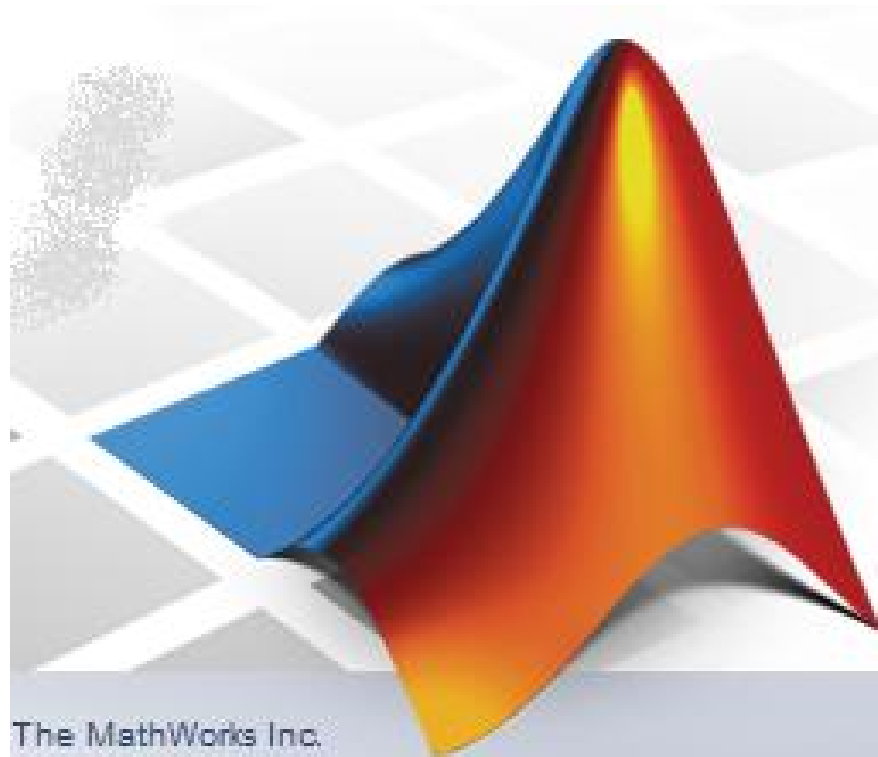


MATLAB (1) - úvod do programovania vedeckých problémov



Program predmetu:

1. týždeň: úvod, základné info o Matlabe, pracovné prostredie Matlabu, interaktívny režim, prvé info o písaní skriptov
2. týždeň: základné operácie s maticami, import a export dát, základné grafické zobrazovanie (grafy a mapy)
3. týždeň: práca s reťazcami, práca so súbormi
4. týždeň: pokročilejšia grafika - popis grafov a máp, 2D grafy
5. týždeň: príkazy, stavba programov, M-súborov
6. týždeň: funkcie – zabudované v Matlabe, tvorba vlastných funkcií
7. týždeň: tvorba vlastných aplikácií, práca s GUI (Graphical User Interface)
8. týždeň: tvorba vlastných aplikácií, nástroj GUIDE

pozn.: zmeny vyhradené



Obsah (8. prednáška)

- GUIDE – nástroj na tvorbu vlastných aplikácií
- Štrukturované premenné
- GUI ‘objects handle’

Základné spôsoby tvorby GUI v Matlabe:

- pomocou nástroja GUIDE

Interaktívna tvorba grafického rozhrania (vyvoláva sa príkazom *guide*), kde sa pomocou myši dajú jednoducho umiestňovať jednotlivé ovládacie prvky. Ich parametre a rozmiestnenie je uložené v binárnom *.fig súbore, zároveň je generovaný aj kód s funkciami, pre aktívne prvky, ktorým treba doplniť jadrá funkcií

