typů. Příkaz PRINT může zahrnovat další klíčová slova pro definici polohy a barvu údaje.

Užití: PRINT může být použito buď samotné nebo může být následováno údajem. Tento údaj může být ve formě jakéhokoli číselného nebo textového vyjádření nebo v jejich směsi. Při použití PRINT s daty, dvě nebo více oddělených dat musí být vždy odděleno středníkem, čárkou nebo apostrofem.

Určitá další klíčová slova mohou být vkládána do jakéhokoli pořadí mezi PRINT a data s tím, že každý příkaz utvářený klíčovým slovem končí středníkem. Tato klíčová slova jsou: CHR\$, TAB, AT, **INK**, PAPER, FLASH, BRIGHT, INVERSE a **OVER**.

Výpis textu pomocí PRINT:

PRINT sám o sobě nebo následován nulovým textem ("") zobrazí slepou řádku a pohybuje cursorem na začátek další řádky.

PRINT následovaný textovou konstantou (jakékoli typy uvnitř "") označuje typy jak se objevují mezi uvozovkami), např.

PRINT "3/542/76/21"

zobrazí na obrazovce 3/542/76/21.



PRINT a další klíčová slova:

PRINT může být následováno klíčovým slovem TAB, číselnou hodnotou, středníkem a poté pozicí údaje, např.

```
60 PRINT TAB x;ař
```

Hodnota následující za TAB (x) je zaokrouhlena na nejbližší celé číslo podle nutnosti a poté je dělena 32 a zbytek převrácen, aby dal hodnotu mezi 0 a 31. Pozice údaje je poté zobrazena v této sloupcové poloze ve stejné nebo další řádce. PRINT může být následován i AT a poté dvěma čísly oddělenými čárkou, středníkem a pozicí dat, např.

```
50 PRINT AT I,c; "Data"
```

První hodnota (I) může mít rozsah 0 až 21 a stanoví počet řádek nebo řádků, ve kterých budou údaje zobrazeny.

Druhá hodnota (c) může mít rozsah 0 až 31 a stanoví počet sloupců, ve kterých bude první typ nebo údaj zobrazen.

Necelá čísla jsou přijímána a zaokrouhlena na nejbližší celé číslo. Povel PRINT AT 11,16; "x" zobrazí hvězdičku ve středu obrazovky.

PRINT může být rovněž následován jedním nebo více funkcemi CHRř - viz funkce CHRř.

PRINT a klíčová slova barev:

Display vydávaný pomocí PRINT je ovlivněn příkazy o barvách nebo povely o barvách dodávanými klíčovými slovy INK, PAPER, FLASH, BRIGHT, INVERSE a OVER, která jsou právě činná.

PRINT může být rovněž následováno jedním nebo více z těchto šesti příkazů, z nichž každý je následován středníkem před pozicí dat, např.

```
50 PRINT AT 11,16; INK 2; FLASH 1; "x"
```

Položka údaj je potom zobrazena s vlastnostmi danými klíčovým slovem barvy. Tyto vlastnosti jsou místní a vztahují se pouze na zobrazenou pozici. Následující provedení příkazu PRINT se vrací do svého původního stavu nebo na předešlou stanovenou celkovou hodnotu. PRINT rovněž uposlechne příkazu kódů řídicích místní barvu, které jsou vloženy do údajů (viz str. 33)



Formát: viz str. 68 originálu prostřední sloupec dole

- RANDOMIZE

Umístění klávesy: T

Příkaz/povel

RANDOMIZE je na klávesnici pod zkratkou RAND, používá se spolu s RND pro vyrábění sekvencí čísel, která jsou buď libovolná nebo předvídatelná.

Užití: RANDOMIZE je užíváno buď ve formě příkazu v programu nebo jako povel. Je volitelně následováno číselnou hodnotou, např.

```
RANDOMIZE 1
```

```
10 RANDOMIZE
```

Hodnota následující po klíčovém slově RANDOMIZE je zaokrouhlena na nejbližší celé číslo a může být v rozmezí od 0 do 65535. Hodnota vyšší než 0 se řídí soustavou proměnných SEED do této hodnoty následující po RND, které vždy generuje stejnou řadu čísel (viz str. 48 o systémových proměnných).

Skutečný sled závisí na hodnotě klíčového slova RANDOMIZE.

Jestliže je klíčové slovo RANDOMIZE následováno 0 nebo žádnou hodnotou, poté je SEED dáno jinou systémovou proměnnou nazývanou FRAMES, která počítá rámy zobrazené na televizi od chvíle, kdy bylo Spectrum zapnuto. Jak se SEED mění 50 nebo 60 krát za sekundu, řada čísel generovaných RND následující za RANDOMIZE nebo RANDOMIZE 0 je velmi náhodná.

Jestliže nepoužíváme RANDOMIZE, RND generuje stejné pořadí čísel od zapnutí a použití tlačítka RESET nebo NEW.

Formát: RANDOMIZE[int-num-expr]

- READ

Umístění klávesy: EXTEND MODE A

Příkaz/povel

READ je užíváno spolu s DATA pro přiřazení hodnoty proměnných za použití hodnot v příkaze DATA.

Užití: READ je běžně užíváno pro tvoření příkazu v programu. Je následováno jednou nebo více číselnými proměnnými nebo textovými proměnnými, každé oddělené čárkou, např.

20 READ a\$,x

Když je poprvé READ prováděno, bere stejné číslo hodnoty, jako je proměnná na začátku prvního výpisu DATA, a přiřadí hodnotu proměnné v daném pořadí. Když je READ prováděno dále, další sada hodnot DATA je přiřazena dalším proměnným pojmenovaným v příkaze READ atd.

Další podrobnosti - viz DATA.

Formát: READ num-var[,num-var][,string-var]

READ string-var[,num-var][,string-var]

- REM

Umístění klávesy: E

Příkaz

REM je užíváno pro založení poznámek nebo připomínek do programu. Tyto mohou být: titul a autor programu, vysvětlení řádků v programu, účel proměnných atd. Poznámky nehrají žádnou roli v chodu programu a mohou být viděny pouze ve výpisu.

Užití: REM tvoří buď vlastní řádek v programu nebo poslední příkaz v řádku. Je následován jakoukoli poznámkou, která může být podle požadavku zakódována, např.

80 INPUT n\$:REM n\$ is name

Když počítač zaznamená REM, neuvažuje žádné údaje až do konce řádku.

Formát: REM any characters

## - RESTORE

Umístění klávesy: EXTEND MODE S

Příkaz/povel

RESTORE je užíváno spolu s READ a DATA. **Provádí příkaz pro vybírání hodnot READ z dílčích** příkazů DATA namísto prvního nebo dalšího příkazu DATA v programu.

Užití: RESTORE tvoří příkaz v programu. Je možné jej vyplnit číselnou hodnotou, např.

160 RESTORE 800

Hodnota udaná za RESTORE je zaokrouhlena na nejbližší celé číslo a musí být číslem řádku programu obsahujícího příkaz DATA. Po klíčovém slově RESTORE přiřadí další příkaz READ hodnoty obsažené v tomto příkaze DATA. Jestliže očíslované řádky neexistují, nebo neobsahují příkaz DATA, poté přechází READ na další příkaz DATA za tímto řádkem. Jestliže je RESTORE následováno nulou nebo nulovou hodnotou, další příkaz READ přechází na první příkaz DATA v programu.

