

Zabudované dialógové okna v MATLABe

Na začiatku výučby MATLABu sme si ukázali najjednoduchšie spôsoby akými je možné zabezpečiť komunikáciu užívateľa s programom. Na túto komunikáciu sme používali dva príkazy: **input** – na získanie informácie od užívateľa a príkaz **disp** – na zobrazenie informácie programu výpisom na monitor.

Teraz si ukážeme niekoľko základných možností, ako túto komunikáciu vykonať využitím grafického užívateľského rozhrania MATLABu.

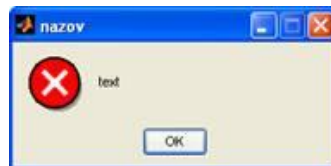
Podobne ako sme začali v minulom semestri hodinu s ukážkou programu, ktorý vypísal nami zadaný text na obrazovku, teraz vytvoríme jednoduché dialógové okno s nami zadaným textom. Na tento účel je v MATLABe príkaz **msgbox**, ktorého použitie je veľmi jednoduché:

```
msgbox('Moje prve okno v MATLABe')
```



Je možné použiť aj ďalšie parametre pri volaní funkcie **msgbox** a to názov okna, prípadne zobrazenú ikonu:

```
msgbox('text', 'nazov', 'error')
```



MATLAB ponúka tieto základné ikony:



Na získanie vstupných údajov od užívateľa MATLAB ponúka tiež elegantný postup a to využitím preddefinovaného dialógového okna, ktoré sa vytvára príkazom **inputdlg**, ktorého prototyp je nasledujúci:

```
hodnoty = inputdlg(zoznam poloziek, nazov okna, sirka riadku, preddefinovane hodnoty)
```

zoznam poloziek – zoznam parametrov, ktoré požadujeme od užívateľa aby ich zadal

nazov okna – podobne ako pri **msgboxe** je názov dialógového okna

sirka riadku – určuje koľko riadkov bude mať každá položka

preddefinovane hodnoty – hodnoty, ktoré sa zobrazia pri vytvorení dialógového okna

Ukážka použitia:

```
polozky={'koeficient A','koeficient B','koeficient C'};  
nazov='Zadaj parametre kvadratickej rovnice';  
preddefinovane={'1','2','5'};  
hodnoty = inputdlg(polozky,nazov,1,preddefinovane)
```



Po stlačení tlačidla OK sa do premennej (typu cell) uložia hodnoty jednotlivých riadkov. Pre úplnosť si ešte ukážeme spôsob ako je z nich možné získať jednotlivé hodnoty.

Predpokladajme, že sme použili predchádzajúci príklad. Ak chceme získať z premennej **hodnoty** jednotlivé číselné hodnoty, vykonáme to nasledovne:

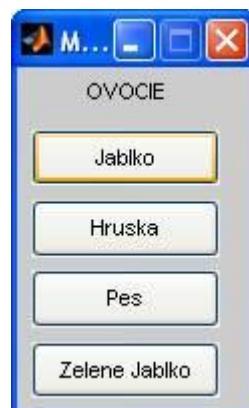
```
koefA=str2num(char(hodnoty(1)))  
koefB=str2num(char(hodnoty(2)))  
koefC=str2num(char(hodnoty(3)))
```

Medzi často využívané preddefinované dialógové okná, ktoré nám MATLAB ponúka patrí **menu**, ktoré slúži na získanie konkrétnej voľby z ponúkaného súboru volieb. Jej prototyp je nasledujúci:

```
zvolenahodnota = menu(nazov okna,moznost1,moznost2....)
```

Použitie **menu** si ukážeme na nasledujúcom príklade:

```
volba=menu('OVOCIE','Jablko','Hruska','Pes','Zelene Jablko')
```



Tvorba funkcií v MATLABe

Do tejto chvíle sme si pri programovaní vystačili bez znalostí o tvorbe funkcií. V prvom rade je nutné si povedať, čo funkcia je. Funkciu možno chápať ako časť kódu, ktorá akceptuje jeden prípadne viac parametrov, vykoná predpísaný algoritmus a vracia do hlavného programu parametre.

