

TUTORIÁL2 - RIEŠENIE ÚLOH V JAZYKU MATLAB S VYUŽITÍM VLASTNÝCH SKRIPTOV A FUNKCIÍ

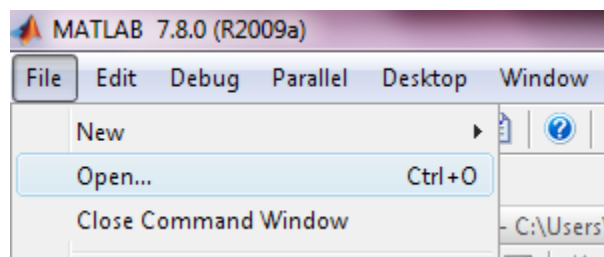
NÁPLŇ

1. PRÁCA S M-FILE V PROGRAMOVOM PROSTREDÍ MATLAB
2. TVORBA SPUSTITELNÝCH SÚBOROV PRÍKAZMI - SKRIPTY
3. TVORBA POUŽÍVATEĽSKÝCH FUNKCIÍ V JAZYKU MATLAB
4. PRÍKLADY NA SAMOSTATNÉ RIEŠENIE

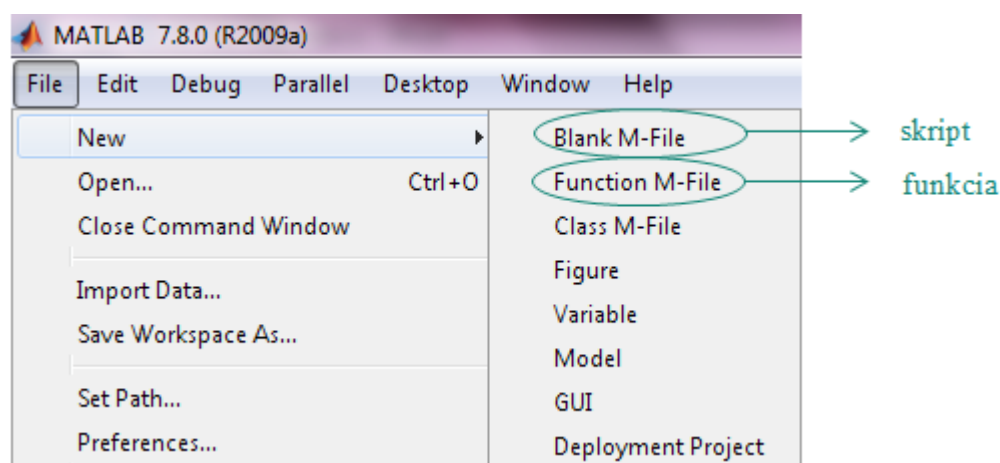
T2 - 1. PRÁCA S M-FILE V PROGRAMOVOM PROSTREDÍ MATLAB

M - file / M - súbor sú textové súbory s koncovkou „.m“, ktoré slúžia na ukladanie postupností príkazov - *skripty*, alebo na ukladanie užívateľských funkcií - *funkcie*. Využívajú sa predovšetkým vtedy, keď je potrebné zadať väčšie množstvo príkazov, čo by v príkazovom okne bolo neprehľadné, zdĺhavé a zložité meniť. Programovacie prostredie Matlab v sebe zahŕňa vlastný textový editor M - súborov, avšak môže sa použiť aj iný textový editor. Takto je k nim zabezpečený priamy prístup a môžu sa kedykoľvek upravovať nezávisle od hlavného programu.