Formát: RESTORE[int-num-expr]

## - RETURN

Umístění klávesy: Y

Příkaz/povel

RETURN je užíváno pro **zapojení** podprogramu a vrácení počítače do hlavního programu nebo předešlého podprogramu.

Užití: RETURN je užíváno jako forma příkazu v programu.

Je užíváno samostatně na konci podprogramu, např.

1080 RETURN

Po výkonu se program **větví** na příkaz následující za posledním příkazem GOSUB, který byl proveden.

Viz heslo GOSUB, kde je podrobnější vysvětlení.

Formát: RETURN

- RND (náhodná čísla)

Umístění klávesy: EXTEND MODE T

Funkce

RND se používá na generování náhodných čísel.

Užití: RND je používáno samostatně jako příkaz nebo povel, např.

60 LET x=RND

RND pak vrací náhodná čísla menší než 1 a větší než 0. Když je počítač zapnut nebo znova nastaven nebo je použito klíčového slova NEW, čísla se postupně vrací pomocí RND ve stejném pořadí. Pořadí je generováno podle mocniny  $75(75,75 \times 75,75 \times 75 \times 75 \text{ atd.})$  dělic každou mocninu 65537 a užívajíc pouze zbytek, poté odčítá 1 od zbytku a dělí tento výsledek 65536. Je-li vyžadována více náhodná sekvence nebo jiná pevná sekvence, užíjte před RND RANDOMIZE.

Celá náhodná čísla:

Mnoho příkazů a funkcí v počítači Spectrum, jako INK, CHR\$, čísla blížká a limitně se blížící celým číslem a RND mohou být použita přímo. INK RND\*x např. vydává náhodnou barvu inkoustu. Další požadavky celých čísel a všech úplných čísel od 1 do x je dáno  $\text{INT}(\text{RND} \times x) + 1$ . Pro generování náhodných celých čísel od 0 do x užíjte  $\text{INT}(\text{RND} \times x + 0.5)$ .

Formát: RND

- RUN

Umístění klávesy: R

Příkaz/povel

RUN startuje program od prvního řádku.

Užití: RUN může být užíván jako přímý povel nebo může tvořit formu příkazu v programu. Je příležitostně následováno číselnou hodnotou, např.

RUN 50

Není-li žádná hodnota za RUN, program běží od prvního řádku. Je-li začleněna hodnota, je zaokrouhlena na nejbližší celé číslo a program poté začíná od tohoto řádku. Jestliže řádek neexistuje, program začíná od dalšího řádku programu. RUN provádí CLEAR před probíháním programu, takže proměnné hodnoty jsou vymazány. Abyste tomu zabránili, užíjte klíčové slovo GOTO následované číslem řádku.

Jestliže byl program zachován použitím klíčového slova LINE, pak běží automaticky při zakládání a RUN není vyžadováno.

Formát: RUN[int-num-expr]

- SAVE

Umístění klávesy: S

Příkaz/povel

SAVE zasílá program na kazetový přehrávač podle pořadí, aby byl zachován na pásce.

Užití: SAVE je používáno jako přímý povel, avšak může mít formu příkazu v programu. Je následováno názvem souboru, který je textové hodnoty, např.

SAVE "filename"

Název souboru může obsahovat až 10 typů. Při provádění je zobrazena zpráva

Start tape, then press any key

Při stisknutí jakékoli klávesy je program vyslán do kazetového přehrávače a na závěr je vyslána zpráva

O OK,0,1

SAVE je užíváno jinak v případě připojení microdrivu.

Detaily - viz Microdrive a Interface 1 - příslušný manuál.

Automatické spouštění:

Jestliže je program koncipován tak, že se má spustit po nahrání, musí být SAVE použito spolu s LINE. Programové jméno

je následováno klíčovým slovem LINE a číselnou hodnotou, např.

SAVE "filename" LINE 1

Hodnota následující za LINE je zaokrouhlena na nejbližší celé číslo podle nutnosti a může být buď 1 anebo číslo řádku v programu. Poté je program uložen na pásek stejným způsobem jako při SAVE. Při nahrávání se program automaticky zpustí od řádku, který má definované číslo nebo od následujícího řádku v programu jestli řádek s definovaným číslem neexistuje. V praxi použití LINE 1 způsobí, že se celý program zpustí automaticky.

Formát: SAVE string-expr [LINE int-num-expr]

#### - SAVE CODE

Umístění klávesy: S, EXTEND MODE, I

Příkaz/povel

SAVE CODE zasílá část paměti s informacemi na kazetový přehrávač k uložení na magnetofonový pásek. Informace mohou být poté načteny zpět do paměti použitím LOAD CODE.

Užití: SAVE může být použito jako přímý povel, nebo může tvořit příkaz v programu. SAVE je následováno názvem souboru, které je textové hodnoty, následováno CODE, které je postupně následováno dvěma číselnými hodnotami oddělenými čárkou, např.

SAVE "picture" CODE 16384,6912

Název souboru který následuje SAVE může mít až 10 písmen. Hodnoty následující za CODE jsou zaokrouhlené na nejbližší celé číslo. je-li to nutné. První udává počáteční adresu (16384 výše) informace v paměti a druhá hodnota (6912) udává počet bytů, které mají být uloženy. Poté je informace odeslána do na pásek stejným způsobem jako program příkazem SAVE.

Informace uložená výše uvedeným příkazem je obrazovka displaye.

Formát: SAVE string-expr CODE int-num-expr,int-num-expr

### - SAVE DATA

Umístění klávesy: S, EXTEND MODE, D

Povel/příkaz

SAVE DATA uloží pole na pásku. Pole může být založeno použitím LOAD DATA.

Užití: SAVE DATA může tvořit příkaz v programu, nebo může být použito jako přímý povel. SAVE je následováno názvem souboru, které je následováno klíčovým slovem DATA, a písmenem nebo písmenem a § a nakonec dvojicí prázdných závorek, např.

```
450 SAVE "numbers" DATA n()
```

```
750 SAVE "names" DATA n§()
```

Název souboru pole může obsahovat až 10 písmen. Písmeno nebo písmeno §, které následuje za klíčovým slovem DATA je název pole v programu, které má být uloženo na pásku. Poté je pole odesláno na pásku stejným způsobem jako program příkazem SAVE.

Formát: SAVE string-expr DATA letter [§]()

### - SAVE SCREEN§

Umístění klávesy: S, EXTEND MODE, SYMBOL SHIFT K

Příkaz/povel

SAVE SCREEN§ ukládá obrazovku displaye na pásku. Ta může být později opět založena do počítače použitím LOAD SCREEN§.

Užití: SAVE SCREEN§ může být použito jako přímý povel, nebo může tvořit příkaz v programu. SAVE je následováno názvem souboru, které je textové hodnoty, následováno SCREEN§, např.  
SAVE "picture" SCREEN§

Název souboru může mít až deset písmen. Poté je obrazovka odeslána na pásku stejným způsobem jako program příkazem SAVE.