Teraz si povedzme niekoľko dôvodov prečo je výhodne/nutné tvoriť funkcie v MATLABe:

- určité časti kódu, ktoré si pripravíme a potrebujeme ich často používať, môže zapísať do funkcie. Takto vytvorenú funkciu môžeme využívať vo viacerých programoch. Navyše ak takúto funkciu raz vyladíme a presvedčíme sa o tom, že pracuje správne ušetríme veľa času, nakoľko ju nebudeme musieť pri každom použití opäť ladiť
- využitie funkcií vnesie do programu väčší poriadok, nakoľko umožňuje oddelenie konkrétnych častí kódu, od hlavného programu. Takto štruktúrovaný program má veľkú výhodu aj v tom, že ak chceme vykonať malú zmenu určitej časti kódu nie je potrebné meniť celý program, ale len konkrétnu časť funkcie
- azda najvýznamnejší dôvod, prečo sa venovať tvorbe funkcií je skutočnosť, že samotný MATLAB vyžaduje zadanie nášho problému vo forme funkcie, pri riešení azda všetkých numerických výpočtov

Ako vytvoriť v MATLABe funkciu?

Funkcia v MATLABe je v skutočnosti bežný M-file, s tým rozdielom, že musí obsahovať niektoré predpísané prvky (všetky M-file, ktoré sme doteraz vytvárali boli v skutočnosti skriptami, v ktorých sa daný kód vykonával postupne od prvého riadku po koniec M-file).

Všeobecný zápis funkcie v syntaxe MATLABu je nasledujúci:

```
function [von1,von2,von3...]=nazovfunkcie(vstup1,vstup2,vstup3...)
```

takto zapísaná funkcia akceptuje vstupy (vstup1,vstup2,vstup3...) a po vykonaní svojho kódu vracia výstupné parametre [von1,von2,von3...].

Príklad 1

Vytvorte funkciu na výpočet koreňov kvadratickej rovnice. Kvadratická rovnica je jednoznačne zadaná tromi koeficientami a, b, c . Každá kvadratická rovnica má dva korene (okrem prípadu ak má rovnica jeden dvojnásobný koreň). Z tohto vyplýva že naša funkcia by mala akceptovať tri parametre (koeficienty a, b, c) a do hlavného programu bude vraciať dva parametre (korene x_1 a x_2).

Do nového M-file zapíšeme nasledujúce riadky:

```
function [x1,x2]=kvadraticka(a,b,c)
D=b^2-4*a*c;
if D>0
    x1=(-b+sqrt(D))/(2*a);
    x2=(-b-sqrt(D))/(2*a);
end
```

Takto vytvorenú funkciu môžeme teraz vyskúšať. V podstate existujú tri základné spôsoby ako to vykonať:

- nami vytvorenú funkciu môžeme použiť priamo z príkazového riadku:

```
>>[koren1,koren2]=kvadraticka(-5,1,5)
```
- prípadne môžeme vytvoriť nový M-file, do ktorého zapíšeme vyššie uvedený riadok kódu
- tretí spôsob (v našom prípade pravdepodobne najjednoduchší) spočíva v spojení „hlavného programu“ (riadok [koren1,koren2]=kvadraticka(-5,1,5)) spolu s funkciou na výpočet koreňov kvadratickej rovnice do jedného M-file. Práve tento postup si teraz bližšie opíšeme

Daný kód by vyzeral nasledujúco:

```
[koren1,koren2]=kvadraticka(-5,1,5)
```

```
function [x1,x2]=kvadraticka (a,b,c)
D=b^2-4*a*c;
if D>0
x1=(-b+sqrt(D))/(2*a);
x2=(-b-sqrt(D))/(2*a);
end
```

Ak by sme takto vytvorený program spustili, dostali by sme od MATLABu chybové hlásenie, ktoré by nám oznamovalo, že nie je možné spúšťať skript, ktorý obsahuje definície funkcií: **Function definitions are not permitted at the prompt or in scripts.** alebo **Function definitions are not permitted in this context.**