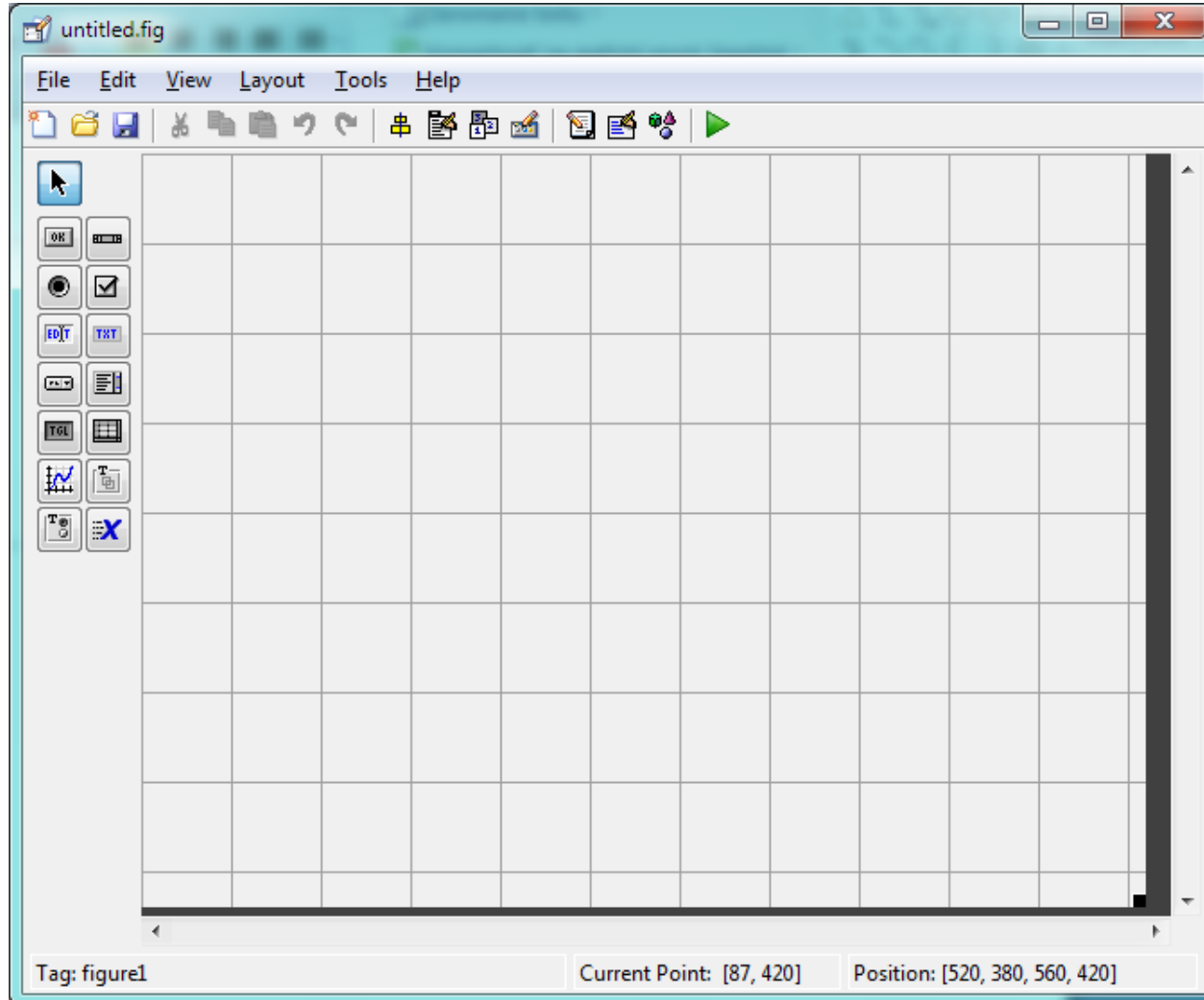
- programovaním

GUI je tvorené priamo písaním príkazov do m-súboru čo je samozrejme viac časovo náročné. Takto vytvorené rozhranie je však spravidla jednoduchšie na pochopenie 'cudzieho' kódu, kompaktnejšie a recyklovateľné

Pozn.: [link](#) na zaujímavú diskusiu o kladoch a záporoch oboch spôsobov

MATLAB (1) - úvod do programovania vedeckých problémov, LS 2017

```
>> guide
```