Formát: SAVE string-expr SCREEN§

### - SCREEN\$

Umístění klávesy: EXTEND MODE, SYMBOL SHIFT K

Funkce

SCREEN\$ stanoví, který typ se objeví v hlavní poloze na obrazovce.

Užití: SCREEN\$ je následováno dvěma číselnými hodnotami, které jsou odděleny čárkou a uzavřeny v závorkách, např.

```
160 IF SCREEN$(I,c) = "X" THEN PRINT "CRASH"
```

Hodnoty následující SCREEN\$ jsou zaokrouhleny na nejbližší celé číslo. První hodnota (I) může mít potom rozsah 0-21 a udává číslo řádku polohy na obrazovce. Druhá hodnota (c) může mít rozsah od 0-31 a uvádí číslo sloupce polohy. SCREEN\$ poté vrací typy zobrazené na této poloze jako textovou konstantu (v uvozovkách - viz "X" výše). Není-li žádný typ na dané poloze, SCREEN\$ se vrací na 0 tisk ("").

SCREEN\$ může být rovněž užito se SAVE a LOAD pro uchování displeje na obrazovce na pásek a opětovné založení tohoto obrazu z pásku. Viz SAVE SCREEN\$ a LOAD SCREEN\$, kde jsou uvedeny podrobnosti.

Formát: SCREEN\$ (int-num-expr,int-num-expr)

### - SGN (znaménko)

Umístění klávesy: EXTEND MODE F

Funkce

SGN značí, které číslo je kladné, záporné nebo nulové.

Užití: SGN je následováno číselnou hodnotou, např.

```
50 LET x=SGN y
```

Vyjádření musí být uzavřeno v závorkách. SGN poté vrací 1, je-li hodnota argumentu (y) kladná, -1, je-li záporná, a 0, je-li nulová.

Formát: SGN num-const

SGN num-var

SGN (num-expr)



- SIN (sinus)

Umístění klávesy: EXTEND MODE Q

Funkce

SIN udává sinus úhlu.

Užití: SIN je následováno číselnou hodnotou, např.

80 LET x=SIN y

Vyjádření musí být uzavřeno do závorek. Hodnota následující SIN je úhel v radiánech a SIN vrací sinus úhlu. Stupně mohou být převedeny na radiány pomocí vynásobení  $\text{PI}/180$ .

SIN vrací kladné hodnoty úhlů mezi 0-180 stupni, záporné hodnoty pro úhly mezi 180-360 stupni.

Příklad: Povel

PRINT SIN (30\*PI/180)

Zobrazí 0.5 - sinus 30 stupňů.

Formát: SIN num-const

SIN num-var

SIN (num-expr)

- SQR (druhá odmocnina)

Umístění klávesy: EXTEND MODE H

Funkce

SQR dává druhou odmocninu čísla.

Užití: SQR je následováno číselnou hodnotou, např.

70 LET x=SQR y

Vyjádření musí být uzavřeno v závorkách. Hodnota následující za SQR (výše uvedený y) musí být vyšší než 0 a SQR vrací druhou odmocninu.

Formát: SQR num-const

SQR num-var

SQR (num-expr)

- STEP

Umístění klávesy: SYMBOL SHIFT D

Viz FOR

- STOP

Umístění klávesy: SYMBOL SHIFT A

Příkaz/povel

STOP zastavuje program na daném bodě. Může být nezbytné použít příkazu STOP pro ukončení hlavní části programu v posloupnosti pro ohraničení podprogramů na samostatný oddíl. STOP je rovněž platné v odlaďování programů.

Užití: STOP je běžně používán pro tvoření příkazu v programu. Je užíván sám o sobě, např.

650 STOP

Při výkonu program zastaví na zprávě

9 STOP statement

objeví se s řádkem a **číslem** příkazu, na kterém je program zadržen.

V této fázi může být rovněž odlaďován program, jako např. zobrazování a výměna hodnot proměnných.

Po stisknutí CONTINUE program dále pokračuje na dalším příkazu s novými hodnotami.

Formát: STOP

- STR\$

Umístění klávesy: EXTEND MODE Y

Funkce

STR\$ převádí číslo do textu.

Užití: STR\$ je následováno číselnou hodnotou, např.

90 LET a\$=STR\$ x

Vyjádření musí být uzavřeno v závorkách. STR\$ poté vrací hodnotu argumentu (x výše) jako textovou konstantu. Jestliže x bylo označeno hodnotou 65, výše uvedený příkaz označený a\$ má hodnotu "65".

Formát: STR $\mathcal{S}$  num-const  
STR $\mathcal{S}$  num-var  
STR $\mathcal{S}$  (num-expr)

- TAB

Umístění klávesy: EXTEND MODE P  
Viz LPRINT, PRINT

- TAN (tangens)

Umístění klávesy: EXTEND MODE E  
Funkce

TAN udává tangentu úhlu.

Užití: TAN je následováno číselnou hodnotou, např.

130 LET x=TAN y

Vyjádření musí být uzavřeno v závorkách. Hodnota následující za TAN je úhel v radiánech a TAN udává tangentu úhlu. Stupně mohou být převáděny do radiánů násobením číslem  $\text{PI}/180$ .

TAN udává kladnou hodnotu pro úhly 0-90 stupňů a 180-270 stupňů. Pro úhly mezi 90-180 stupni a mezi 230-360 stupni udává TAN zápornou hodnotu.

Formát: TAN num-const  
TAN num-var  
TAN (num-expr)

- THEN

Umístění klávesy: SYMBOL SHIFT G  
Viz IF

- TO

Umístění klávesy: SYMBOL SHIFT F

Funkce

TO má dvě odlišné využití v jazyce Sinclair Basic. Je užíváno spolu s FOR pro vytvoření smyčky FOR NEXT (viz FOR) a je rovněž užíváno při rozdělování řetězcových proměnných do menších.

Užití TO pro rozdělování řetězcových proměnných:

Klíčové slovo TO je použito pro definici prvního a posledního typu podtextu uvnitř hlavního textu. TO je předcházeno hodnotou textu, otevřením závorky, poté volitelnou číselnou hodnotou. Je následováno další volitelnou číselnou hodnotou a poté uzavřeno závorkou, např.

```
80 PRINT a$ (4 TO 7)
```

Výraz textu musí být rovněž uzavřen v závorce. Hodnota textu (a\$ výše) je text, který má být rozdělen. Dvě číselné hodnoty (4 a 7) definují polohu prvního a posledního typu podtextu uvnitř textu. Klíčové slovo TO poté udává podtext (typy 4-7 daného textu).

První číselná hodnota má základní hodnotu 1 a druhá má základní hodnotu rovnou pozici posledního typu v textu.

