

## 13. Matlab - riadenie behu programu

Blaho Michal · MATLAB/Comsol

21.08.2009



Pri písaní jednoduchých skriptov a funkcií je občas potrebné porovnať výrazy a na základe výsledku vykonať určitú časť programu. Inokedy zase potrebujeme opakovať určitú časť programu niekoľko krát po sebe. Na tieto a ďalšie iné veci sa využívajú riadiace príkazy, ktoré sú súčasťou každého skriptovacieho alebo programovacieho jazyka. Dnes si popíšeme ako tieto príkazy využívame v Matabe.

V programoch sa často stretnete s tým, že je potrebné vyhodnotiť určitú podmienku a na základe jej pravdivosti vykonať sériu príkazov. Matlab používa na túto činnosť príkaz **if**, ktorého syntax je nasledovná

```
if podmienka1
    príkazy1
elseif podmienka2
    príkazy2
else
    príkazy3
end
```

Príkaz **if** vyhodnocuje či je podmienka1 splnená a ak áno, vykoná príkazy1. Ak podmienka1 nie je splnená prejde sa do vetvy **elseif** a vyhodnotí sa pravdivosť podmienky2. Ak je podmienka2 splnená vykonajú sa príkazy2. Ak by ani jedna z podmienok nebola pravdivá vykonáva sa vetva **else**, teda príkazy3. Slovo **end** ukončuje príkaz vetvenia. Použitie príkazu si ukážeme na príklade

```
if a>b
    disp('Vacsia je premenna a')
elseif a<b
    disp('Vacsia je premenna b')
else
    disp('Premenne su rovnake')
end
```

Vetva **if** môže byť len jedna. Vetva **elseif** nemusí byť alebo sa môže využiť niekoľko krát, podľa toho koľko podmienok potrebujeme. Vetva **else** sa nemusí využiť, ale ak ju

chcem môže byť v príkaze iba jeden krát. Aby nedošlo ku chybám na tvorbu podmienok, ktoré pracujú s celými maticami sa využívajú príkazy **isequal**, **isempty**, **all** a **any**. Príkaz **if** sa dá jednoduchšie prepísať pomocou príkazu **switch** ak porovnávame premennú alebo výraz na nejakú hodnotu. Syntax príkazu **switch** je nasledovný

```
switch výraz
    case hodnota1
        príkazy1
    case hodnota2
        príkazy2
    otherwise
        príkazy3
end
```

Výraz za príkazom **switch** je vyhodnotený a porovnávaný na hodnoty za slovom **case**. Ak hodnota zodpovedá výrazu sú vykonané príkazy. Slovo **otherwise** nám vraví čo sa má vykonať keď nie je žiadna hodnota zodpovedajúca výrazu. Matlab končí prehľadávanie keď nájde hľadavú hodnotu preto sa na rozdiel od iných programovacích jazykov nepoužíva kľúčové slovo **break** pri používaní príkazu **switch**. Príklad použitia príkazu môžete vidieť tu

```
switch meno
    case 'Michal'
        disp('Meno je Michal')
    case 'Juraj'
        disp('Meno je Juraj')
    otherwise
        disp('Nezname meno')
end
```

Opakovanie príkazov v určitých blokoch je nevyhnutnou súčasťou každého písania skriptov alebo programov. Šetríme tým počet riadkov programu a zvyšujeme jeho čitateľnosť. Pre opakovanú činnosť programu môžeme využiť príkazy **while** a **for**. Príkaz **while** má nasledujúcu syntax

```
while podmienka
    príkazy
end
```

Cyklus **while** opakuje príkazy pokým je splnená podmienka. Znamená to teda, že cyklus sa nemusí vykonať ani raz. Pri nevhodných podmienkach môže nastať nekonečný cyklus. Ukončenie nekonečného cyklu sa dá urobiť pomocou klávesovej skratky **CTRL+C**. Príklad použitia cyklu môže byť nasledovný

```
i=0;
while i<10
    i=i+1;
end
```

Niekedy pomocou príkazu **while** nevieme zaručiť presný počet opakovaní, pretože podmienka môže byť splnená viac krát ako by sme chceli. Príkazy môžeme opakovať aj presný počet krokov pomocou príkazu **for**, ktorého syntax je nasledovná

```
for premenná = štart : prírastok : koniec
    príkazy
end
```

Cyklus ukladá aktuálnu hodnotu indexu do premennej, ktorá môže byť využitá neskôr v príkazoch. Krokujeme od štartovacej hodnoty po konečnú hodnotu so zadaným prírastkom. Prírastok je štandardne nastavený na jednotku. Ak by sme chceli s indexom klesať jednoducho dáme záporný prírastok. V takomto prípade musí byť hodnota štartu väčšia ako hodnota konca.

```
for i = 10 : -1 : 3
    disp(i)
end
```

Indexu môžeme priradiť aj vektor, ktorý sa bude prechádzať po prvkoch

```
A = [1 3 5 4 2]
for i = A
    disp(i)
end
```

Niekedy je potrebné vykonať určité príkazy, keď nastane chyba vo vašom programe alebo v jeho časti. V takomto prípade nám pomôže príkaz **try - catch**, ktorého syntax je nasledovná.

```
try
    príkazy1
catch
    príkazy2
end
```

V bloku **try** sa nachádzajú príkazy, v ktorých sa môže vyskytnúť chyba. Ak chyba nastane Matlab ihneď preskočí do bloku **catch**, kde vykonajú príkazy zodpovedajúce chybe. Poslednú chybu si vieme zistiť príkazom **lasterr**. Ak nastane chyba v časti **catch** Matlab ukončí vykonávanie príkazov.

Riadenie behu programu môžeme ovplyvniť príkazmi **continue**, **break** a **return**. Každý z príkazov má svoje špecifické vlastnosti. Príkaz **continue** preskakuje aktuálnu iteráciu cyklu **for** alebo **while**. Príkazy za **continue** sa už nevykonávajú a pokračuje sa ďalším krokom cyklu. Príkaz **break** slúži na ukončenie vykonávaného cyklu **for** alebo **while**. Vykonáva program tak pokračuje ďalej za koncom týchto cyklov. Ak je cyklov vnorených viac ukončuje sa len ten v ktorom sa príkaz **break** nachádza. Príkaz **return** ukončuje aktuálnu postupnosť príkazov a vracia riadenie tomu, kto vyvolal aktuálny program alebo skript, prípadne klávesnici. Týmto príkazom teda ukončíme činnosť skôr ako nastane jej bežný koniec o úroveň vyššie.

