

C9 RET ;mesto za povratek, če ni bilo skoka.
 03 Skok INC BC
 C9 RET

Naslednji primer vključuje ukaz DJNZ (ukazu RST 10, ki tudi nastopa, se bomo posvetili v kasnejših poglavjih). Na zaslon bo izpisal Spectrumov znakovni niz, z grafičnimi znaki vred.

```
0670 LD B, 70H ;zanka naj se ponovi 112-krat
3E90 Zanka LD A, 90H
90 SUB B
D7 RST 10 ;izpiši znak
10FA DJNZ Zanka
C9 RET
```

Podrobnosti programa vam (najbrž) še niso povsem razumljive, pa si zato ne belite glave. Ko preberete poglavje "Izpisovanje na zaslon", o težavah z razumevanjem ne bo ne duha ne sluha.

PODPROGRAMI

Ko načrtujemo program, opazimo, da so nekatera opravila ali izračuni potrebni večkrat na različnih mestih v programu. Dele programa, ki opravljajo takšne naloge, izluščimo iz programa in jih izdelamo neodvisno od njega. Imenujemo jih podprogrami. Njihova vloga je podobna vlogi podprogramov ali definiranih funkcij v basicu. Podobna je tudi njihova uporaba. Celo funkcija USR pravzaprav obravnava naše strojne programe kot podprograme, saj moramo na koncu vedno uporabiti ukaz RET (RETURN = vrni se). Dobro in lepo napisan program pogosto uporablja podprograme (učeno rečemo, da je strukturiran). V takšnem programu lahko tudi marsikateri dolg skok nadomestimo z uporabo podprograma.

Iste podprograme lahko uporabljamo skupaj z različnimi glavnimi programi. To je celo zelo v navadi. Tako si ustvarimo zalogo podprogramov (navadno ji pravimo knjižnica podprogramov), ki je zelo dragocena, če ti programi opravljajo pogosto potrebno delo. Vzemimo nekoliko poenostavljen primer: če bi se na debelo ukvarjali s pisanjem iger v strojnem jeziku, bi pogosto rabili zaključen del programa, ki bi ob koncu igre na zaslon izpisal "KONEC" ter število doseženih točk. Takšen del bi oblikovali kot podprogram in bi ga lahko priključili vsem igram. Če bi se ukvarjali s statistiko, bi bil npr. eden od podprogramov namenjen izračunu poprečne vrednosti in podobno.

Ker so podprogrami samostojni deli, jih lahko tudi preizkušamo neodvisno od glavnega programa. Tako z uporabo podprogramov skrajšamo čas pisanja programov, njihovo preizkušanje in prevajanje v strojni kod. Takšni programi so tudi bolj čitljivi in lažje razumljivi. Vse to precej olajša delo.

Kot vsi dobri programi, je tudi Spectrumov nadzorni program strukturiran. Njegove podprograme (z njimi se boste seznanili npr. s pomočjo knjige I. Logana in F. O'Hare The Complete Spectrum ROM Disassembly, Melbourne House Publishers, Cheddington 1983) lahko uporabite v svojih programih, kar pomeni poleg omenjenih prednosti tudi precej krajši glavni program.

Ukaz za skok v podprogram imenujemo klic podprograma in ima splošno obliko