První hodnota může proto být vypuštěna, začíná-li podtext prvním typem v textu, a druhá hodnota může být vypuštěna, končí-li podtext posledním typem textu.

```
Formát: string-const([num-expr]TO[num-expr])  
         string-var([num-expr] TO[num-expr])  
         (string-expr)([num-expr]TO[num-expr])
```

- USR (uživatelský podprogram)

Umístění klávesy: EXTEND MODE L

Funkce

USR se používá pro vyvolání podprogramu ve strojovém kódu, který byl umístěn do paměti na zvláštní adresu. Je rovněž užíváno pro umístění údajů pro uživatelsky definované grafiky na vyhrazených místech v horní části paměti.

USR a strojový kód:

Pro užití strojového kódu je USR následováno číselnou hodnotou, např.

```
80 PRINT USR 65000  
100 RANDOMIZE USR 65000
```

Vyjádření musí být uzavřeno v závorkách. Hodnota následující za `USR` je zaokrouhlena na nejbližší celé číslo a je počáteční adresou v paměti, na které byl umístěn podprogram ve strojovém kódu.

**Další** příkaz obsahující `USR` poté vyvolává podprogram z této adresy a `USR` vrací hodnotu obsahu na dvojici `bc` registrů. `RANDOMIZE USR` nebo `RESTORE USR`, např. nechá probíhat pouze podprogram, zatímco `PRINT USR` navíc zobrazí hodnoty registru.

`USR` a grafiky definované uživatelem:

Pro vytvoření grafik definovaných uživatelem je použito `USR` spolu s `POKE`. Je následováno textovou konstantou nebo proměnnou, která udává adresu pro příkaz `POKE`, např. `50 POKE USR "a", 255`

Textová hodnota následující `USR` může být jednoduché písmeno v rozsahu `A-U` nebo `a-u`. Velká písmena nejsou odlišována od malých.

`USR` poté udává počáteční adresu jedné z 21 částí paměti vyhrazené pro grafiky definované uživatelem. Každá část obsahuje 8 adres, na jejichž 8 bytů jsou dávány typy pro vytvoření daných grafik. Byty mohou být udány v desetinné formě nebo binární formě (viz `BIN`).

Formát: `USR int-num-const`  
`USR int-num-var`  
`USR (int-num-expr)`  
`USR string-const`  
`USR string-var`

- `VAL` (hodnota)

Umístění klávesy: `EXTEND MODE J`

Funkce

`VAL` mění text s číselnou hodnotou na číslo.

Užití: VAL je následováno textovou konstantou nebo proměnnou, např.

```
70 LET x=VAL aš
```

Hodnota textové konstanty nebo proměnné je rozdělena uvozovkami a musí být číselné hodnoty. VAL vyhodnocuje tento údaj, udává jej jako číselnou konstantu.

Příklady: Jestliže aš má hodnotu "435", poté výše uvedený příkaz označuje hodnotu 435-x. VAL rovněž může vyhodnotit vyjádření, např.

```
10 INPUT aš,x
```

```
20 PRINT VAL aš
```

Textová hodnota je označena jako aš - musí být vyjádřena použitím x, např. "xxx". Poté je x přiřazena číselná hodnota, např. 5. VAL odstraní uvozovky textu, čímž získá xxx a vyhodnotí ji užitím hodnoty přiřazené x, zobrazí výsledek 25.

Formát: VAL string-const

VAL string-var

- VALš

Umístění klávesy: EXTEND MODE, SYMBOL SHIFT J

Funkce

VALš vyhodnocuje text jako vyjádřený text.

Užití: VALš je následováno textovou proměnnou, např.

```
130 PRINT VALš aš
```

Hodnota textové proměnné je oddělena od uvozovek a musí být vyjádřením textu. VALš vyhodnocuje vyjádření a udává hodnotu jako textovou konstantu.

Příklady: Vyzkoušejte následující program:

```
10 INPUT aš,xš
```

```
20 PRINT VALš aš
```

Textové hodnoty jsou takové, že aš musí být vyjádřením použití xš, např. "xš + xš".

Textová hodnota je poté označena jako x\$, např. "DO". VAL\$ oddělí uvozovky od hodnoty a\$, čímž získá x\$ + x\$ a vyhodnotí je použitím hodnoty **přidané x\$,** zobrazí výsledek DODO.

Formát: VAL\$ string-var

#### - VERIFY

Umístění klávesy: EXTEND MODE, SYMBOL SHIFT R

Příkaz/povel

Klíčové slovo VERIFY kontroluje, že program byl přesně uložen **do paměti** po SAVE.

Užití: VERIFY je běžně užíváno jako přímý povel přesně stejným způsobem jako LOAD a je následováno názvem programu, např.

VERIFY "filename" (název souboru)

Když byl pásek spuštěn, název každého programu, který je nalezen, je zobrazen a **všechny programy** na pásku stejného jména **jsou porovnávány** s programem v paměti. Jestliže jsou **nalezeny** dva stejné, objeví se zpráva OK,0:1

Klíčové slovo VERIFY je použito při připojení microdrive jiným způsobem. Viz příručka pro Microdrive a Interface 1.

VERIFY CODE a VERIFY DATA:

VERIFY CODE může být použito stejným způsobem jako LOAD CODE pro ověření, že část paměťové informace byla uložena na pásku.

VERIFY DATA pracuje stejným způsobem jako LOAD DATA pro kontrolu, že pole byla uložena na pásku.

VIZ LOAD CODE a LOAD DATA - další podrobnosti.

Formát: VERIFY string-expr

VERIFY string-expr CODE[int-num-expr][,int-num-expr]  
VERIFY string-expr DATA letter[\$]()

- VERIFY CODE

Umístění klávesy: EXTEND MODE, SYMBOL SHIFT R, EXTEND MODE I  
Viz VERIFY

- VERIFY DATA

Umístění klávesy: EXTEND MODE, SYMBOL SHIFT R, EXTEND MODE D  
Viz VERIFY



## Odkazy na obrazovce

Když počítač Spectrum zastaví provádění **BASICu**, objeví se na spodní straně obrazovky zpráva. To značí, že byl povel nebo program ukončen nebo že byla objevena chyba. Každá zpráva sestává z kódového čísla nebo písmena, za kterým je vlastní zpráva, a poté číslo řádku a příkazu, na kterém se počítač zastavil. Povel je ukázán jako řádek 0 a uvnitř řádku, příkaz 1 je na začátku řádku, příkaz 2 je za **prvním sloupcem** nebo THEN atd. Klíčové slovo CONTINUE normálně způsobí, že program pokračuje na příkazu udaném ve zprávě.

0 OK

Úspěšné ukončení nebo přístup ke skoku na řádkové číslo vyšší než jaké je udáno v programu. Klíčové slovo CONTINUE nezohledňuje tuto zprávu a pokračuje v příkazu stanoveném v předchozí zprávě.

1 NEXT without FOR (bez FOR)

NEXT bylo začleněno bez odpovídajícího FOR a proměnná existuje se stejným jménem jako řídící proměnná.

2 Variable not found

Nebyla nalezena proměnná - jednoduchá proměnná, která byla použita bez přiřazené hodnoty nebo založení hodnoty z pás-ku, nebo řídící proměnná, která byla použita s NEXT bez prvního nastavení na příkaz FOR, nebo popsaná proměnná, která byla použita před vymezením pole pomocí DIM nebo založením pole z pás-ku.

