Politechnika Wrocławska

Trochę o programowaniu

Zastosowanie programu MATLAB w zagadnieniach inżynierskich

Wojciech Myszka

Katedra Mechaniki, Inżynierii Materiałowej i Biomedycznej

8 listopada 2024

HR EXCELLENCE IN RESEARCH

1. Podstawowe konstrukcje programistyczne

Podstawowe konstrukcje programistyczne

Rozgałęzienie programu: instrukcja `if` I

```
if expression
    statements
elseif expression
    statements
else
    statements
end
```

- `expresion` wyrażenie, o wartości logicznej `true` (prawda) lub `false` (fałsz)

  do jego konstruowania można użyć operatorów relacyjnych (<, >,...) oraz logicznych (&&, &, ||, |, ~)

Rozgałęzienie programu: instrukcja `if` II

Uwaga

Zwracam uwagę na dwa rodzaje operatorów: `&&` i `&` (oraz `||` i `|`).
Te podwójne realizują znaną z języków C i C++ ideę *short circuit* — kończą wyliczanie, gdy wynik jest już określony.

Ale...

...w przypadku instrukcji `if`, `elseif`, `while` zawsze stosowana jest idea „skracania'.

Rozgałęzienie programu: instrukcja `switch`

```
switch switch_expression
   case case_expression
      statements
   case case_expression
      statements
    ...
   otherwise
      statements
end
```

Wartość `switch_expression` porównywana jest z kolejnymi wartościami `case_expression` i realizowana jest ta operacja, dla której wyrażenia są zgodne; gdy nigdzie nie wystąpi zgodność —wykonywane jest polecenie pod etykietą `otherwize`.

`case_expression` może przyjmować kilka wartości, na przykład `{52, 78}`

Politechnika Wrocławska

Pętle: instrukcja `for` I

```
for index = values
   statements
end
```

W tym poleceniu parametr `values` może przyjmować następującą formę:

- `initVal:endVal` — zmienna `index` będzie przyjmować wartości z zakresu od `initVal` do `endVal` (włącznie) zmieniające się z krokiem 1.
- `initVal:step:endVal` — zmienna `index` będzie przyjmować wartości z zakresu od `initVal` do `endVal` zmieniające się z krokiem `step`.
- `valArray` — zmienna `index` przyjmie wartości z pobrane z kolejnych kolumn (tablicy dowolnego typu)

Politechnika Wrocławska

Pętle: instrukcja `for` II

```
for I = eye(4,3)
    disp('Current unit vector:')
    disp(I)
end
```

a rezultat:

```
Current unit vector:
     1
     0
Current unit vector:
     0
     1
Current unit vector:
     0
```

Pętle: instrukcja `for` III

0

Politechnika Wrocławska

Pętle: insrtukcja `while`

```
while expression
    statements
end
```

`expression` ma identyczne znaczenie jak w przypadku instrukcji `if`.
Praktycznie identyczna jak w C

Nie ma instrukcji do

Można ją zasymulować w następujący sposób

```
while true
    statements
    if ~expression
        break
    end
end
```

Politechnika Wrocławska

Skrypty

1. Kod najlepiej zapisać w pliku typu `m` lub `mlx`.
2. Kod można uruchamiać krok po kroku, żeby lepiej rozumieć co się dzieje.

Funkcje

Podczas programowania, tak jak w każdym języku, należy kod dzielić na mniejsze fragmenty o precyzyjnie zdefiniowanych zadaniach, ustalonych danych (parametrach) wejściowych i wynikach (parametrach wyjściowych)

Uwagi I

1. Załóżmy, że mamy tablicę zawierającą zarówno dodatnie jak i ujemne wartości.

   W jaki sposób wybrać z tej tablicy **tylko** wartości dodatnie?

2. Można napisać krótki program:

```
X = randn(1,100)
j = 1;
for i=1:size(x,2)
    if x(i) > 0
        y(j) = x(i);
        j = j + 1;
    end
```

Uwagi II

```
end
plot(y)
```

3. Czy można prościej?

```
Y = x(x > 0);
isequal(y, Y)
ans =
  logical
   1
```

Gdy a jest zmienną skalarną, warunek a > 0 w odpowiedzi daje 1 (prawda) lub 0 (fałsz); gdy a jest tablicą dostajemy tablice zer i jedynek

Uwagi III

```
>> a=[1 -1 2 -2]
a =
     1    -1     2    -2
>> a > 0
ans =
  1×4 logical array
   1   0   1   0
```