

Riadenie toku programu

Na riadenie toku programu v MATLBE existuje viacero príkazov (podobne ako v iných programovacích jazykoch):

- **for** vykonáva dopredu stanovený počet opakovaní určitej sekvencie príkazov
- **while** podobe ako **for** vykonáva opakovanie sekvencie programu pokiaľ je splnená logická podmienka. (Sú prípady keď dopredu neviem koľkokrát je nutné opakovať určitú časť programu a v týchto prípadoch použijeme príkaz **while**.)
- **if-elseif-else** vyhodnotí skupinu výrazov na základe logickej podmienky.

Okrem horeuvedených spôsobov riadenia programu MATLAB obsahuje aj ďalšie príkazy, ale tie budú uvedené neskôr. Každá sekvencia programu, kde sú použité horeuvedené príkazy na určitý výpočet musí byť nakoniec ukončená príkazom **end**.

Cyklus FOR

Cyklus **for** vykonáva dopredu zadaný počet opakovaní určitej časti (sekvencie) programu. Aby program vedel, ktorú časť má opakovať, je nutné zvolený blok ukončiť kľúčovým slovom **end**.

Od teraz si budem pamätať !!!

Na iteračný výpočet pri ktorom dopredu (pred začatím cyklu) poznám počet opakovaní použijem príkaz **for**.

Prototyp:

```
for premenna=vyras
...telo cyklu
end
```

Príklad použitia:

```
for i=1:10
disp(i^2);
end
```

Príklad 2.1

Zostavíme program, ktorý vypíše 10 krát jednoduchý text : *Ukazka pouzitia iteracneho cyklu FOR*.

Do nového M-súboru napíšeme:

```
for i=1:10
disp('Ukazka pouzitia iteracneho cyklu FOR');
end
```

Príklad 2.2

Druhý (trochu viac „zmysluplný“). Predstavte si, že máte mladšieho brata v prvom ročníku základnej školy. Váš brat má za úlohu vypočítať 10 krát súčet a rozdiel dvoch čísel. Brat vás požiada, aby ste mu danú domácu úlohu vyriešili a ako satisfakciu za danú službu vám ponúkne svoju zbierku plyšových strašidiel.

Nakoľko ovládáte iteračný cyklus **FOR** nemal by to byť problém.

```

for a=1:10
cislol=input('Zadaj cislo 1 ... ');
cislo2=input('Zadaj cislo 2 ... ');
disp('Sucet je');
disp(cislol+cislo2);
disp('Rozdiel je');
disp(cislol-cislo2);
end

```

Príklad 2.3

Tento príklad bude slúžiť na výpočet faktoriálu čísla n , ktoré zadá užívateľ .

```

clear all;close all;clc
n=input('Zadaj cislo n a ja vypocitam n! .... ');
fakt=1;
for(i=1:n)
fakt=fakt*i;
end
disp('Faktorial ma hodnotu :');
disp(fakt);

```

Cyklus WHILE

Cyklus **while** opakuje zvolenú časť programu pokiaľ je splnená nami zvolená podmienka. Podobne ako pri príkaze **for** je nutné časť programovej sekvencie, ktorú chceme opakovať ukončiť kľúčovým slovom **end**.

Od teraz si budem pamätať !!!

Na iteračný výpočet pri ktorom dopredu nepoznám počet opakovaní použijem príkaz **while**.

Prototyp:

```

while vyraz
...telo cyklu
end

```

Príklad použitia:

```

clear all;close all;clc
i=0;
suma=0;
while suma<100
suma=suma+i;
i=i+1;
end
disp('Posledné číslo z radu (1,2,..), kde SUMA<100 je :');
disp(i-2);

```

Ak ste si pozorne prezreli predchádzajúce riadky pravdepodobne vás napadlo, že v prototypy cyklu **while** je uvedený výraz, ktorý rozhodne či sa telo cyklu sa vykoná. Na rozdiel od cyklu **for**, kde telo cyklu bolo vykonané určitý (dopredu známy) počet krát. Na tvorbu daných výrazov sa používajú tzv. relaxačné operátory. Zoznam relaxačných operátorov je uvedený v nasledujúcej tabuľke:

Operátor	Význam
>	Väčší
<	Menší
<=	Menší nanajvýš rovný
>=	Väčší nanajvýš rovný
==	Rovná sa
~=	Nerovná sa

Príklad 2.4

Možno najjednoduchší, ale aj veľmi výstižný príklad použitia cyklu **while** je program, ktorý bude od užívateľa vyžadovať zadanie hesla do tej chvíle, kým zadané heslo nebude totožné s heslom 12345. Tento príklad bol zvolený z dôvodu veľmi názornej ilustrácie cyklu **while**, nakoľko my ako programátori nemôžeme vedieť koľkokrát užívateľ zadá chybné heslo, preto použitie cyklu **for** v tomto prípade nie je veľmi vhodné. Do nového M-súboru zapíšeme nasledujúce riadky:

```
n=input('Zadaj heslo :');
while n~=12345
disp('Zadal si zle HESLO skus znova');
n=input('Zadaj heslo :');
end
disp('Tvoje heslo je spravne');
```

Tvorba podmienok Príkazy if – else

MATLAB umožňuje robiť vetvenie programu na základe určitých nami zvolených podmienok. Na vetvenie Matlab ponúka tradičné príkazy, podobné ako v iných programovacích jazykoch, príkazy **if - else**. Takisto, aj pri tomto príkaze treba na koniec uviesť kľúčové slovo **end**.

Od teraz si budem pamätať !!!

Na vetvenie programu použijem príkaz **if – else**, prípadne **if – elseif – else** ak si to zadaný problém vyžaduje.

Prototyp:

```
if vyraz
...prikazy 1
else
...prikazy 2
end
```

Príklad použitia:

```
cislo=input('Zadaj cislo :');
if cislo>10
disp('Tvoje cislo je vacsie ako 10');
elseif cislo==10
disp('Tvoje cislo je prave 10');
else
disp('Tvoje cislo je mensie ako 10');
end
```

Príklad 2.5

Nasledujúci program bude slúžiť na výpočet koreňov kvadratickej rovnice. Kvadratická rovnica (ako všetci vieme) má riešenie v oblasti reálnych čísel vtedy (Poriadny matematik by povedal práve vtedy a len vtedy) ak je diskriminant väčší alebo rovný ako nula.

Vytvorte v Matlabe program, ktorý v prípade, že diskriminant bude záporný nám túto informáciu oznámi, v opačnom prípade nech vypočíta korene kvadratickej rovnice.

```
clear all;close all;clc
a=input('Zadaj hodnotu a ... ');
b=input('Zadaj hodnotu b ... ');
c=input('Zadaj hodnotu c ... ');
D=b^2-4*a*c;
if D<0
    disp('Diskriminant je zaporny, nemozem pocitat');
else
    x1=(-b+sqrt(D))/2/a;
    x2=(-b-sqrt(D))/(2*a);
    disp('Vysledok je :');
    disp(x1);
    disp(x2);
end
```

Poznámka:

Samozrejme MATLAB je schopný korene kvadratickej rovnice vypočítať automaticky pomocou príkazu **roots**. Nemá problém so záporným diskriminantom, keďže vie rátať aj s komplexnými číslami.

Uvedený príklad by mohol vyzerať takto:

```
a=input('Zadaj hodnotu a ... ');
b=input('Zadaj hodnotu b ... ');
c=input('Zadaj hodnotu c ... ');
disp(roots([a b c]));
```