3 Subscript wrong

Popis je mimo vymezení pole - špatný popis.

4 Out of memory

Není dostatek prostoru v pamětech na dokončení příkazu nebo povelu.

5 Out of screen

INPUT generován více než na 23 řádek v spodní části obrazovky nebo čísla řádků od 22 nebo více byla použita u klíčových slov PRINT AT.

6 Number too big (číslo příliš vysoké)

Počítač vyzkoušel vyrobit číslo vyšší než přibližně  $10^{38}$ .

7 RETURN without GOSUB

Číslo příkazu RETURN je vyšší než číslo podprogramu příkazu GOSUB.

8 End of file (konec uspořádání)

Zpráva z microdrivu.

9 STOP statement

STOP bylo použito pro zastavení programu. Klíčové slovo CONTINUE je možno použít pro pokračování na dalším příkaze.

A Invalid argument (nevhodný údaj)

Byla zadána špatná funkce nebo hodnota.

B Integer out of range (celé číslo mimo rozsah)

Hodnota byla zaokrouhlena na nejbližší celé číslo, které je mimo rozsah, který může být přijat.

C Nonsense in BASIC (nesmysl v programovacím jazyce BASIC)

Příkaz nebyl vydán v jazyce, který dává smysl v Basicu.

D BREAK - CONT repeats

BREAK bylo stisknuto. Opakování příkazu pomocí CONTINUE, kdy dojde k pokračování na daném zastaveném místě.

E Out of DATA

READ se pokusil číst za koncem konečného příkazu DATA v programu.

F Invalid file name (neplatný název souboru)

SAVE bylo použito se jménem obsahujícím více než 10 typů.

G No room for line (není místo pro řádku)

Není dostatečné paměťové místo pro vstup nových programových řádků.

H STOP in INPUT

Bylo zakódováno STOP v závislosti na INPUT nebo začalo zavádět data. CONTINUE opakuje příkaz INPUT.

I FOR without NEXT

Smyčka FOR NEXT nebyla provedena, neboť limity nebo hodnoty STEP byly chybné (např. FOR x=5 TO 0 bez STEP) a odpovídající NEXT nebylo nalezeno.

J Invalid I/O device

Zpráva pro manipulaci s microdrivy (vadné zařízení vstupů a výstupů).

K Invalid colour (chybná barva)

Hodnota udaná pro INK, PAPER, FLASH, BRIGHT, INVERSE nebo OVER nebo odpovídající řídicí typ jsou mimo rozsah.

L BREAK into program

Bylo stisknuto tlačítko BREAK. Zpráva udává, jaký byl poslední příkaz, který má být prováděn, a klíčové slovo CONTINUE pokračuje od dalšího příkazu.

M RAMTOP no good

Hodnota uvedená pro RAMTOP je buď příliš velká nebo příliš malá.

N Statement lost

Byl proveden skok na příkaz, který již neexistuje.

O Invalid stream

Zpráva pro manipulaci s microdrivy.

P FN without DEF

Byl vydán příkaz FN bez odpovídajícího příkazu DEF FN.

Q Parameter error

Příkaz FN obsahuje špatnou hodnotu, která má být podrobena funkci, nebo jedna z hodnot je špatného typu (textová místo číselná nebo naopak).

R Tape loading error

Závada při plnění nebo ověřování.

## Co je za BASICem

Basic je všeobecný počítačový jazyk, který pracuje velice dobře pro většinu aplikací. Samozřejmě to není jediný jazyk, který můžete použít na počítači Spectrum. Software, které udává jiné jazyky, jako FORTH, micro-PROLOG a LOGO, je možno rovněž použít. Tyto jazyky pracují zcela odlišným způsobem a otevírají nové možnosti vašemu počítači.

S ohledem na to, že Basic je všeobecný jazyk, může být poněkud těžkopádný pro některé aplikace. Je navíc poměrně pomalý. Jiné jazyky mohou udávat vyšší pružnost kombinovanou s jednoduchostí programování a vyšší rychlostí. FORTH např. umožňuje definici vlastních slov a jejich užití v instrukcích, kterým počítač rozumí a které vykonává přibližně 10x rychleji než obdobné povely v jazyce Basic. S micro-PROLOG počítač porozumí jednoduchým anglickým frázím. Uschovává je ve své paměti jako báze pro dialog s uživatelem. LOGO je počítačový jazyk vyvinutý pro výukové využití. Jeho rysy spočívají ve velice jednoduchých povelích, které mohou být užity velmi pružně. Chcete-li psát skutečně velmi rychlé programy pro váš počítač ZX Spectrum+, budete potřebovat pochopit, jak programovat ve strojovém kódu.

## Strojový kód

Basic je používán pro umožnění zadání instrukcí počítači ve formě, která je snadno pochopitelná. Spectrum CPU - čip Z80A - defakto nerozumí jazyku Basic. Část paměti obsahuje trvalý program nazývaný převaděč Basic, který převádí vaše instrukce v jazyce Basic do soustavy kódových signálů. Tyto kódy ve skutečnosti řídí Z80A, aby zpracovával vaše instrukce.

Převaděč zabírá poměrný čas pro překlad vašich instrukcí v jazyce Basic do kódu Z80A nebo strojového kódu.

Můžete obejít jazyk Basic - přejete-li si zaslat strojový kód přímo do Z80A. Program bude poté prováděn velice rychle. Cena, kterou za to zaplatíte, je zvláštní čas potřebný pro napsání programu ve strojovém kódu. Na rozdíl od Basicu je velmi "nepřátelský", což znamená, že je poměrně těžké se mu naučit. Programování ve strojovém kódu je mimo vymezení této knížky. Přesto existuje mnoho knížek, které učí, jak kódovat Spectrum ve strojovém kódu. Pro získání představy o rychlosti strojového kódu запиšte a nechte proběhnout následující krátký demonstrační program.

Program - str. 75 originálu.

Vyzkoušejte změnit hodnotu 55 v ř. 50 na jakoukoli hodnotu od 1 do 255 a pozorujte změnu pruhů. Rovněž můžete vyrobit barevné pruhy pomocí užití povelu INK. Toto však není účelem programu. Nyní sledujte, jak rychle je display vyráběn užitím strojového kódu - je zdánlivě mžikový. V Basicu zaujme přes 2 sekundy naplnění obrazovky.

Tento program pracuje, protože příkazy DATA obsahují 16 kódů, které jsou uloženy na adresách 65000 vlastních v paměti podle řádků 10 až 30. Řádek 70 zasílá kódy do Z80A a display je vyráběn průběžně. Osmý kód řídí šířku pruhů. Mnoho her, které jsou dostupné pro užití na vašem počítači, jsou napsány ve strojovém kódu, pro vyvinutí této ultra-rychlé činnosti. Pro pomoc zápisu strojového kódu jsou programy nazývané assembly. Tyto umožňují instrukce, která zapisujete namísto čísel, která jsou taková, jaké strojový kód sám vyžaduje. Instrukce nejsou anglická slova jako klíčová slova Basicu, avšak zkratky nebo mnemotechnické pomůcky s představením činnosti, kterou musí počítač podstoupit. Musíte mít proto pochopení a porozumět, jak počítač pracuje krok za krokem dříve, než můžete použít tohoto jazyka.

