

Sveučilište u Zagrebu
FAKULTET STROJARSTVA I BRODOGRADNJE
Zavod za robotiku i automatizaciju proizvodnih sustava
Katedra za strojarsku automatiku

Računalna matematika

Upravljanje programom, skripte i funkcije

V. Milić

Zagreb, 2020./2021.

Sadržaj

- 1 Upravljanje programom
 - `if-elseif-else-end` kontrola tijeka programa
 - `for-end` petlja
 - `while-end` petlja
- 2 Skripte i funkcije
 - Skripte
 - Funkcije
- 3 Primjeri
- 4 Preporučena literatura

1 Upravljanje programom

- `if-elseif-else-end` kontrola tijeka programa
- `for-end` petlja
- `while-end` petlja

2 Skripte i funkcije

- Skripte
- Funkcije

3 Primjeri

4 Preporučena literatura

UPRAVLJANJE PROGRAMOM

- Programsko upravljanje je redoslijed po kojem se programski kôd izvršava. Upravljanje tijekom programa u svim programskim jezicima, temelji se na uvjetnim naredbama, petljama i pozivima funkcija.
- Često se neka naredba ili niz (blok) naredbi treba izvršiti samo u slučaju ako je neki uvjet zadovoljen. Ponekad je to izvršavanje ovisno o nekoliko međusobno povezanih uvjeta. Kontrola tijekom `if`, koja koristi proširenja `elseif` ili `else` zaključak, služi za takvo uvjetno izvršavanje naredbi.
- Petljom `for` ostvaruje se iterativno izvršavanje naredbe ili bloka naredbi, a upravlja se preko iteracijskog izraza. Koristi se kada je broj iteracija unaprijed poznat.
- Petljom `while` izvršava se naredbu ili blok naredbi niti jedan, jedan ili više puta, ovisno o ispunjenju upravljanog uvjetnog izraza. Može se shvatiti kao ponavljajuća `if` kontrola tijekom.
- Osim gore navedenih najčešćih naredbi, za upravljanje programom postoje još naredbe: `switch-case`, `try-catch`, itd.

if-elseif-else-end KONTROLA TIJEKA PROGRAMA

- MATLABova sintaksa je sljedeća:

```
if izraz
    naredbe
elseif izraz
    naredbe
else
    naredbe
end
```

- *izraz* predstavlja jedan ili više uvjeta definiranih relacijskim ($==$, $\sim=$, $>$, itd.) operatorima i povezanih logičkim ($\&\&$, $\|$, itd) operatorima.
- Svaki rezultat iz *izraz* izračunava se kao istinit (true, tj. 1) ako nije prazan (vektor ili matrica) i sadrži ne-nul elemente. Inače, rezultat iz *izraz* izračunava se kao neistinit (false, tj. 0).

for-end PETLJA

- MATLABova sintaksa je sljedeća:

```
for iteracijska_varijabla = iteracijski_niz  
    naredbe  
end
```

- Naredba for slijedno re-povezuje *iteracijska_varijabla* na svaki element po redu unutar *iteracijski_niz*. Naredba ili naredbe od kojih se sastoji tijelo petlje izvršavaju se jedna po jedna za svaki element.
- Naredbe unutar tijela petlje se ne izvršavaju ako *iteracijski_niz* ne oslobađa nikakve elemente (prazan vektor). U tom slučaju, *iteracijska_varijabla* se ne povezuje na niti jedan način pomoću naredbe for.
- Ako *iteracijski_niz* oslobađa barem jedan element, kad se naredbe petlje završe, *iteracijska_varijabla* ostaje povezana na posljednju vrijednost na koju ju je naredba petlje povezala.

while-end PETLJA

- MATLABova sintaksa je sljedeća:

```
while izraz
    naredbe
end
```

- *izraz* predstavlja jedan ili više uvjeta definiranih relacijskim i logičkim (rijetko) operatorima.
- Ako *izraz* nije istinit (false, tj. 0), niti jedna od naredbi u tijelu petlje se ne izvršava. Ako je *izraz* istinit (true, tj. 1), *naredbe* se izvršavaju.
- Kada se *naredbe* izvrše, uvjet se ponovno izračunava, da se vidi treba li se izvršiti nova iteracija. Ovaj se proces nastavlja sve dok *izraz* ne postane neistinit, nakon čega petlja završava. Tijelo petlje treba sadržavati dio koji će u nekom trenutku učiniti *izraz* neistinitim, ili petlja nikada neće završiti, osim ako se ne izvrši naredbu `break`.