Tento M - editor sa spustí v novom okne po otvorení M- súboru



Alebo po vytvorení nového M- súboru



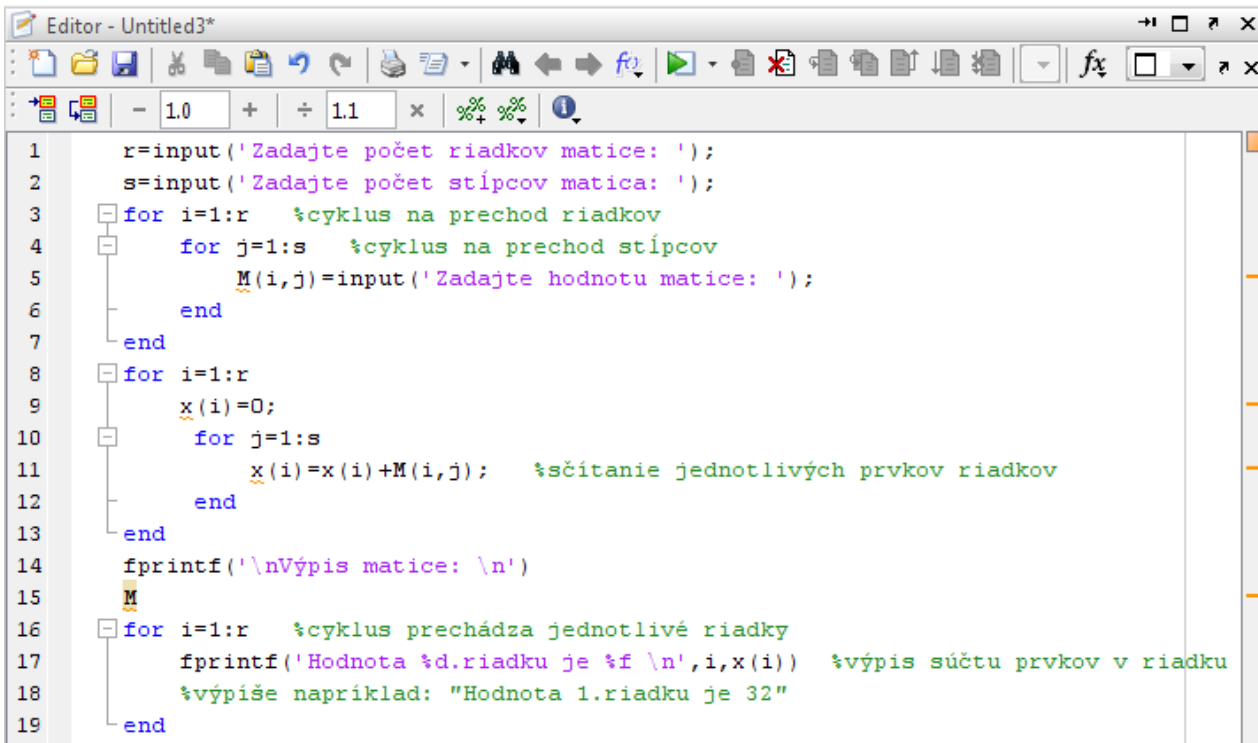
Ako už bolo spomenuté, existujú dva typy M- súborov a to súbor so skriptom a súbor s funkciou, v nasledujúcej tabuľke je ich porovnanie.

Skript	Funkcia
Nezadávajú sa vstupné argumenty a ani nevypisuje výstupné	Môže obsahovať vstupné a vracať výstupné premenné
Pracuje s premennými vo Workspace	Vnútorne premenné sú lokálne pre funkciu, Workspace „nevidí“
Vhodné pre automatizáciu volania tých istých príkazov viac krát	Vhodné pre rozšírenie funkčnej základne matlabovských aplikácií

