Programski jezik C

Klavdija LUNAR

24.11.1997

Ponovitev od zadnjič

Pri vsakem programu moramo napisati komentarje.

```
/* DESC: (okrajsava za opis):
         izracuna priblizek konstante 2 pi do napake
   PARM: (okrajsava za parametri)
          napaka - napaka pri racunanju je najvec tako velika
   RESULT: (okrajsava za rezultat)
           2 pi
   PRE: (predpogoj, ki mora biti zagotovljen, da bo procedura
          pravilno delovala)
        napaka > 0
   POST: (popogoj, ki je zagotovljen, ce je predpogoj zagotovljen
          in procedura pravilno deluje)
        2 pi - napaka <= RESULT <= 2 pi + napaka
*/
```

Koraki, ki jih bomo naredili, da bomo naredili pravilen program:

1. $S_0 = 4; n = 2;$
2. $S_{i+1} = 2 * n * sqrt(2 - sqrt(4 - sqr(s/n)));$
3. $n = 2 * n;$
4. if ($|S_{i+1} - S_i| >$ napaka) then GOTO 2; (ko pride do te vrstice, se zopet vrne na 2. korak)
5. return S_n;

Proizvajalec je pritrdil, da je ta algoritem zadovoljiv.

Najprej smo sestavili testni program, da ugotovimo, če je program pravilno napisan:

```
#include<stdio.h>
#include<conio.h>

#include "Pribl_2_pi.h"
/*-----------------------------------------------------------------*/
void main(void) {
  double dva_pi, napaka;

  clrscr();
  printf("Vpisi napako> ");
  scanf("%lf", &napaka);            /* pričakuje nekaj, kar je float */
  dva_pi= Pribl_2_pi(napaka);
  printf("Priblizek 2pi je %lf +/- %lf. \n", dva_pi, napaka);
  (void) getch();
} /* main */

/*-----------------------------------------------------------------*/
#include<math.h>
#define sqr(x)  ((x)*(x))

double Pribl_2_pi(double napaka) {
  long n;                            /* iterativni stevec */
  double s,                          /* trenutni priblizek */
         s_old;                      /* prejsni priblizek */
  s= 4;                              /* zacetne vrednosti */
  n= 2;
  do {
    s_old= s;                        /* zapomnimo staro vrednost */
    s= sqrt(2-sqrt(4-sqr(s/n)));     /* nova vrednost */
    n= 2*n;
    s= s*n;
  }
  while (fabs(s-s_old) > napaka);    /* dokler ni manjsi od napake */
  return s;
} /* Pribl_2_pi */
```

Sedaj smo napisali testni program. Če želimo, da nam izračuna na 3 decimalke natančno popravimo.

Zakaj se nam pokaže napaka? Ko program poženemo, se nam pokaže:

```
Vpisi napako>
```

Pokaže se nam novi približek in pod njim stari približek.

Oznaka *abs* pomeni absolutno vrednost in nam da pozitiven rezultat. Oznaka *double fabs(double x)* pa nam rezultat vrne v double obliki.

Do sedaj smo si pogledali testni program in sedaj bomo to izpisali na datoteke. Napisali bomo datoteko našega programa za izračun približka 2π in ga poimenovali pogodba.

pogodba
testni program
koda

To pogodbo bomo nesli v drugo datoteko in jo preimenovali v prib2Pi.h.

.h je definicijski modul, ki mora imeti svojo pogodbo.

Kako smo prenašali iz testnega programa v svoje datoteke? Najprej smo besedilo, ki ga želimo prenesti, označili, potem smo ga shranili. Nato *Edit* in *Copy*, potem *File*, *New*, *Edit*, *Paste* in dobili smo, kar smo shranili, na novem področju. Potem pa *Save As* in shranimo pod neko ime. Mi smo našo pogodbo shranili pod *prib2Pi.h*.

Ta program ne loči velikih in malih črk.

Zanima nas razlika med „"" “ in „<>“.

V koničastih narekovajih (<>), so moduli, ki so na nekem sistemu (sistemski moduli). Prevajalnik vê, kje jih mora poiskati. Narekovaji ("") so naši moduli. Če želimo jih lahko dodamo v sistem.

Datoteki Pribnap.c in Prib2pi.h se nahajata na istem področju. Če hočemo priti en korak nazaj kliknemo na *../*, če pa želimo priti še bolj nazaj pa kliknemo večkrat na *../../../*. To je relativno postavljanje datotek.

Pogodbo smo shranili pod *Prib2pi.h*, kodo pa smo shranili pod *Prib2pi.c*. Zadnji je izvedbeni ali implementacijski modul.

Sedaj si bomo pogledali našo pogodbo:

```
double Pribl_2_pi (double napaka);
/* DESC: izracuna priblizek konstante 2pi do napake ( natancno )
    PARMS: napaka - napaka pri racunanju je najvec tako velika
    RESULT: 2pi
    PRE: napaka > 0
    POST: 2pi - napaka <= RESULT <= 2pi + napaka
*/
```

Ostala nam je še koda:

```
#include <math.h>
#define sqr(x)   ((x) * (x))

double Pribl_2_pi(double napaka) {
long n;                         /* iterativni stevec */
double s,                       /* trenutni priblizek */
       s_old;                   /* prejsni priblizek */
s= 4;                           /* zacetne vrednosti */
n= 2;
do {
  s_old= s;                           /* zapomnimo staro vrednost */
  s= sqrt( 2-sqrt( 4-sqr( s/n)));     /* nova vrednost */
  n= 2*n;
  s= s*n;
} while( fabs( s-s_old) > napaka );  /* dokler ni manjsi od napake */
return s;
} /* Pribl_2_pi */
```

Kako smo Prib2pi.c shranili? To smo naredili tako, da smo najprej tekst označili, potem pa *Edit* nato *Cut*. Potem smo klikniki na *Window*, nato *Cascade*. Potem gremo zopet na *Edit* nato *Paste* ter na koncu kliknemo na *File* nato *Save As*. In na koncu napišemo še ime pod katero bomo shranili to datoteko.

Sedaj vključimo še izvedbo in nato prevedemo. Vse deluje kot v testnem programu. Bistvo vsega je, da ločimo med izvajalcem in naročnikom.