

Wstęp do Programowania

Lista 1

1 Wprowadzenie do środowiska MATLAB

Zad. 1

Zapoznaj się z podstawowymi oknami dostępnymi w środowisku MATLAB: **Command Window**, **Current Folder**, **Workspace** i **Command History**.

Zad. 2

Zapoznaj się z podstawowymi poleceniami dostępnymi w środowisku MATLAB:

- `clc`, `close`, `clear`, `who`;
- `pwd`, `ls`, `dir`, `cd ..`;
- `help`.

Sprawdź działanie klawiszy: ↑ i ↓ w konsoli (**Command Window**).

2 Zmienna i tablica

Zad. 3

W środowisku MATLAB nie jest wymagane deklarowanie zmiennych. Zmienne zapisywane są z przestrzeni roboczej (**Workspace**), gdy **poda się nazwę zmiennej i przypisze się do niej wartość**.

W oknie **Command Window** sprawdź polecenia:

```
>> a = 2  
>> a = 2;
```

Zmienna `ans` (ang. *ANSwer*).

```
>> a = 2  
>> b = ans + 3
```

Komentarz %:

```
>> a = 2 % Komentarz  
>> a = 2; % Komentarz
```

Zad. 4

W MATLAB operacje wektorowe i macierzowe są w niego wbudowane i wykonywane automatycznie. Na nasze potrzeby – na tym etapie nauki – **wektor będziemy traktować jako tablicę jednowymiarową a macierz jako tablicę dwuwymiarową.**

Ręczne wypełnienie tablicy jednowymiarowej:

```
>> tablica = [1 2 3 4 5 6 7]
```

Ręczne wypełnienie tablicy dwuwymiarowej:

```
>> tablica = [1 2 3; 4 5 6; 7 8 9]
```

Zapamiętaj jaką rolę pełni w tym przypadku średnik.

Dostęp do elementu tablicy jednowymiarowej:

```
>> tablica = [1 2 3 4 5 6 7]
>> tablica(5)
```

Sprawdź również polecenia:

```
>> tablica(8)
>> tablica(0)
```

Dostęp do elementu tablicy dwuwymiarowej:

```
>> tablica = [1 2 3; 4 5 6; 7 8 9]
>> tablica(2, 3)
```

3 Instrukcja warunkowa if i operatory

W MATLAB instrukcja warunkowa **if** ma kilka wariantów:

Wariant 1

```
1 if wyrażenie
2     polecenie\polecenia
3 end
```

Wariant 2

```
1 if wyrażenie
2     polecenie\polecenia
3 else
4     polecenie\polecenia
5 end
```

Wariant 3

```
1 if wyrażenie 1
2     polecenie\polecenia
3 elseif wyrażenie 2
4     polecenie\polecenia
5 else
6     polecenie\polecenia
7 end
```

Do konstruowania wyrażeń można wykorzystać m.in. operatory (np. arytmetyczne, relacji i logiczne). Poniżej przedstawiono zestawienie najważniejszych operatorów wykorzystywanych w MATLAB.

Operatory arytmetyczne:

- dodawanie: +
- odejmowanie: -
- mnożenie: *
- dzielenie: /
- potęgowanie: ^

Operatory relacji:

- mniejszy od: <
- mniejszy lub równy: <=
- większy od: >
- większy lub równy: >=
- równy: ==
- nierówny: ~=

Operatory logiczne:

- koniunkcja: &
- alternatywa: |
- negacja: ~
- alternatywa wykluczająca: xor

Zad. 4

Pobranie danych od użytkownika z wykorzystaniem polecenia `input`:

```
>> waga = input('Podaj swoją wagę w kilogramach: ');
```

Zad. 5

Wyświetlenie zawartości zmiennej z wykorzystaniem polecenia `sprintf`:

```
>> fprintf('Wpisałeś: %d\n', waga);
```

Zad. 6

Napisz program, który pobierze od użytkownika liczbę a następnie wyświetli ją.

Zad. 7

Napisz program, który pobierze od użytkownika dwie liczby, doda je do siebie a wynik wyświetli.

Zad. 8

Napisz program, który pobierze od użytkownika dwie liczby, a następnie wykona: dodawanie, odejmowanie, dzielenie i mnożenie. Ostateczny wyniki wyświetli.

Zad. 9

Napisz program, który zapyta użytkownika o wiek a następnie poinformuje go czy jest już w wieku emerytalnym.

Zad. 10

Zmodyfikuj program z zad. 9, który poinformuje użytkownika o liczbę lat jakie pozostały użytkownikowi do emerytury jeżeli nie jest w wieku emerytalnym.

Zad. 11

Napisz program, który pobierze od użytkownika liczbę a następnie wyświetli ją jeżeli jest to liczba ujemna.