T2 - 2. TVORBA SPUSTITEL'NÝCH SÚBOROV PRÍKAZMI - SKRIPTY

Skripty vid. Tutoriál1-4.Riadiace štruktúry jazyka Matlab

Riešený príklad 1 - Vytvorte postupnosť príkazov - skript, v ktorom vyzvete užívateľa na zadanie rozmerov matice a jej následné naplnenie. Vypíšte súčet riadkov matice.



```
1  r=input('Zadajte počet riadkov matice: ');
2  s=input('Zadajte počet stĺpcov matice: ');
3  for i=1:r %cyklus na prechod riadkov
4      for j=1:s %cyklus na prechod stĺpcov
5          M(i,j)=input('Zadajte hodnotu matice: ');
6      end
7  end
8  for i=1:r
9      x(i)=0;
10     for j=1:s
11         x(i)=x(i)+M(i,j); %sčítanie jednotlivých prvkov riadkov
12     end
13 end
14 fprintf('\nVýpis matice: \n')
15 M
16 for i=1:r %cyklus prechádza jednotlivé riadky
17     fprintf('Hodnota %d.riadku je %f \n',i,x(i)) %výpis súčtu prvkov v riadku
18     %výpíše napríklad: "Hodnota 1.riadku je 32"
19 end
```

T2 - 3. TVORBA VLASTNÝCH FUNKCIÍ V JAZYKU MATLAB

V Matlabe existujú vstavané funkcie ako napríklad *sin*, *sqrt* či *abs*, ktoré bežne používame. Ich názov hovorí na čo slúžia, ale užívateľ nemusí vedieť nič o výpočte vykonávanom sa vo vnútri funkcie. Užívateľ jednoducho zadá vstupný parameter a následne mu je vrátený výstupný parameter. Výhodou využívania funkcií je „skrytie“ algoritmu vypočítavajúceho danú funkciu pod názov funkcie. Nie však všetky funkcie, ktoré používateľ potrebuje sú vopred naprogramované, preto existujú používateľské funkcie.

Funkcie sú M-fily s presne definovanou štruktúrou, ktoré obsahujú postupnosti príkazov a zároveň vstupné a výstupné premenné.

```
function [ output_args ] = Untitled( input_args )  
%UNTITLED Summary of this function goes here  
% Detailed explanation goes here  
  
end
```

Charakteristické znaky funkcie :

1. Definícia funkcie - názov funkcie, vstupné a výstupné parametre
2. Prvý riadok nápovedy - stručný popis funkcie, je prehľadávaný funkciou `lookfor`
3. Nápoveda - popis funkcie
4. Vlastný kód

Poznámka :

- Prvý riadok nápovedy a nápoveda sú nepovinné časti.
- **Názov funkcie sa musí zhodovať s názvom súboru bez príponu .m**

Vstupné parametre môžu byť pri každom spustení funkcie iné. Výhodou funkcií je použitie jedného postupu - algoritmu pre rôzne hodnoty vstupných premenných. V takomto prípade nie vhodné použitie skriptov, keďže by sa hodnoty museli stále prepisovať.

Všetky premenné vo funkcii sú lokálne, to znamená, že existujú len počas spustenia funkcie. Vo funkcii nie je možné použiť iné premenné ako tie, ktoré vložíme do vstupných parametrov, alebo tie, ktoré vo funkcii vytvoríme. Taktiež platí, že všetky premenné po skončení funkcie zaniknú. Vstupné parametre sú podávané hodnotou, takže ak zmeníme ich hodnotu v tele funkcie nezmení sa tým hodnota premennej, s ktorou bola funkcia volaná.

Riešený príklad 2 - Vytvorte funkciu, ktorej vstupným parametrom bude strana kocky a výstupným objem kocky. Do tela funkcie napíšte kód, ktorým zrealizujete výpočet objemu kocky.

```
1 function [V] = objem_kocky(a)
2 % FUNKCIA výpočet objemu kocky
3 % Funkcia vypočíta objem kocky na základe
4 % vstupného parametra - hrany kocky
5 V=a^3;
6 % do výstupnej premennej sa vloží hodnota vstupnej premennej umocnená na 3
7 end
```

V príkazovom riadku si do premennej X vložíme hodnotu objemu kocky so stranou 12 pomocou vopred uloženej funkcie „objem_kocky“

```
>> X=objem_kocky(12)

X =

    1728

fx >> |
```

