

NOVO

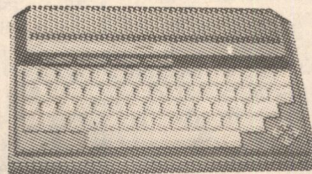
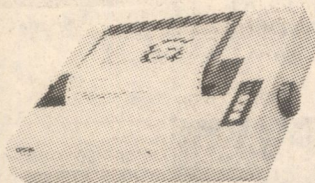
v kioskih

Prve dni junija izide
posebna izdaja Teleksa
v sodelovanju z Radiom Študent

MOJ MIKRO

Iz vsebine:

NOVO NA TUJEM TRGU: posebna poročevalca na dunajskem sejmu informacijske in biro tehnike



TESTI: color graph, test prvega domačega osebnega mikroračunalnika in ekskluzivni test mikroračunalnika prihodnosti (apple macintosh)

INTERVJU: slovenski politik o problemih uvoza mikroračunalnikov in računalniškega opismenjevanja

LISTINGI PROGRAMOV: hardwarski nasveti in programi za računalnike modelov spectrum, commodore 64 in BBC

PONUDBA V TUJINI: izčrpen pregled vseh najbolj razširjenih osebnih računalnikov in opreme, nasveti za nakup, tehnične novosti

PROGRAMSKA OPREMA: možnosti in predlogi za najbolj razširjene osebne računalnike

RAČUNALNIŠKA GRAFIKA: vrtoglav razvoj posebnega področja

OSEBNI ROBOTI: za osebnimi računalniki se že pojavljajo tudi osebni roboti

IN ŠE: drobne zanimivosti iz domovine in tujine, računalniške igrice, mikroprocesorji, predstavitev prve kasete Radia Študent itd.

MOJ MIKRO

Revija v barvah
in z bogato opremo
V vseh kioskih
že v prvi polovici junija



Telekov tečaj za začetnike – Tele

PROGRAM A

```

A1
A2
A3
A4
CALL B
A6
A7
A8
A9
A10
A11
END PROGRAM A
*
SUBROUT B
B1
B2
B3
CALL C
B5
B6
RETURN
*
SUBROUT C
C1
C2
C3
C4
C5
RETURN

```

čun sinusa, bi poklicali funkcijo SINUS in ji namesto x zapisali vrednost kota, katerega sinus želimo. Spremenljivke, ki se pojavljajo v funkciji, so lokalne in ne vplivajo na ostali program. Ideja je predstavljena na naslednji shemi:

```

A1
A2
A3
SINUS (45)
A5
A6
END PROGRAM A
*
FUNCTION SINUS (x)
  program, ki
  izračuna sin (x)
RETURN

```

Podobno velja tudi za procedure, ki enako spremljajo parametre od glavnega programa, ko jih ta pokliče, le da so njihove spremenljivke globalne. V nekaterih jezikih je mogoče procedure klicati rekurzivno, to pomeni, da lahko procedura kliče samo sebe. A o tem kdaj drugič.

Ob podprogramih naj omenimo še procedure in funkcije in predavanje parametrov. Funkcija je sklenjen del programa, ki ob vsakem klicu izvede kak postopek, katerega rezultat je vrednost, ki nas zanima. Vsakič lahko damo funkciji različne vhodne podatke (parametre), iz katerih potem izračuna rezultat. Če bi v programu denimo večkrat rabili funkcijo SINUS, je ne bi vsakič posebej programirali, ampak bi sprogramirali funkcijo SINUS (x) in vsakič, ko bi potrebovali izra-

Modularno programiranje predvideva klicanje manjših podprogramov za vsak del programa, ki naj izvede kako posebno funkcijo. Na ta način je program razdeljen na MODULE, od katerih ima vsak svojo nalogo; dobiti mora ustrezne vhodne podatke in predati naprej ustrezne izhodne rezultate.

Prihodnjič: podatkovni tipi in načini predstavljanja podatkov v računalniku.

POPRAVEK

V zadnji številki je tiskarski škrt zagodel nekaj napak. Ker gre za napake, ki so povsem spremenile smisel besedila, navajamo popravke.

1. Simbolični zapis polj v stavku zbirnega jezika bi moral biti takle:

LABELA MNEMONIK OPERANDI KOMENTAR
P2 ADD A,B seštej registra A in B

Namesto v zgornji obliki je bil stavek z imeni polj vred zapisan v stolpcu, kar je popolnoma nesmiselno.

2. Pri opisu stavka EQUATE bi se moral primer glasiti takole:

EPSILON EQU H'1F4A'

a je bilo namesto EQU zapisano WQU.

3. Okrajšava za DEFINE ADDRESS bi morali biti DANamesto AD.

4. V tabeli operacijskih kod instrukcij ne bi smelo biti večje med mnemoniki in operandi. Zadnja vrstica bi se morala glasiti:

Instrukcija koda (HEX)

MOV dir,A OF14 (16-bitni naslov operanda 1)

Tabela je bila zapisana v treh stolpcih namesto v dveh in zadnja vrstica razbita v dve vrstici.

5. V primeru prevedenega programa bi moralo v komentarju pri naslovu 0410 pisati:

direktni naslov operanda MIŠKA (glej 4. vrsto)

pisalo pa je »(glej 2. vrsto)«.

Za napake se opravičujemo.