



PROCES PISANJA PROGRAMOV

Mitja Nemec

Univerza v Ljubljani, Fakulteta za elektrotehniko,
Tržaška 25, 1000 Ljubljana, SLOVENIJA
e-mail: mitjan@fe.uni-lj.si



Program sestavljajo:

- Funkcije
 - ukazi
- Podatki
- Komentarji



Ukazi

- Zaporedje
- Organiziranje po blokih
- En ukaz – eno dejanje
 - Ena vrstica – eno dejanje



Pisanje programa - funkcije



```
// izracun moci
```

```
Pa = Ua*Ia*cos(fia)
```

```
Pb = Ub*Ib*cos(fib)
```

```
Sa = Ua*Ia
```

```
Sb = Ub*Ib
```

```
if (a == sin(sqrt(2*b+c)))
```



Funkcije

- Lastnosti funkcij
 - Vsaka funkcija naj ima samo en namen
 - Dolžina funkcije
 - Ime funkcije
- Vzroki za novo funkcijo:
 - Zmanjševanje kompleksnosti
 - Izogibanje podvojene kode
 - Skrivanje sekvenc

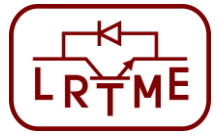


Funkcije

- Poimenovanje funkcij:
 - Ime naj pove kaj vse funkcija dela
 - Izogibaj se dvoumnih imen
 - V splošnem glagol/samostalnik
 - Uporabljaljaj predpis-e za splošne operacije (get, set,...)



Pisanje programa - funkcije



NarediIzracun () / IzracunajObresti ()

Pari

add/remove, increment/decrement

open/close, source/target

lock/unclock, min/max,

input/output, first/last

old/new, get/set



Funkcije

- **Struktura funkcije**
 - Preberi in obdelaj vhode
 - Algoritem
 - Nastavi izhode in zaključí
- **Defenzivno programiranje**
 - Preveri, če so vhodne vrednosti v pričakovani obliki/dosegu
 - Preveri, če so izhodne vrednosti v pričakovani obliki/dosegu



Pisanje programa - funkcije



```
float PID_regulator(float zeljena, float dejanska)
{
    if (zeljena > 10.0)
    {
        zeljena = 10.0;
    }
    // kompleksni izračuni
    izhod = rezultat kompleksnih izračunov
    if (izhod > 1.0)
    {
        izhod = 1.0;
    }
    if (izhod < 0.0)
    {
        izhod = 0.0;
    }
    return (izhod);
}
```



Podatki

- Osnovni podatkovni tipi
 - Cela števila: int, long
 - Realna števila: float, double
 - Znakovni tipi: char
 - Vrste: int[] , char[], string
- Pretvorbe med tipi
 - Naj bodo vedno eksplicitne
 - Preveri če je pretvorba sploh možna



Pisanje programa - podatki



```
int a;
unsigned int b;
float c;

a=c;           // to potrebuje komentar
a=(int)c;     // ne potrebuje komentarja
c=(float)a;

if(b < 32767)
{
    c = (int)b;
}
else
{
    c = 32767;
}
```



Podatki

- Deklaracija
 - Čim bliže mestu uporabe
 - Izogibajte se implicitni deklaraciji (Matlab, ...)
- Inicializacija
 - Vedno postavite podatek v znano stanje
- Doseg
 - Podatek naj bo “živ” čimmanj časa
 - Podatek naj bo dosegljiv čim manj funkcijam



Pisanje programa - podatki



```
float PID_regulator(float zeljena, float dejanska)
{
    int i;
    // malo morje kode, ki ne uporablja i-ja
    for (i=0; i<10; i++)
    {
        // koda ki uporablja i
        i = zeljena * temperatura;
    }
}
```



Pisanje programa - podatki



MATLAB

```
for i=0:1:100    % i ni bil nikjer prej deklariran

% problem v gnezdenih zankah
for i=0:1:100
    % malo kode
    % se malo kode
    % gnezdena zanka
    for i=0.0:0.1:1

        end
    end
end
```





Podatki

- Poimenovanje
 - Ime naj polno opiše kaj se nahaja v podatku
 - Izogibajte se generalnih imen (“i, temp, x”)
 - Tipično 10-16 znakov
 - Držite se samo enega sloga
- Uporaba
 - Vsak podatek se uporablja samo za en namen
 - Pazite se začasnih spremenljivk (“temp”)

Pisanje programa - podatki



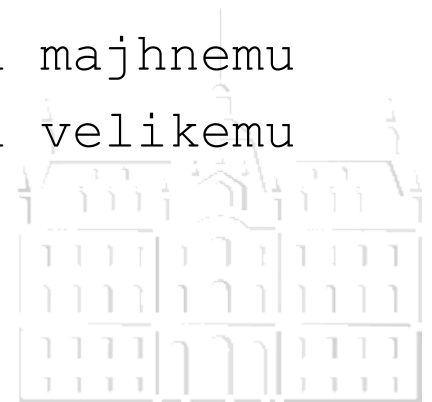
MATLAB

```
% problem v gnezdenih zankah
for koordinata_x=0:1:100
    % gnezdena zanka
    for koordinata_y=0.0:0.1:1

        end

    end
end
```

```
Motor_1_Faza_a_Tok      % od velikega proti majhnemu
Napetost_Faza_B_Motor % od majhnega proti velikemu
```



Abstraktni podatki

- Strukture, tabele, ...

```
// izracun moci
```

```
Faza_a.P = Faza_a.U*Faza_a.I * cos(Faza_a.fi);
```

```
Faza_b.P = Faza_b.U*Faza_b.I * cos(Faza_b.fi);
```

```
Faza_a.S = Faza_a.U * Faza_a.I;
```

```
Faza_b.S = Faza_a.U * Faza_a.I;
```



Oblikovanje kode

- Presledki vertikalni horizontalni
- Pravilno oblikovanje ukazov, ki so v več kot eni vrstici
- Združevanje kode po blokih
- Zamikanje
 - Vztrajno se držite samo enega sloga

```
if (condition)
{
    statement1;
    statement2;
}
```

```
if (condition)
{
    statement1;
    statement2;
}
```

```
if (condition){
    statement1;
    statement2;
}
```



- Koda se piše, da se jo lažje bere
- Enoten standard čez cel projekt
 - Če nimate svojega jih je veliko prosto dostopnih
- Orodja za urejanje sloga kode



Dokumentiranje

- Glava datoteke: avtor, verzija, opis
- Komentiranje podatkov
 - Vsak podatek posebej
 - Blok podatkov
- Komentiranje funkcije
- Komentiranje blokov



Tipi komentarjev

- Ponavljanje kode

```
a=b; // v a dam b
```

- Razlaga kode

```
c = sqrt(a*a + b*b) // dolzina
```

- Oznaka v programu

- // !! PAZI TOLE NI DOKONCANO !!



Tipi komentarjev

- **Povzetek kode**

```
// izracunam vsoto, nato izracunam vsoto  
kvadratov, potem pa vse skupaj poracunam v  
R.M.S.
```

- **Opis namena kode**

```
// tukaj izracunam R.M.S vrednost napetosti  
precej vrstic komplicirane kode
```



Razhroščevanje

- Lociranje napake
 - Sporočila prevajalnika
 - Optimirana koda
 - Nazadnje dodana koda
- Razumevanje napake
- Odpravljanje napake
- Shranjevanje kode
- Temeljito testiranje



- Programiraj za lažje branje in ne za lažje pisanje
- Programiraj za ljudi in ne za računalnik
- Uporabljaljaj standarde