Riešený príklad 3 - Vytvorte funkciu, ktorej výstupná premenná bude obsahovať n - faktoriál čísla, ktorý bude vstupným parametrom.

```
1 function [nfaktorial] = faktorial(x)
2 %FUNKCIA výpočet n - faktoriálu
3 % vypočíta faktoriál vstupného argumentu x
4 nfaktorial=1;
5 for i=1:x
6     nfaktorial=nfaktorial*i;
7 end
8
9 end
```

V príkazovom riadku si do premennej N vložíme hodnotu faktoriálu čísla 5 (5!) pomocou vopred uloženej funkcie „faktorial“.

```
Command Window
>> N=faktorial(5)

N =

    120
```

Riešený príklad 4 - Vytvorte funkciu na výpočet obsahu trojuholníka. V hlavnom programe vyzvite užívateľa na zadanie strany a výšky na túto stranu.

```
% HLAVNÝ PROGRAM
b=input('Zadajte stranu trojuholníka: ');
vb=input('Zadajte výšku na zadanú stranu: ');
St=obsah_trojuholnika(b,vb);
fprintf('Obsah trojuholníka je : %f',St);
```

3. Obsah premennej S sa vloží do premennej St

```
% FUNKCIA
function [ S ] = obsah_trojuholnika( a,va )
S=a*va/2;
end
```

1. Zavolanie funkcie „obsah_trojuholnika“ so vstupnými premennými b a vb

2. Vykonanie tela funkcie, kde sa namiesto premenných a a va vložila premenné b a vb

Riešený príklad 5 - Vytvorte funkciu, ktorá určí hodnotu zlomku $\frac{(a^2-3a-2)}{(a-1)^2(a+2)}$. Túto funkciu použite v hlavnom programe, pomocou ktorého budete môcť vypočítať hodnotu funkcie pre a zadané z klávesnice. Skontrolujte či pre danú hodnotu má rovnica riešenie. Ak nie výsledok bude 0. Výsledok vytlačíte na obrazovku.

```
function [ H ] = zlomok( a )
if (a==1 || a==-2)
    H=0;
else
    H=(a^2-3*a-2)/((a-1)^2*(a+2));
end
end
```

Riešený príklad 6 - Vytvorte funkciu, ktorá bude využívať Pytagorovu vetu pre výpočet chýbajúcej strany. V hlavnom programe (skripte) ponúknite možnosť výberu chýbajúcej strany - prepona/odvesna a zohľadnite to aj vo funkcii. Dĺžky zvyšných dvoch strán zadá užívateľ na vstupe.

Funkcia:

```
function [ x ] = pytagoras( o,a,b )
    switch o
        case {1}
            x=sqrt(a*a+b*b);
        case {2}
            if a>b
                x=sqrt(a*a-b*b);
            else x=sqrt(b*b-a*a);
            end
        end
    end
end
```

Hlavný program:

```
o=input('Pre výpočet prepony zadaj "1" \nPre výpočet odvesny zadaj "2"\n');
while (o>2 || o<1)
    o=input('Pre výpočet prepony zadaj "1" \nPre výpočet odvesny zadaj "2"\n');
end
a=input('Zadajte hodnotu prvej strany: ');
b=input('Zadajte hodnotu druhej strany: ');
c=pytagoras(o,a,b)
```

Riešený príklad 7 - Vytvorte funkciu, ktorá bude vracať súčet vektorov, súčin vektorov po prvkoch a súčet absolútnych hodnôt jednotlivých prvkov vektorov. Vektory budú vstupnými argumentmi funkcie.

```
function [ sucin, sucet, SAH ] = vektory( v1,v2 )
% sucin - výstupný vektor, ktorý predstavuje súčin vstupných vektorov
% sucet - výstupný vektor, ktorý predstavuje súčet vstupných vektorov
% SAH - výstupná hodnota - súčet absolútnych hodnôt jednotlivých prvkov vstupných vektorov
if length(v1)~=length(v2)
    error('vektory nemajú rovnakú dĺžku')
else
    sucet=v1+v2;
    SAH=0;
    for i=1:length(v1)
        SAH=SAH+abs(v1(i))+abs(v2(i));
        sucin(i)=v1(i)*v2(i);
    end
end
end
```