- 1 Upravljanje programom
 - `if-elseif-else-end` kontrola tijeka programa
 - `for-end` petlja
 - `while-end` petlja
- 2 **Skripte i funkcije**
 - Skripte
 - Funkcije
- 3 Primjeri
- 4 Preporučena literatura

SKRIPTE

- Sve naredbe u MATLABu mogu se izvesti i tako da se spremne u datoteku s pomoću bilo kojeg ASCII tekst *editora*, a potom se ta datoteka pozove iz MATLAB-a. Naravno, u MATLABu već postoji *editor*, tj. program za upis i promjenu teksta.
- Takvu datoteku nazivamo skripta (engl. *script-files*). Budući da se ove datoteke spremaju kao `ime.m` one se također nazivaju M-datoteke (engl. *M-files*).
- Skripta nema konstrukcijsku mogućnost prenošenja ulaznih podataka u naredbe koje se u skripti nalaze, nego radi s postojećim varijablama radnog prostora, uz mogućnost stvaranja novih varijabli.
- Kod poziva skripte, u naredbenoj se liniji napiše samo njeno ime, nakon čega se njezine spremljene naredbe izvršavaju jedna za drugom. Na zaslonu se pojavljuju samo rješenja koje naredbe proizvode, a ne i same naredbe.
- Kad radimo sa skriptom njen put (staza, engl. *path*) mora biti definiran u MATLABovom popisu putova. MATLAB po definiciji najprije traži u *folderu* iz kojega je on sam pokrenut i u trenutačnom *folderu* (engl. *current folder, current directory*).

FUNKCIJE

- Funkcije se mogu svrstati u 3 kategorije: ugrađene (engl. *built-in function*), funkcije u MATLABovim programskim alatima (engl. *toolbox*) i korisničke funkcije (engl. *user-defined function*).
- Korisničke funkcije u MATLABu se po pozivu ne razlikuju od ugrađenih funkcija, a po ustroju odgovaraju funkcijama od kojih su napravljeni svi MATLABovi programski alati.
- Funkcije se zapisuju kao ASCII datoteke na disk, a njihovo ime obavezno mora imati ekstenziju `.m`. Ime funkcije treba odgovarati imenu datoteke.
- Korisničkim funkcijama potrebno je davati različita imena od ugrađenih funkcija i tom prilikom poštivati sva pravila koja vrijede za imena varijabli.
- Kako bi se funkcija mogla izvoditi, mora biti smještena u trenutni direktorij ili u direktorij koji je u putu pretraživanja MATLABa.

- Funkcije u MATLABu imaju sljedeći oblik:

```
function [izl_1, izl_2, ... ] = ime_funkcije(ul_1, ul_2, ...)  
% linije komentara koji se ispisuju s help naredbom  
funkcijski kod: izrazi, kontrolne strukture i pozivi drugih funkcija  
izl_1 = izraz konačnog rješenja;  
...
```

- `izl_1, izl_2, ...` `ul_1, ul_2`, se zovu formalni parametri ili samo **parametri**, koji se koriste kod poziva funkcija za pridruživanje stvarnim vrijednostima koji se navode kao **argumenti**. U najjednostavnijem slučaju, funkcija nema nikakvih parametara, što znači da kod poziva funkcija ne uzima nikakve argumente.
- Svaki poziv funkcije pridružuje stvarne vrijednosti, argumente, s parametrima specificiranim u definiciji funkcije.
- Parametri su lokalne varijable funkcije, te svaki poziv funkcije povezuje te varijable s odgovarajućim stvarnim vrijednostima koje pozivatelj navodi kao argumente.
- Niz naredbi koji nije prazan, poznatiji kao tijelo funkcije (engl. *function body*), ne izvršava se kada i naredba `function`. Ono se izvršava kasnije, svaki put kada se funkcija poziva.

- Funkcija može imati podfunkciju (engl. *subfunction*) koja je dostupna drugim funkcijama. Podfunkcija unutar iste datoteke definira se nakon već napisane funkcije. Podfunkcije nisu vidljive izvan datoteke u kojoj su definirane.
- Funkcije mogu imati i promjenjivi broj ulaznih i/ili izlaznih argumenata. U definiciji takvih funkcija koriste se naredbe `varargin` i `varargout`.
- Ako je funkcija vrlo jednostavna, npr. sastoji se samo od jednog izraza, tada je možemo definirati u jednoj liniji MATLABove naredbe, tj. nije potrebno definirati tu funkciju kao posebnu datoteku. Takve funkcije se onda nazivaju **anonimne** funkcije, a njihova sintaksa je sljedeća:

```
ime_funkcije = @(ul_1, ul_2, ...) izraz
```

- Također, ako je potrebno načiniti funkciju koja je definirana nekim jednostavnim matematičkim izrazom moguće je koristiti naredbu `inline`, čija je sintaksa sljedećeg oblika

```
ime_funkcije = inline('izraz', 'ul_1', 'ul_2', ...)
```

Međutim, iz MATLABa: "inline will be removed in a future release. Use Anonymous Functions instead."

- 1 Upravljanje programom
 - if-elseif-else-end kontrola tijeka programa
 - for-end petlja
 - while-end petlja

- 2 Skripte i funkcije
 - Skripte
 - Funkcije

- 3 **Primjeri**

- 4 Preporučena literatura




Pogledati priložene M-datoteke

- 1 Upravljanje programom
 - if-elseif-else-end kontrola tijeka programa
 - for-end petlja
 - while-end petlja

- 2 Skripte i funkcije
 - Skripte
 - Funkcije

- 3 Primjeri

- 4 Preporučena literatura

-  B. H. Hahn, D. T. Valentine. *Essential MATLAB for Engineers and Scientists. Fifth Edition*. Academic Press Elsevier, Waltham, MA, USA, 2013.
-  The MathWorks, Inc. *MATLAB Primer*. Natick, MA, USA, 2016.
-  M. Essert, T. Žilić. *MATLAB – Matrični laboratorij*. FSB, Zagreb, 2004.