## Počítačový žargon - co to znamená

Mnohá slova užívaná ve výpočetní technice se používají i v každodenním životě, avšak mnohdy s odlišným významem. Zde vám předkládáme vysvětlení významů některých těchto slov, která se objevila v této příručce, ve spojení se speciálními termíny. Latinská slova mají vysvětlení sama v sobě. Narazíte-li na slovo nebo větu, která vám není jasná a není v tomto výčtu, pokuste se jí vyhledat v indexu.

**ADRESA A** - jednoduchá jednotka paměti. Adres je v počítači ZX Spectrum+ 65 536.

**ARGUMENT A** - hodnota, která je přiřazena k funkci pro získání výsledku.

**ARRAY A** - oblast (pole) - skupina údajů, která je uložena pohromadě v jedné části paměti.

**ATTRIBUTES** - rysy - kódy, které udávají barvy typů.

**BASIC** - počítačový jazyk užitý pro ZX Spectrum+ a většinu domácích mikropočítačů.

**BINARY CODE** - binární kód - dvojková soustava - druh kódu, který je užíván v počítači. Je složen z posloupnosti stavů ano/ne, např. úrovní napětí log1 a log0.

**BIT** - stav jedné logické hodnoty (ano nebo ne).

**BYTE** - skupina osmi bitů, která reprezentuje číslo o hodnotě 0 až 255 (dek.). Každá adresa v paměti obsahuje jeden byte.

**CHARACTER** - typ - jakékoli písmeno, číslice (0 až 9), zna-

ménko nebo grafická jednotka, které mohou být vytištěny nebo zobrazeny.

CHARACTER SET - kompletní soubor předem nastavených typů a určitých řídicích kódů užívaný počítačem.

COMMAND **A** - povel A - jednoduchá instrukce, která je prováděna počítačem, nebo přímý povel.

CONCATENATION - Slučování - kombinace stringů slučováním dohromady.

CONSTANT **A** - konstanta - číslo nebo skupina jednoho nebo více písmen nebo jiných typů.

CPU (Central Processing Unit) - ústřední mikroprocesor -  
- centrální část počítače, která provádí výpočty a řídí další jednotky. ZX Spectrum+ užívá mikroprocesor Z80.

CURSOR - kurzor - poloha na obrazovce, kde má být něco zobrazeno. Může být vyznačen blikajícím **znaménkem** označujícím mód, ve kterém je právě počítač (mode = stav).

DATA - informace, která počítač získal z programu nebo jsou do **něho zakládány z počítače v posloupnosti** pro získání výsledků.

DIRECT COMMAND **A** - přímý povel - soubor jedné nebo více instrukcí, který je prováděn bezprostředně po zadání počítači.

EDIT - vydání - mění podrobnosti uvnitř programu.

ENTER - vstup - dává kompletní instrukce nebo informace počítači.



**EXPRESSION A** - vyjádření - věta - kombinace konstant, proměnných a klíčových slov.

**FALSE** - chyba - veškeré stavy nebo výsledky, které počítač zhodnotí jako nepravdivé nebo **nepřesné**. Chyba má číselnou hodnotu 0.

**FUNCTION** - činnost - funkce - počítač vybírá jednu nebo více proměnných (nebo argumentů) a užívá je pro získání výsledků jiné hodnoty.

**GRAPHICS** - grafika - vytváření **představ**, jako jsou obrázky, grafy a diagramy, počítačem.

**HARDWARE** - počítač sám a veškerá připojená zařízení nebo stroje jako periferie.

**INFORMATION** - informační slova, čísla a znaménka v jakékoli kombinaci, které počítač vyžaduje pro manipulaci.

**INPUT** - vstup - programy a údaje zadávané do počítače.

**INTERFACE A** - jednotka, která propojuje počítač a/nebo periferie spolu a která zajišťuje, že mohou navzájem komunikovat.

**K** - míra kapacity paměti počítače. 1K je rovno 1 kilobytu nebo 1024 bytů. Paměťová kapacita v K je rovna celkovému počtu adres v paměti, z nichž každá může uschovat jeden byte. ZX SPECTRUM+ má 48K RAM a 16K ROM, celkem 64K.

**KEYWORD** - klíčové slovo - počítačová instrukce v jazyce Basic. Může vyžadovat některé hodnoty pro svou činnost.

**LINE** - řádek - instrukce nebo sada instrukcí v programu. Je daná číslem, takže je prováděna v přesném pořadí podle dalších řádků.

**LISTING** - řádky programu vypsané v posloupnosti.

**LOAD** - zakládání - plnění programu nebo dat do počítače z úložných zařízení jako jsou náboje nebo kazety.

**LOGIC** - proces, kterým počítač rozhoduje, zda výsledky jsou správné nebo špatné, dále zda stavy jsou pravdivé nebo nepravdivé.

**LOOP** - smyčka - část programu, která se opakuje jednou nebo vícekrát.

**MACHINE CODE** - strojový kód - jazyk, kterému rozumí ZX SPECTRUM+. Programy v Basicu jsou překládány do strojového kódu počítačem, který je zpracovává.

**MEMORY** - paměť - část počítače, která uschovává program a data podle požadavků a rovněž uschovává stálé pracovní instrukce.

**MODE** - ve Spectru je jeden z pěti stavů, který ukládá, které klíčové slovo a typ mohou být tou kterou klávesou na klávesnici vydány. V průběhu programování je mód indikován pomocí blikajícího písmenka uvnitř kurzoru.

**NESTING** - vzájemné prolínání - uzpůsobení smyček - prolínání jedné do druhé, kdy v rámci programu jedné smyčky může být prováděna smyčka další.

NUMERIC VARIABLE - číselná proměnná - proměnná, která udržuje číslo. Číselné proměnné sestávají z jednoho nebo více písmen.

OPERATOR - instrukce, která je činná buď aritmeticky nebo logicky.

OUTPUT - výsledky vydávané počítačem.

PERIPHERAL - periferie - veškerá zařízení, která jsou připojena k počítači.

PIXEL - nejmenší barevný bod, který se může objevit na obrazovce. **Krátký** pro "obrazovou buňku".

PRINT - tisk - zobrazuje buď výsledky nebo grafiku na obrazovce nebo je tiskne pomocí tiskárny.

PROGRAM - sled instrukcí, které mají být prováděny počítačem.

RAM (Random Access Memory) - paměť s náhodným vstupem - část paměti, která může být naplněna údaji a jinými proměnnými hodnotami. Rovněž nazývána těkavou pamětí. RAM - obsah je vymazán poté, co je odpojena energie. ZX SPECTRUM+ má RAM 48K.

REGISTER - malé paměťové jednotky oddělené od hlavní paměti. Registry uvnitř CPU jsou užity pro provádění procesů počítání.

REPORT - zpráva zobrazená počítačem o průběhu jeho činností.

RESOLUTION - rozklad - stupeň podrobností možných při vykreslení grafiky.