Riešený príklad 8 - Vytvorte funkciu ktorej vstupnými parametrami budú dve ľubovoľné matice. Funkcia porovná ich rozmery, v prípade nerovnosti rozmerov vypíše chybu, v opačnom prípade bude hodnotiť prvky matíc na rovnakých pozíciách na základe ich párnosti. Výsledky zapíše do výslednej matice, ktorá bude výstupným argumentom, a to tak, že do nej zapíše počet párných prvkov na danej pozícii(0,1 alebo 2).

```
function [ V ] = matice( A,B )
if size(A)~=size(B)
    error('Zadané matice nemajú rovnaké rozmery')
else
    a=size(A);
    for i=1:a(1)    %cyklus na prehliadanie riadkov matíc
        for j=1:a(2)    %cyklus na prehliadanie stĺpcov matíc
            if rem(A(i,j),2)==0 %overenie párnosti prvku matice A na pozícii (i,j)
                if rem(B(i,j),2)==0 %overenie párnosti prvku matice B na pozícii (i,j)
                    V(i,j)=2;    %prípád, že obidve podmienky sú vyhodnotené ako pravdivé
                else
                    V(i,j)=1;    %prvá podmienka platí a druhá nie
                end
            else
                if rem(B(i,j),2)==0 %overenie párnosti prvku matice B na pozícii (i,j)
                    %v prípade, že prvok matice A na danej pozícii nie je párný
                    V(i,j)=1;    %prvok matice A nepárny a prvok matice B párný
                else
                    V(i,j)=0;    %prípád, že obidve podmienky sú vyhodnotené ako nepravdivé
                end
            end
        end
    end
end
end
```


TUTORIÁL2 - 4. PRÍKLADY NA SAMOTATNÉ RIEŠENIE

1. Vytvorte funkciu, ktorej výstupný parameter bude štvrtá mocnina vstupného parametra.
2. Vytvorte funkciu, ktorá vypočíta obsah trojuholníka so vstupnými premennými základňou a výškou.
3. Vytvorte funkciu, ktorá vráti skalárny súčin zadaných vektorov.
4. Máme funkciu výnosov $f(b)=895b+64-2^b$. Vytvorte funkciu v Matlabe, ktorá vytvorí vektor hodnôt funkcie výnosov na základe vstupného vektora a vyhledá bod, v ktorom sú výnosy najvyššie. Použite funkciu v hlavnom programe, v ktorom si vyžiadate ručný vstup hodnôt, ktoré budú vstupnými hodnotami pre funkciu výnosov. Na vstup dajte kontrolu, či ste zadali vektor. Ak nie vytvorte cyklus, aby sa funkcia vykonala až po správnom zadaní hodnôt. Vo výsledku vypíšte hodnoty funkcie a aj najvyššiu hodnotu a bod.
5. Vytvorte funkciu, ktorá vynásobí 2 vektory medzi sebou po prvkoch a zo súčinu vypíše všetky čísla deliteľné 3 alebo 4. Vektory zadajte z klávesnice a výsledky vypíšte.
6. Vytvorte funkciu, ktorej vstupné parametre budú dva vektory rovnakej dĺžky. Funkcia má umocniť každý i -ty prvok 1.vektora i -tým prvkom 2.vektora.
7. Vstupnými parametrami funkcie nech sú dve matice. Vytvorte funkciu na porovnanie týchto matíc. Funkcia zistí či sú matice rovnakých rozmerov. V prípade že nie sú výsledkom bude 0. V opačnom prípade bude funkcia postupne porovnávať prvky matíc na rovnakých pozíciách. Výsledky porovnania zapíše do výslednej matice nasledovne. Ak je prvok prvej matice väčší ako prvok druhej matice do výslednej matice sa zapíše 1, ak je prvok prvej matice menší ako prvok druhej matice do výslednej matice sa zapíše 2. V prípade, že sa prvky rovnajú zapíše sa 0.