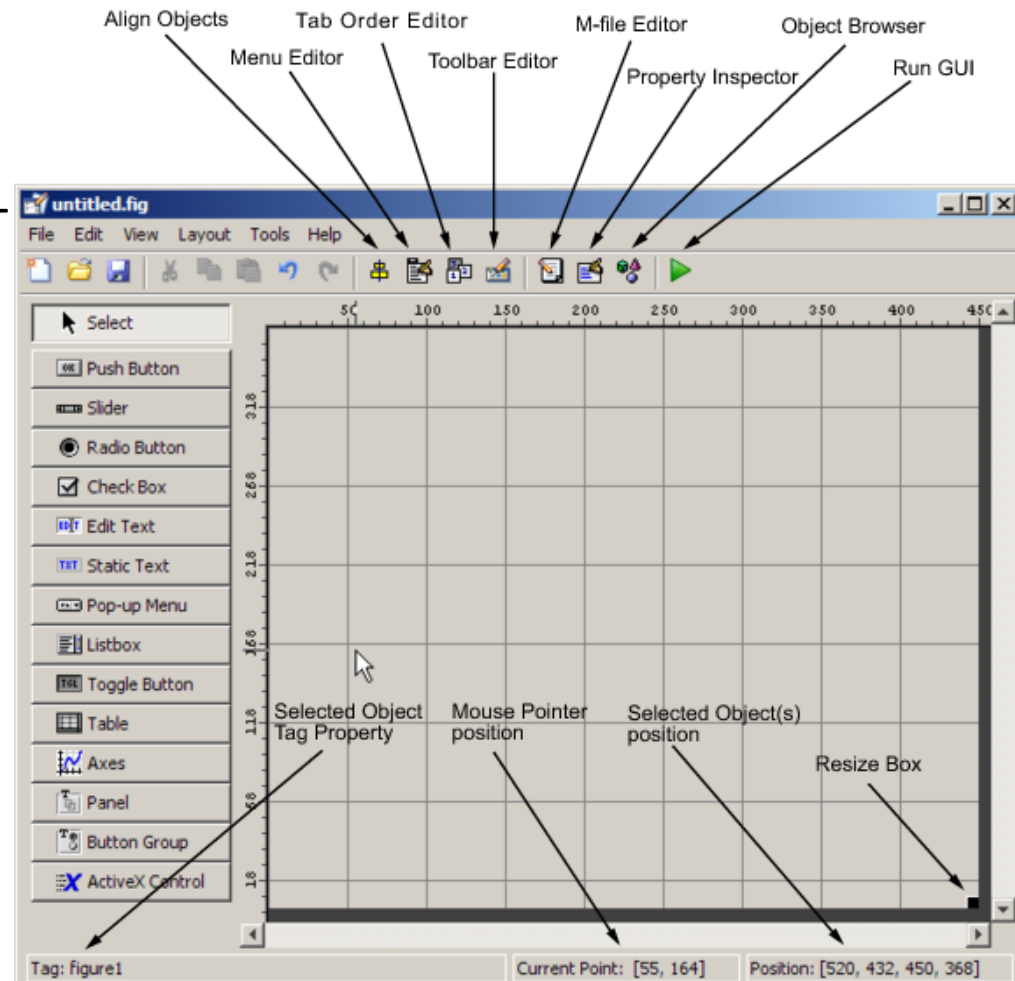
MATLAB (1) - úvod do programovania vedeckých problémov, LS 2017

Základné aktívne prvky rozhrania GUIDE

-Vkladanie ovládacích prvkov štýlom 'drag & drop'

-Rozmiestňovanie pomocou myši + pokročilé možnosti zarovnávania a distribúcie prvkov

