

ga programiranja. Konec koncev je tudi v zbirnem jeziku moč strukturirano programirati. Klasičen primer je jezik FORTRAN, ki ni strukturiran, vendar so ga izboljšali in naredili STRUKTRAN, strukturiran jezik. Primer dobro strukturiranega jezika je npr. PASCAL.

Stavčni konstrukti

V višjih jezikih se pojavljajo konstrukti IF-THEN-ELSE, DO UNTIL, DO WHILE in še drugi. Oglejmo si vsaj prve tri.

IF-THEN-ELSE stavek omogoča odločitve glede ne določene pogoje. Telo stavka je takole:

IF (pogoj), THEN blok programa, če je bil pogoj izpolnjen ELSE blok programa, če pogoj ni bil izpolnjen ENDIF

Navadno je lahko (pogoj) poljubno sestavljen stavek. Nekaj primerov pogojev: (A=B), ((A=B) in (B=C)), ((A=B) ali (B=C)), itd. Kadar je pogoj izpolnjen, se izvaja blok takoj za THEN, sicer se izvede blok za ELSE, nikdar pa ne oba. Ključne besede pomenijo:

IF = če
THEN = potem
ELSE = sicer

ENDIF = konec »če« stavka

DO UNTIL ali REPEAT UNTIL (oboje pomeni »ponavljal, dokler ni«) tudi pregleduje pogoj in če ta ni izpolnjen, ponavlja programski blok tega stavka, sicer zaključi operacijo. Prva ponovitev se izvede ne glede na to, če je pogoj izpolnjen ali ne, ker testiranje pogoja poteka na koncu programskega bloka. Telo stavka je takole:

DO UNTIL (pogoj) programski blok, ki se izvaja, če pogoj ni izpolnjen

ENDDO
DO WHILE ali REPEAT WHILE (oboje pomeni »ponavljal, dokler je«) tudi pregleduje pogoj in če je izpolnjen, potem se izvaja programski blok v telesu stavka, sicer se ponavljanje konča. Testiranje pogoja je na začetku stavka (pred programskim blokom). Telo WHILE stavka ima obliko:

DO WHILE (pogoj) programski blok, ki se izvaja, dokler je pogoj izpolnjen

ENDDO
Obstajajo še drugi stavčni konstrukti, a omenjeni so značilni za mnoge jezike, čeprav so sicer različni. Na-

mesto DO WHILE lahko jezik zahteva WHILE (pogoj) DO ali še kako drugače, vendar je funkcija enaka.

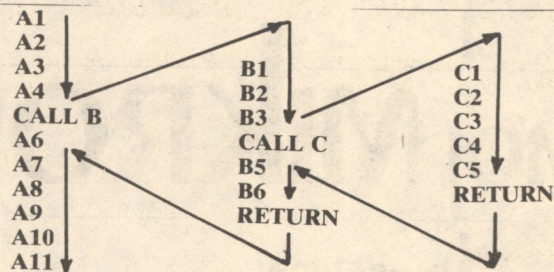
K pravilom strukturiranega programiranja sodi tudi pojem MODULAR-NOST. A za to si moramo najprej pogledati pojem podprograma ali subrutine.

Podprogrami

Ko smo govorili o zbirnem jeziku, smo omenili inštrukciji CALL in RETURN. Obe sta potrebni pri klicanju podprogramov. Podprogram je manjši program, ki ga kličemo iz kakega drugega programa. Ko pokličemo podprogram, si mora računalnik zapomniti naslov naslednjega stavka, ki ga bo moral izpeljati, ko bo podprogram končan. V ta namen uporablja podatkovno strukturo SKLAD, ki smo jo že omenili, da bomo o njej morali še precej povedati kasneje. Zaenkrat povejmo le to, da se pri klicu podprograma z inštrukcijo CALL naslov naslednjega programskega stavka shrani v sklad, kjer čaka, dokler ni podprogram končan. Na koncu podprograma je vedno inštrukcija RETURN, ki iz sklada po-

tegne shranjeni naslov in vrne izvajanje v program, ki je poklical podprogram. Podprogrami so lahko gnezdeni drug v drugem.

Princip je shematično prikazan spodaj. Simboli (npr. A2) predstavljajo programske stavke. Puščice kažejo tok izvajanja.



V zgornjem primeru se stavki izvajajo po naslednjem vrstnem redu: A1 A2 A3 A4 B1 B2 B3 C1 C2 C3 C4 C5 B5 B6 A6 A7 A8 A9 A10 A11. V resnici pa

so podprogrami zapisani takole:



MALI OGLASI

- za prodajo opreme, literature
- za nakup
- za izmenjavo
- za najrazličnejše storitve

SINCLAR 48 K, nov, prodam. Ponudbe pod »4,4 M«. TX-507
ZX SPECTRUM 48 K, ZX 81, najpopolnejši slovenski prevod navodil za uporabo računalnika in programiranje v basic, najnovejši in najboljše programe za spectrum in ZX 81 (MANIC – MINER I in II, ATIC – ATTACK, OSTRON, PHENIX inš.), prodam. Telefon (061) 447-156. TX-508
SINCLAIR ZX 81 16 KB RAM prodam za 2 M. V ceno vključeno 2 uri instrukcij. Informacije vsak dan od 18. ure naprej na telefon (061) 444-274. TX-505

MATELL, RAČUNALNIK za video igre, malo rabljen, ugodno prodam. Ponudbe pod »2,9 M«. TX-124
ZA ZX SPECTRUM prodam 100 odličnih programov iz časopisa finclair user za 1000 din. Programi so: igre, dobivanje 56 barv, 42 črk v vrsti, scrolling 3 d risanje, spectrum kot osciloskop itd. Tone Kordeš, Mestni trg 9, 64220 Škofja Loka. TX-125

