

8. Matlab - štruktúry

Foltin Martin · MATLAB/Comsol

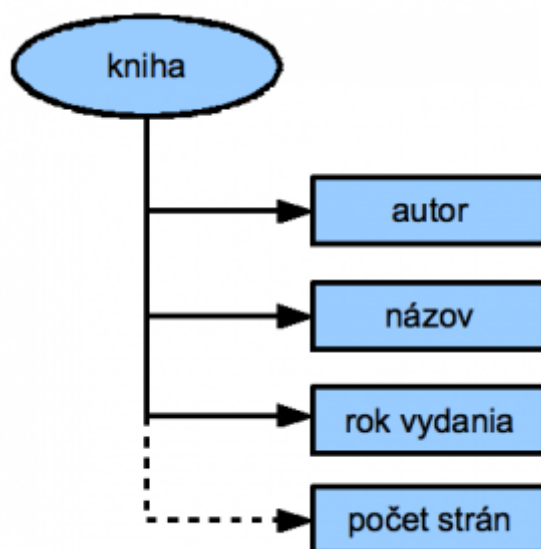
03.07.2009



V predošlej časti seriálu sme sa venovali formátovaniu výstupu. Dnes by sme si mali priblížiť dôležitý dátový typ a tým je štruktúra. Jej pochopenie je dôležité pre budúcu prácu s objektami a dátovým typom bunka (*cell*).

Štruktúra je pomerne zvláštny dátový útvar definuje v sebe pole premenných rôzneho typu. Každá položka pola však musí mať presne definovaný typ a nie je možné ich kombinovať. V rámci pola sa však môžu kombinovať dátové typy ľubovoľne. Na uvedenom príklade budeme kombinovať v štruktúre textové reťazce s celými číslami. Bez problémov by sme ale mohli začleniť do štruktúry aj vektor, alebo maticu.

Náš príklad predstavuje evidenciu kníh v knižnici. Samozrejme uloženie v štruktúre nie je tak efektívne ako keby sme vytvorili databázu, ale pre menšiu knižnicu uloženie v štruktúre postačuje. V knižnici budeme evidovať štruktúru s menom kniha. V rámci položky kniha budeme evidovať autora knihy (*autor*), názov diela (*nazov*) a rok vydania (*rok_vydania*). Táto štruktúra je znázornená na obr. 1.



Obr. 1. štruktrúa **kniha**

V Matlabe máme dve možnosti ako takúto štruktúru vytvoriť. Prvým z nich je

jednoduché priradenie do jednotlivých položiek. Položku od názvu štruktúry oddeľujeme bodkou.

```
štruktúra.položka = hodnota_položky
```

V našom prípade zavedme do evidencie knihu autora Danila Charmsa, Cirkus Sardam vydanú v roku 1995.

```
>> kniha.autor='Danil Charms';
>> kniha.nazov='Cirkus Sardam';
>> kniha.rok_vydania=1995;
>> kniha
kniha =
autor: 'Danil Charms'
nazov: 'Cirkus Sardam'
rok_vydania: 1995
```

O tom koľko naša štruktúra zaberá miesta v pamäti a ako ju Matlab vníma sa presvedčíme príkazom **whos**.

```
>> whos
Name Size Bytes Class Attributes
kniha 1x1 430 struct
```

Druhý spôsob ako definovať štruktúru je využiť príkaz **struct**. Syntax príkazu je :

```
s = struct('field1',VALUES1,'field2',VALUES2,...)
```

V našom prípade by to teda znamenalo

```
>> kniha=struct('autor','Danil Charms',...
'nazov','Cirkus Sardam',...
'rok_vydania',1995)
```

```
kniha =
autor: 'Danil Charms'
nazov: 'Cirkus Sardam'
rok_vydania: 1995
```

Oba uvedené zápisy sú plne ekvivalentné.

Štruktúra v takomto tvare môže nájsť svoje miesto v istých programátorských situáciách (napr. vlastnosti objektov). Skutočná sila a význam štruktúry sa ukáže keď z v rámci štruktúry vytvoríme pole. V našom prípade by to znamenalo definovanie druhej knihy.

```
>> kniha(2).autor='Rob Grand';
>> kniha(2).nazov='Cerveny Trpaslik';
>> kniha(2).rok_vydania=1990;
>> kniha
```

```
kniha =  
1x2 struct array with fields:  
autor  
nazov  
rok_vydania
```

Vidíme, že štruktúru môžeme bez problémov indexovať a tak vytovríme celé pole obsahujúce všetky knihy v našej knižnici. Môžeme si teda štruktúru predstaviť ako kartotečné lístky na ktorých sú záznamy o jednotlivých knihách. K zvolenej knihe sa vieme dostať ak vieme jej poradové číslo v poli.

```
>> kniha(2)  
ans =  
autor: 'Rob Grand'  
nazov: 'Cerveny Trpaslik'  
rok_vydania: 1990
```

Štruktúra nám ale ponúka aj ďalší náhľad na informácie. Ak potrebujeme zistiť všetky mená autorov v knižnici, použijeme príkaz

```
>> kniha.autor  
ans =  
Danil Charms
```

```
ans =  
Rob Grand
```

Môžeme pristupovať aj k jednotlivým pomožkám štruktúry

```
>> kniha(2).nazov  
ans =  
Cerveny Trpaslik
```

V prípade, že sa časom ukáže nedostatočný počet položiek, môžeme do štruktúry jednoducho doplniť novú (napr. počet strán).

```
>> kniha(1).pocet_stran=236  
kniha =  
1x2 struct array with fields:  
autor  
nazov  
rok_vydania  
pocet_stran
```

Táto informácia sa samozrejme dostane len k prvej knihe. Pre ostatné bude vytvorené len prázdne pole, ktoré je možné neskôr naplniť.

```
>> kniha.pocet_stran  
ans =
```

236

```
ans =  
[]
```

O tom ktoré položky naša štruktúra obsahuje sa dozvieme použitím príkazu **fieldnames**. Výstupom tohto príkazu je dátový typ *cell* (bunka) o ktorom sa dozviete v budúcej časti.

```
>> fieldnames(kniha)
```

```
ans =  
'autor'  
'nazov'  
'rok_vydania'  
'pocet_stran'
```

Ak by sa časom ukázalo, že sme definovali niektorú z položiek zbytočne a potrebovali by sme ju odstrániť, použijeme príkaz **rmfield**. Syntax príkazu je

```
S = rmfield(S, 'field')
```

Pre názornosť vymažeme položku `pocet_stran` z našej štruktúry `kniha`.

```
>> kniha=rmfield(kniha, 'pocet_stran');  
>> fieldnames(kniha)
```

```
ans =  
'autor'  
'nazov'  
'rok_vydania'
```

Aktuálny počet evidovaných kníh môžeme zistiť príkazom **length**. Vidíme, že tento príkaz je použiteľný aj vo sfére štruktúr.

```
>> length(kniha)  
ans =  
2
```

V tomto článku sme si predstavili základné úkony, ktoré je možné so štruktúrami v Matlabe vykonávať. Podrobnejšie informácie nájdete v dokumentácii k Matlabu, alebo v nápovede. Akékoľvek otázky súvisiace s oblasťou štruktúr môžeme riešiť v diskusii pod článkom, alebo na diskusnom fóre www.matlab.sk.