ROM (Read Only Memory) - část paměti obsahující stálé programy a instrukce pro počítač. ZX Spectrum+ má 16K ROM. ROM nelze smazat.

SAVE - úschova - uschování programu nebo dat v úložném zařízení jako je **náboj** nebo kazeta.

SCROLL - posuv - pohyb, který umožňuje displayi přesáhnout rozměr jedné obrazovky.

SOFTWARE - veškeré programy včetně stálých programů v ROM nebo cartridžích.

STATEMENT - příkaz - buď klíčové slovo, které je použito pro vytvoření instrukce v programovém řádku nebo instrukce sama.

STRING - řetězec písmen (text) - řetězcová proměnná - skupina jednoho nebo více typů uzavřená do uvozovek, která je odlišuje od čísel a číselných proměnných.

STRING VARIABLE - řetězcová proměnná - proměnná, která udržuje string. Řetězcové proměnné vždy sestávají z jednoho písmene a znaménka \$ - americký dolar.

SYNTAX - správná posloupnost klíčových slov, konstant, proměnných a vyjádření (vět), požadovaných pro tvorbu platných instrukcí Basic.

TRUE - pravdivý - jakýkoli stav nebo výsledek, který počítač shledá jako pravdivý nebo **přesný**. Pravdivost má číselnou hodnotu 1.

VALUE - hodnota - jakékoli číslo nebo řetězec písmen (text), které může být dáno nebo představeno konstantou, proměnnou nebo vyjádřením (větou).

VARIABLE - proměnná - jedna nebo více jednotek v paměti, která udržuje dílčí konstanty pro použití počítačem. Každá je dána pomocí jména nebo písmena pro snadnou identifikaci. ZX Spectrum+ rozlišuje mezi číselnými proměnnými a řetězcovými proměnnými.

## INDEX

Čísla stránek psaná v originálu **proložene** se vztahují na ilustrace a nadpisy - záhlaví.

Seznam, který následuje, je seřazen podle originálu.

Anténní zdířka a kabel 4-5

Obměna programů 9

Zesilování zvuku 37

Animace 34-5

Aritmetické operátory 22;22

ATTR 35

Sloupcové diagramy 25;25

BASIC 18,49-73

BEEP 36;18

BIN 33

Binární kód 44

Barva okraje 24-5;6

Program odraženého míče 35

Závorky 23

BREAK 19

Bright 31

Výpočty 22-3;22,23

Mód velkých písmen 21;20

CAPS LOCK 21;18

CAPS SHIFT 8,21;18

Cartridge, microdrive 12,46;46

Kazetový magnetofon jako zesilovač 37;37

-volba 12

-připojení 5,13;13

-počítadlo 14

-zakládání programů 14-16

-uschování programů 38-40

-nastavení tónových korekcí 14,15,16

-nastavení hlasitosti 14,15,16

Magnetofonové kazety 12,44, 45

- skladování 12

- označení 14

- zvuk 12

- uskladnění 12

Centrální procesorová jednotka (CPU) 43, 44, 48, 75; 43, 45

Tvoření typů 32-3

Volba typů 20

Souprava typů 51

Šachovnice - program 33

Čipy 42-3

CIRCLE 28

Srážky 34-5

Sloupec 23, 51

Barva 24-5;24-5

- kódy 24

- řídicí kódy 33

- kombinace 25

- zobrazení kláves 19

- míchání 32

- zkoušení 6; 24

Čárka 23, 51

Povely 22, 50

Připojení 5

- magnetofon 13

- síť 5

- televizor 4

Řízení kurzoru 19

DATA 33

DELETE 10

DRAW 28-9

Zdířka EAR 37; 5, 13  
Přímý konektor 5, 43, 47  
EDIT 18, 21  
- řádky programu 21  
ENTER 9, 10, 11, 19  
Zakládání programu 8-9  
Chyby, opravy 10, 21  
- zprávy na obrazovce 74  
EXTEND MODE 8, 21; 18  
- rozšíření 21; 20  
  
FLASH 31  
Program Blikající kruh 9  
FOR NEXT 26-7, 29, 30, 31, 34  
FORTH 75  
Úplné zastavení 23, 51  
Funkce 50  
  
GOTO 23  
GRAPH 21; 18, 26  
Grafiky, animace 34-5  
- barvy 24-5  
- tvoření typů 32-3  
- zakládání souborů vzorů 29;29  
- vysoký rozklad 26, 28-9  
- nízký rozklad 26-7  
- vzory 30-1  
- náhodné efekty 30  
Grafický mode 21; 20  
Rastr, vysoký rozklad 28, 80  
- nízký rozklad 26, 80  
Hardware - definice 12  
Grafiky s vysokým rozkladem 26, 28-9