Zad. 12

Napisz program, który pobierze od użytkownika liczbę a następnie wyświetli wartość bezwzględną z tej liczby.

Zad. 13

Napisz program, który pobierze od użytkownika trzy liczby a następnie wyświetli największą z nich.

4 Instrukcja iteracyjna for

Instrukcja iteracyjna **for** wykonuje określoną liczbę obiegów pętli. Liczba powtórzeń sekwencji poleceń zależy od wartości początkowej i końcowej, które zdefiniowane są w *wyrażeniu*.

Konstrukcja pętli **for** w MATLAB ma postać:

```
1 for wyrażenie
2   polecenie\polecenia
3 end
```

gdzie: *wyrażenie* zazwyczaj zapisywane jest w postaci:

wyrażenie = wartość początkowa : krok : wartość końcowa.

Zad. 14

Napisz program, który wypełni tablicę jednowymiarową liczbami: 1, 2, 3, 4, 5. Uzyskany wynik wyświetl w konsoli.

Zad. 15

Napisz program, który wypełni tablicę jednowymiarową liczbami: od 10 do 100 z krokiem 5. Uzyskany wynik wyświetl w konsoli.

Zad. 16

Napisz program, który zsumuje liczby w samodzielnie wypełnionej tablicy. Uzyskany wynik wyświetl w konsoli.

Zad. 17

Napisz program, który wyszuka w samodzielnie przygotowanej tablicy największą liczbę. Uzyskany wynik wyświetl w konsoli.

Zad. 18

Zmodyfikuj program z zad. 17, który wyszuka w samodzielnie przygotowanej tablicy największą liczbę oraz zapamięta pozycję wyszukanej liczby w tablicy. Uzyskany wynik wyświetl w konsoli.

Zad. 19

Napisz program, który przepisze elementy z tablicy źródłowej do tablicy docelowej w odwrotnej kolejności. Wyświetl zarówno tablicę źródłową jak i docelową.

Zad. 20

Napisz program, który zapisze elementy w tablicy w odwrotnej kolejności bez wykorzystania dodatkowej tablicy. Wyniki wyświetl w konsoli.

Zad. 21

Napisz program, który zsumuje tylko nieparzyste liczby z samodzielnie przygotowanej tablicy. Wyniki wyświetl w konsoli.

Instrukcja typu **for** mogą być zagnieżdżane. Przykład zagnieżdżenia instrukcji **for**:

```
1 for wyrażenie 1
2   for wyrażenie 2
3     polecenie \ polecenia
4   end
5 end
```

Zad. 22

Napisz program, który wypełni tablicę dwuwymiarową kolejnymi liczbami. Wyniki wyświetl w konsoli.

Zad. 23

Napisz program, który wyświetli liczby z nieparzystych kolumn tablicy dwuwymiarowej przygotowanej w zadaniu 22. Wyniki wyświetl w konsoli.

Zad. 24

Napisz program, który wyświetli w konsoli:

```
*
**
***
****
*****
```

Zad. 25

Napisz program, który wyświetli w konsoli:

```
*
**
***
****
*****
****
***
**
*
```

5 Instrukcja iteracyjna `while`

Instrukcja `while` – w przeciwieństwie do instrukcji `if` – wykonuje liczbę obiegów pętli, która nie jest znana z góry. Konstrukcja pętli `while` w MATLAB ma postać:

```
1 while wyrażenie
2     polecenie\polecenia
3 end
```

Zad. 26

Napisz program, który będzie dodawał liczby wprowadzane przez użytkownika dopóki nie zostanie wprowadzona liczba 0.

Zad. 27

Napisz program, który wypełni tablicę jednowymiarową liczbami: od 10 do 100 z krokiem 5 z wykorzystaniem instrukcji `while`. Uzyskany wynik wyświetl w konsoli.

Zad. 28

Napisz program, który będzie dodawał liczby wprowadzane przez użytkownika dopóki nie zostanie wprowadzona liczba 0.

Zad. 29

Napisz program, który wyliczy silnię z podanej przez użytkownika liczby. Uzyskany wynik wyświetl w konsoli.

6 Instrukcja wyboru `switch`

Instrukcja wyboru `switch` wykorzystuje się do zaprogramowania wyboru jednego z kilku możliwych wariantów przebiegu obliczeń. Postać instrukcji w MATLAB jest następująca:

```
1 switch wyrażenie
2 case lista wyboru
3     polecenie\polecenia
4 case lista wyboru
5     polecenie\polecenia
6 otherwise
7     polecenie\polecenia
8 end
```

Zad. 30

Napisz program, który zapyta o nazwę jednej z figur geometrycznych: trójkąt, prostokąt, pięciokąt, sześciokąt a w wyniku poda liczbę boków figury.

Zad. 31

Napisz program, który będzie prostym kalkulatorem operacji arytmetycznych korzystający z instrukcji wyboru **switch**.