-Jednoduché pozmenenie existujúceho rozhrania



MATLAB (1) - úvod do programovania vedeckých problémov, LS 2017

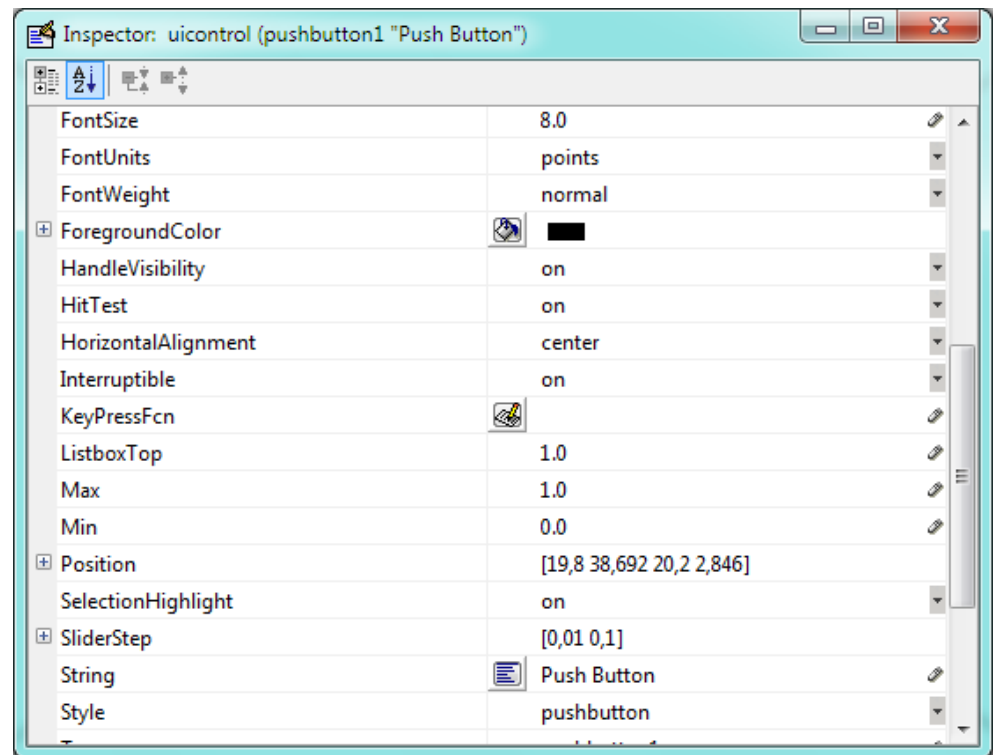
Nastavenie zobrazenia objektu GUI

Funkcia *propertyinspector*

-Vyvoláva sa pravým tlačidlom alebo dvojklikom na zvolený objekt

-Umožňuje interaktívne meniť vlastnosti objektu (podobne ako príkaz *set*)

-Prehľad o všetkých nastaviteľných parametrov objektu



Callback – vyvolanie funkcie

Položka Callback zabezpečuje vyvolanie konkrétneho príkazu alebo funkcie pri použití prvku GUI – napr. pri stlačení tlačidla

Funkcie sú automaticky generované z GUIDE, treba ich však naplniť príkazmi

Syntax zápisu funkcie:

```
function pushbutton1_Callback(hObject, eventdata, handles)
```

`hObject` – handle aktuálneho objektu

`eventdata` – nevyužitá premenná, ktorá ma byť definovaná v budúcich verziách Matlabu

`handles` – štrukturovaná premenná so všetkými premennými typu handle, ktoré sú použité v aktuálnom GUI

všetky definované funkcie – callbacky si automaticky posúvajú tieto polia `hObject`, `eventdata`, `handles`, ktoré môžeme využiť na posúvanie premenných medzi funkciami

MATLAB (1) - úvod do programovania vedeckých problémov, LS 2017

Krátka odbočka o štruktúrovaných premenných

Umožňujú spájať viaceré premenné pod jednu nadradenú, nezávisle od ich typu alebo veľkosti

Napr:

```
clc; clear all; close all
```

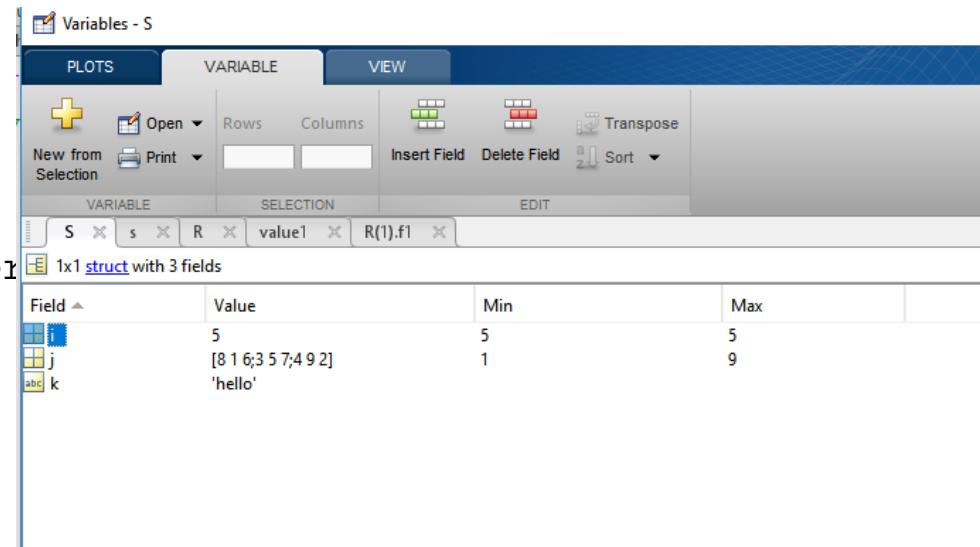
```
% deklarovanie štrukturovanej pr
```

```
S.i=5;
```

```
S.j=magic(3);
```

```
S.k='hello';
```

```
Who
```



Field	Value	Min	Max
i	5	5	5
j	[8 1 6; 3 5 7; 4 9 2]	1	9
k	'hello'		

Načo nám to je?

-vytváranie skupín premenných

-znižujú počet položiek vo workspace = prehľadnejší workspace

-praktické pre posúvanie pri funkciách a callbackoch

Callback – posúvanie premenných

Príkaz *guidata*

Preddefinovane využíva štruktúrovanú premennú *handles*, kde môžeme vytvárať nové polia

```
handles.mojedata=123;
```

Uloženie zmien do zdieľanej premennej realizujeme príkazom:

```
guidata(hObject,handles)
```

Pozn: príkaz *guidata* stačí použiť raz na konci volanej funkcie, nie pri každom vytvorení nového poľa v premennej *handles*