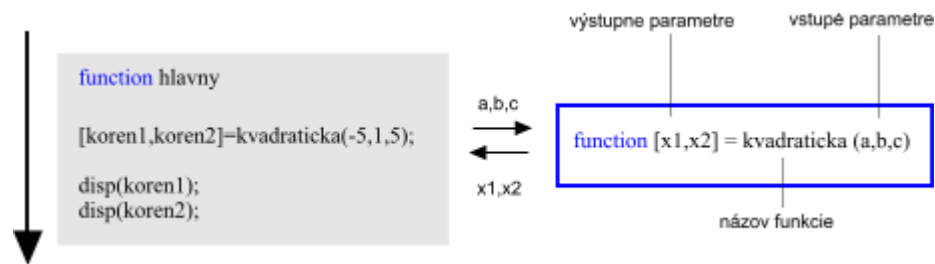
Preto je nutné upraviť náš „hlavný program“(ktorý je vlastne skript) na akúsi hlavnú funkciu programu. Je to v podstate veľmi jednoduché:

```
function hlavny
[koren1,koren2]=kvadraticka (-5,1,5);
disp(koren1);
disp(koren2);

function [x1,x2]=kvadraticka (a,b,c)
D=b^2-4*a*c;
if D>0
    x1=(-b+sqrt(D))/(2*a);
    x2=(-b-sqrt(D))/(2*a);
end
```

Poznámka: názov M-file by mal byť totožný s názvom prvej funkcie.

Takto sme vytvoril funkciu **hlavny**, ktorá neakceptuje žiadne parametre a takisto ani žiaden parameter nevracia. Tento veľmi jednoduchý program je možné schematicky znázorniť nasledujúco:



Hlavný program má v podstate len tri riadky. V prvom riadku je volaná funkcia **kvadraticka** s tromi konštantnými parametrami a, b, c , ktorá vráti späť do hlavného programu premenné x_1 a x_2 (korene kvadratickej rovnice). Zvyšné dva riadky sa starajú o vypísanie hodnôt vypočítaných koreňov.

Dôležité upozornenie: ako vidíte M-file môže obsahovať viacero funkcií. Prvá funkcia v súbore, tzv. **hlavná funkcia**, je „viditeľná“ pre funkcie v iných súboroch alebo môže byť volaná z príkazového riadku, čo sme si už ukázali. Ďalšie funkcie v M-file sú tzv. **lokálne funkcie**. Lokálne funkcie sú „viditeľné“ len pre funkcie v tom istom M-file. Lokálne funkcie môžu byť zapísané v ľubovoľnom poradí s tým, že hlavná funkcia je vždy prvá. Lokálne funkcie v aktuálnom M-file sú nadradené funkciám v iných súboroch, to znamená, že ak voláte funkciu v rámci aktuálneho M-file, MATLAB skontroluje, či je volaná funkcia lokálnou funkciou predtým než hľadá iné hlavné funkcie. Tento prístup dovoľuje vytvoriť alternatívne verzie určitej funkcie pri zachovaní originálnej v inom M-file. Všetky funkcie, vrátane lokálnych, majú vytvorené vlastné pracovné priestory, ktoré sú oddelené od základného pracovného priestoru. Lokálne funkcie teda nemôžu pristupovať k premenným používanými inými funkciami, pokiaľ nie sú poslané ako argumenty. Naproti tomu, tzv. **vnorené funkcie** (funkcie nachádzajúce sa v inej funkcii) môžu pristupovať k premenným používanými funkciami, ktoré ich obsahujú. Ak hociktorá funkcia v M-file obsahuje vnorenú funkciu, všetky funkcie v M-file musia byť ukončené príkazom **end** indikujúcim koniec funkcie. V opačnom prípade je ukončenie funkcie voliteľné.

