To so že precej natančna navodila, kar tudi želimo. Drugo zlato pravilo je namreč: koraki naj bodo majhni. Vsebujejo naj kar najmanj odločitev (pogojnih skokov, pogojnih klicev ipd.) - če je mogoče, le eno. Z razčlenitvijo dobimo iz posameznih korakov samostojne dele, ki jih lahko tudi samostojno preizkušamo (spomnite se podprogramov!). In še tretje pravilo: vse podrobnosti prihranimo za konec.

V načrtu oblikujemo eno glavno vejo (glavni program) in več stranskih. Glavni program naj vsebuje ukaze oziroma opravila, ki bodo največkrat izvajani. Stranske veje naj obsegajo ostala, sorazmerno redkeje potrebna opravila. Paziti je treba, da v glavnem programu ni korakov, ki jih lahko uvrstimo v stranske veje. Izvajanje programa bo tako hitrejše, razumevanje pa lažje.

Ko pripravljamo obsežnejše programe, pride prav "dnevnik". To so zapiski sprememb, pripomb, vprašanj... Delo ter kasnejše prebiranje in izdelava komentarja so na ta način lažji.

KODIRANJE

Opazili ste, da smo posameznim korakom dali imena - pravimo jim tudi simbolični naslovi (npr. Cilj, Konec). Ta kratka imena, ki povedo, kaj se dogaja v posameznem koraku, so zelo praktična že pri izdelavi algoritma. Ko kodiramo, moramo te naslove nadomestiti z absolutnimi števili - naslovi v pomnilniku. Pri delu z zbirnikom lahko nadomeščamo posredno, če pa uporabljamo urejevalnik, moramo simbolične naslove takoj (neposredno) nadomestiti z absolutnimi. Pri neposrednem nadomeščanju je treba paziti na pravi vrstni red! Nizki zlog naslova je v pomnilniku prvi, visoki zlog drugi. Pomote pri zapisu so pogost vir napak.

Druga "neprijetna" skupina so pri kodiranju relativni skoki. Pri izračunu odmikov je potrebna precejšnja pazljivost (poglejte še enkrat poglavje o skokih). Poleg tega je treba pri dopolnjevanju programa (dodajanju ali odvzemanju ukazov) preveriti, ali nismo spremenili odmika kakega kratkega skoka.

KAM SPRAVITI PROGRAM ?

Vprašanje, kam spraviti program v strojnem jeziku, je bilo pri nekaterih računalnikih (npr. ZX 81) velik problem. Za ZX Spectrum tega nikakor ne moremo trditi. Prostorov, kamor lahko shranjujemo strojni kod, je celo več.

PROSTI POMNILNIK

Z ukazom CLEAR nn rezerviramo pomnilniški prostor od naslova "nn+1" naprej. Na ta način nam je popolnoma na voljo del pomnilnika, zavarovan pred vplivi programa v basicu in pred posegi nadzornega programa. Najbolje je postaviti programe tik pod vrh pomnilnika. Tako jih ne bo potrebno premikati, kadar bomo uporabljali večje podatkovne zbirke ali kadar bomo dodajali nove programe. Takšen način shranjevanja - nad mejo basicovega področja - je najbolj enostaven in brez vsakršnih preglavic. Zato vam ga najbolj priporočamo.

Poleg te možnosti je seveda še nekaj drugih. Oglejmo si jih!

UPORABA VRSTICE REM

Prva takšna možnost so REM vrstice programa v basicu. Če je REM prva vrstica programa v basicu, lahko znake v njej nadomestimo z zlogi strojnega programa. Recimo, da je ta 40 zlogov dolg. Kot prvo vrsto v basicu napišemo

```
1  REM 1234567890123456789012345678901234567890
```

Vrsta ima 40 znakov. Če bi imel naš strojni program 60 zlogov, bi potrebovali 60 znakov; če bi imel le tri zloge, bi v vrstico lahko zapisali le tri. Popolnoma vseeno je tudi, katere znake uporabimo. Prednost našega načina je, da natančno vemo, koliko znakov je v vrstici. Tako le stežka pride do napak.

Uporaba REM vrste ima svoje prednosti in pomanjkljivosti. Poglejmo najprej pomanjkljivosti.