IF THEN 29

Barva tisku 24-5

INPUT 23, 29

Vstupní a výstupní propoje 45

Interface 45, 46-7

INV VIDEO 18

INVERSE 31

Joystick - ruční ovladač pro hry 45, 47

Klávesnice 18-19

- typy grafik 26

- módy 20-1

Klíčování - zakládání 8, 9

Klávesy 18-9;18-9

- jejich činnost 20-1;20-1

Mode klíčových slov 20; 20

Klíčová slova 9, 18-19, 50, 52-73, 20-1

- volba 19, 20

LET 23

Písmenový mode 21; 20

Řádky 8

- vymazávání 21

- vyhledávání a opravy 21

LIST 21

Vypisování programu 8, 21

LOAD 14-16

Plnění programem 13, 14-15; 14-16

Logické integrované obvody 43

LOGO 75

Cykly 26-7, 30

Reproduktor 43

Nízkorozkladové grafiky 26-7



Strojový kód 75

Program Šílená mozaika 10

Paměť 12, 42-3, 44-8

Paměťová mapa 48

Zdířka MIC 37; 5, 13

Microdrivy 46; 5, 46

- cartridge 12, 45

- jejich plnění 46

Micro-PROLOG 75

Omyly, opravy 10, 21

Modemy 46

Mody 20-1

Program Tabulka násobení 23

Hudba 36-7

Názvy programů 8

NEW 11, 12; 18

Nové programy 11

9 Voltů DC zdířka 5, 43

Čísla kláves 19

Číslice 50

Barva papíru, podkladu 24-5

Program Vzory 9

Periferie 45, 46-7

Obrazy, konstrukce 30-1

- hrubá grafika 26-7

Výška tónu 36

Obrazové buňky 28

PLOT 28

Tečka 23, 51

POKE 48

Program Polyhedra 10

Napájecí zdroj 4, 5; 5, 43

PRINT 22

Tiskárny 45, 47; 45, 47

Řádky programu, smazání 21, opravy 21

Programování 17-40

Programy, výměna 9

- začátek nových 11

- oprava chyb 10

- zakládání 8-9, 44

- plnění 12, 13, 14-15; 14-15

- opětovné spuštění 10

- rozběh 8-9, 44

- ukládání 13, 38-40

- ověřování 39

Interpunkční znaménka 23, 51

Program Pyramidy 31

Uvozovky 23, 51

Radiové rušení 4

Program Duna 26-7

RAM (Random Access Memory) 42, 48; 42, 45

RAM - přídatná paměť 4

RAMTOP 48

Náhodný efekt 30

READ 33

Připravení software 12-13; 13

REM 39

Opětovné spuštění 11, 12; 5

Opětovné spuštění programu 10

Pružný plochý pásový vodič 46

RND 26, 30

ROM (Read Only Memory) 48; 43, 45

ROM cartridge 12, 47; 46-7

RS232 Interface 47; 45

Rozběh programů 8-9

SAVE 38-9  
Ukládání 13, 38-40  
Zprávy z obrazovky 74  
Posouvání 8  
Středník 23, 51  
Formy, jejich plnění 29; 29  
Program Východ slunce 11  
Znaménka, výpočty 22, 50  
- výběr 19  
Sinclair Basic 49-73  
Program Kreslení 29  
Zdířky 5  
Software 12  
- plnění 14-6;14-6  
- příprava ke zpuštění 12-13; 13  
- vhodnost 12  
- typy 12  
Zvukové efekty 36-7  
Pomlka 19  
Program Čtverce 30  
Program Hvězdy 28  
Program Hvězdy a pruhy 11  
Příkazy 22, 50  
STEP 29  
Ukládání 44, 45  
Písmena 22  
Podprogramy 30-1  
SYMBOL SHIFT 8, 21; 19  
Symboly, výběr 20  
Program Symetrické vzory 30  
Systémové proměnné 48  
  
Pásy 12, 45  
- manipulace 12  
- označení 14, 39  
- zvuk 12  
- uskladnění 12

Televizní přijímač, připojení 5

- vhodnost 4

- ladění 6; 6

Nastavení tónových korekcí, kazetový magnetofon 12, 14, 15

TRUE VIDEO 18

Ladění televizoru 6; 6

Televizní kodér 42

ULA 42

Typy stanovené uživatelem 80; 32-33

Proměnné 22-3,50

Regulátor napětí 43

Nastavení hlasitosti, kazetový magnetofon 12, 14, 15

Z80 Mikroprocesor 43, 75; 45

ZX Interface 1 45, 46-7

Program ZX Robot 27

ZX 16K RAM 4

Poprvé vydáno v roce 1984 panem Dorling Kindersleyem ,  
Henrietta Street , London ve spolupráci s Sinclair  
Research Ltd, Cambridge

Copyright 1984

ZX Spectrum+, ZX Microdrive a ZX Interface jsou obchodní  
známky Sinclair Research Limited.

Veškerá práva jsou vyhrazena. Žádná část této publikace  
nesmí být reprodukována, uschována v přenosové soustavě  
nebo přenášená jinou formou nebo jinými způsoby, elektro-  
nický, mechanický, fotografováním, záznamem nebo jinak,  
bez předchozího písemného povolení vlastníků Copyrightu.

### Mříž - rozložení displaye na obrazovce

Mříž ukazuje souřadnice pro nízko- i vysokorozkladovou grafiku. Ačkoli jednotlivá klíčová slova pro grafiky pracují buď se souřadnicemi nízkého nebo vysokého rozkladu, můžete užít obou - nízkého i vysokého rozkladu při **vydání** displaye.

Nízkorozkladová mříž se dělí do dvou částí, hlavní displayová plocha a dvě řádky na spodní části obrazovky. Klíčová slova PRINT AT vydají typy v horní části a klíčová slova INPUT AT ve spodní části. Souřadnice nízkého rozkladu jsou ukázány v horní levé části mříže.

Vysoký rozklad zabírá pouze hlavní displayovou plochu.

Jsou použita klíčová slova PLOT, DRAW a CIRCLE.

Souřadnice vysokého rozkladu jsou udány na pravé dolní straně mříže.

Str. 80 originálu dole - síť grafiky vytvářené uživatelem:

Přidejte čísla, která jsou zadána, a získáte **desetinnou** hodnotu pro každou řádku sítě, poté použijte klíčová slova POKE USR, čímž programujete typ.

### Doplňková literatura

Předkládáme seznam knih pro veškeré majitele nových počítačů ZX Spectrum +. Doporučujeme je pro kohokoli, kdo se chce naučit více o programování a přeje si využít celý potenciál ZX Spectrum+. Můžete objednat jakoukoli z těchto knih vyplněním štítku připojeného ke knížce.

Poznámka: Ačkoli většina z těchto knih je napsána pro originál ZX Spectrum, veškeré programy, které obsahují, budou fungovat i na vašem ZX Spectrum+.

Programování pro ZX Spectrum+ krok za krokem

- kompletní učebnice pro samouky - programovací kurzy, včetně exkluzivních barevných fotografií. Je to první dostupná kniha pro ZX Spectrum+.

Kniha jedna

- praktický průvodce programováním v jazyce BASIC pro úplné začátečníky - překrývá všechno od prvního programování až po vytváření barevných grafik a ožívování. Ilustrováno více než 150 fotografiemi záběrů obrazovky.

Kniha dvě

- další techniky pro zkušenější programátory - baleno s typy na zvukové a zvláštní efekty. Vytváření grafik a grafů. Zdokonalené programy, oživení a průvodce při skládání her arkádového stylu.

Very Basic BASIC - ZX Spectrum

Pro první uživatele - důležitý průvodce pro prvních 15 hodin vlastníka ZX Spectrum.

30 hodin BASIC - ZX Spectrum vydání

- jednoduchá knížka pro samouky k naučení základů Basicu na ZX Spectrum.

Kniha her Spectrum

21 vzrušujících velmi kvalitních her - úplné s vypsáním programy, instrukcemi a vysvětleními.

40 vzdělávacích her pro Spectrum

- učení je zábavné! Souhrn programů konstruovaných pro pomoc mladým při manipulaci s počítačem - zdokonalení jejich všeobecných znalostí.

Grafiky a zvuky Spectrum

- jak využít co nejvíce schopností Spectra při tvoření grafik a zvuků - techniky kreslení, grafů, barevného ožiování, zvukových efektů, hudby a her.

Strojový jazyk Spectrum pro absolutní začátečníky

- praktický úvod do psaní rychlých, silných a úsporných programů a podprogramů ve strojovém kódu - včetně úplného výpisu vzrušujících arkádových her ve strojovém jazyce.

Strojový kód Spectrum

- jestliže nevíte nic o strojovém kódu, ale chcete se jej naučit, zde je šance! Zjistíte, jak vyrábět rychlé grafiky, nestálé barevné změny, posouvání a rychlé hledání a mnoho více.

Jak získat co nejvíc z vašich ZX Microdrivů

- vše co potřebujete vědět, abyste získali nejlepší využití vlastního ZX Microdrivu a Interface - s mnoho praktickými příklady a úplnými připravenými programy.

100 programů pro ZX Spectrum

100 odlišných výpisů programů včetně kazetového záznamu se všemi programy. Unikátní knihovna software s hrami, obchodními aplikacemi, grafikami, vědeckými a matematickými programy a programy pro použití v domácnosti.



Elektronická verzia s korektúrami v poznámkach: 27.12.2007  
Peter Turányi alias Softhouse  
<http://softhouse.speccy.cz>