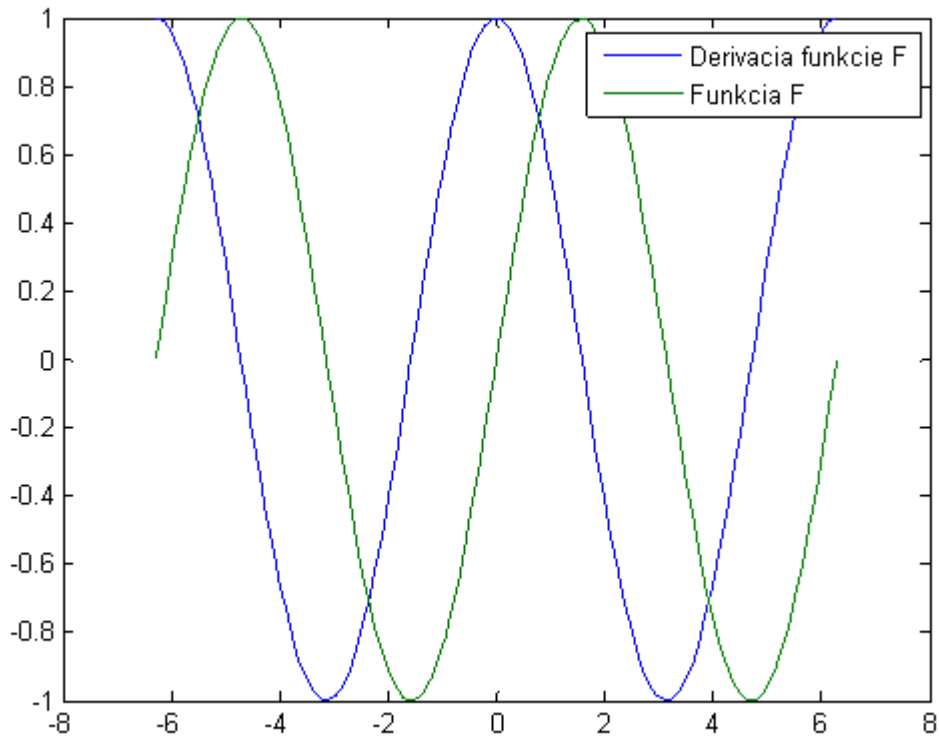
Príklad 2

Zostrojte program na výpočet prvej derivácie funkcie.

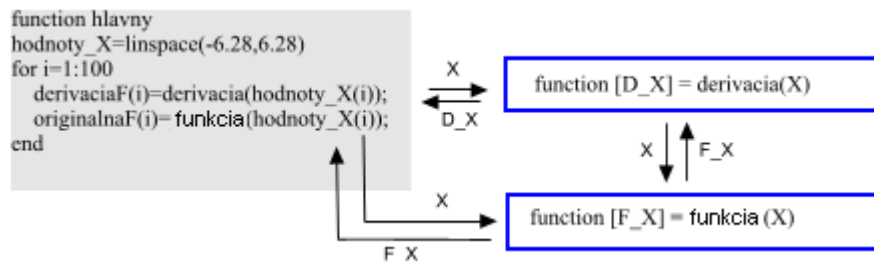
```
function hlavny
hodnoty_X=linspace(-6.28,6.28,100);
for i=1:100
    derivaciaF(i)=derivacia(hodnoty_X(i));
    originalnaF(i)=funkcia(hodnoty_X(i));
end
plot(hodnoty_X,derivaciaF,hodnoty_X,originalnaF)
legend('Derivacia funkcie F','Funkcia F');

function [D_X]=derivacia(X)
delta=1e-6;
D_X=(-funkcia(X)+funkcia(X+delta))/delta;

function [F_X]=funkcia(X)
F_X=sin(X);
```



V tomto príklade máme dve funkcie a to funkciu **funkcia**, v ktorej sa nachádza závislosť $F(X)$, ktorej prvú deriváciu chceme vypočítať. Druhá funkcia sa stará o samotný výpočet prvej derivácie, pričom k tomuto výpočtu využíva funkciu **funkcia**. Tento program je možné schematicky znázorniť nasledujúco:



Uvedený príklad má hneď niekoľko výhod:

- ak chceme získať prvú deriváciu inej funkcie ako $\sin(x)$, bez problémov prepíšeme daný riadok vo funkcii **funkcia**
- v prípade, ak chceme použiť iný zápis, ktorý aproximuje prvú deriváciu stačí ak prepíšeme funkciu **derivacia**