HIŠNI RAČUNALNIK COMMODORE VC 20, skupaj s kasetarnjem prodam za 5 M. Telefon (061) 445-839, dopoldne. TX-120

ZX 81, 16 K s programskimi kasetami (šah, bioritem, igre) prodam za 21.000 din. Tudi demonstriram in pojasnim. Miro Ham, Brilejeva 9, Ljubljana, telefon (061) 554-444 dopoldne in (061) 576-995 popoldne. TX-119

ŠABLONE ZA ZX spectrum vam olajšajo delo pri programiranju in pri igrah. Naročite jih lahko v kompletih 10 komadov. Cena kompleta je 440 din. Pošljem po povzetju. Ponudbe pošljite na naslov: Marjan Zorčič, Pod hribom 58 B, 61000 Ljubljana, telefon (061) 555-436. TX-118

IZŠLA JE KNJIGA, ki smo jo že dolgo pogrešali. Jure Špiler: BASIC (Uvod v računalništvo in programski jezik BASIC). Knjiga je dobrodošel domači vir znanja in napotkov za programiranje v zadnjem času tudi pri nas vedno bolj razširjenih računalnikov. Uporabljani so ustajeni slovenski strokovni izrazi. Knjiga je primerna za vse uporabnike hišnih računalnikov, kot so Apple, BBC, Commodore, Sinclair ZX 81, spectrum in drugi. Opisani so vsi stavki in funkcije, ki se uporabljajo v Basicu na hišnih računalnikih. Vsak pojem je pojasnjen tudi z ustreznim primerom. Najhitreje jo naročite po pošti. Cena 1000 din boste plačali ob prejemu pošiljke. Naročila pošljite na naslov: JURE ŠPILER, BASIC, p. p. 302, 61001 Ljubljana. TX-109

Oglase za tekočo številko sprejemamo do vključno ponedeljka. Cena enega oglasa: 200 din, ne glede na dolžino. Male oglase lahko bralci oddajo vsak dan med 7. in 19. uro, ob sobotah pa od 7. do 11. ure na blagajni malih oglasov, v Ljubljani, Titova 35 (črna stolpnica) ali v Šubičevi 1. Lahko pa jih oddajo po telefonu, v istem času, na številko 223-311 (ključna številka 061 za bralce izven Ljubljane). Male oglase lahko pošljete seveda tudi pisмено, in sicer na naslov: ČGP Delo Stik, Oglasno trženje, Titova 35.

naše vprašanje, strokovnjakov odgovor

bazi, vse drugo pa je neresnično. Taka vprašanja so seveda nesmiselna, saj odgovore nanje poznamo. Lahko pa postavimo vprašanje še drugače:

? – ima (janez, X).
? – ima (Y, knjiga).

To lahko tolmačimo kot »kaj ima janez«. Prolog bi odgovoril: »X=denar«. Drugo vprašanje pa prevedemo z »kdo ima knjigo?« in odgovor bi bil »Y=jana«. Tudi ti odgovori nas niso ravno hudo presenetili. Zaenkrat smo uporabili le deklaracijo dejstev in vprašanje. Oglejmo si torej še primer, kjer bomo uporabili tudi pravila:

ženska (jana).
ženska(mojca).
ima-rad (jana, kino).
ima-rad (mojca, hrana).
ima-rad (janez, X):
ženka (X), ima-rad (X, kino).

Najprej prevedemo ta programček v navaden jezik. Prve štiri vrstice preberemo kot »jana je ženska, mojca je ženska, jana ima rada kino, mojca ima rada

hrano«. Zadnji stavek pa je pravilo, ki pomeni »janez ima rada žensko, če ima ta rada kino«. Postavimo še vprašanja:

? – ima-rad (janez, mojca).

? – ima-rad (janez, X).

Vprašali smo »ali ima janez rad mojco?«. Prolog pogleda v bazo podatkov in ugotovi, da to dejstvo ne obstaja. Nato najde pravilo in ponovno pogleda bazo podatkov. Prvi pogoj »mojca je ženska« je izpolnjen, nikjer pa ni rečeno, da ima »mojca rada kino«, zato drugi pogoj ni izpolnjen in Prolog bi odgovoril »no«. Drugo vprašanje pa se glasi »koga ima janez rad?« Prolog ugotovi, da pogoje izpolnjuje Jana in odgovori »Y=jana«.

Če bi hoteli napisati program, ki bi postavljaj diagnozo bolezni in predpisoval zdravila, bi morali napisati bazo podatkov o različnih možnih simptomih, ki jih povzročijo različne bolezni, pravila o zdravljenju, ki so za določeno zdravljenje po-

trebna, pri kom drugem pa škodljiva in tako naprej (na medicino se ne spoznam preveč). Nato pa bi prologu postavili vprašanje in odgovoril bi v skladu s pravili, ki smo jih opisali v programu.

Verjetno ste opazili, da v programu nismo uporabili nobene angleške besede, ki so pogoste v drugih programskih jezikih (npr. if, for, then, else...). To so pravzaprav ukazi, ki določajo potek reševanja problema. V prologu je teh ukazov malo in uporabljajo se veliko manj kot v drugih programskih jezikih. Vzrok temu je v nedoločenosti poteka reševanja s strani programerja, ki na to nima velikega vpliva. Ugodna posledica tega pa je, da lahko program napišemo tako rekoč v slovenščini. Upoštevati moramo le sintaktična pravila prologa. Eno teh pravil je, da pišemo objekte z malo, spremenljivke pa z veliko začetnico. Zato naj vas ne moti, če so lastna imena pisana z malo.

Mirko Maher