

# Zestaw 3

---

## 1. Definiowanie funkcji - [link](#)

Poza użyciem funkcji wbudowanych, Mathematica pozwala użytkownikowi na definiowanie własnych funkcji. Składnię definiowania i wywołania funkcji prezentują poniższe przykłady.

```
In[1]:= MojaFunkcja[x_] := x ^ 2 + 4 * x - 4;  
(*definiowanie funkcji kwadratowej*)  
MojaFunkcja[5]  
(* wywołanie funkcji dla argumentu x=5 *)
```

Out[2]= 41

Możemy definiować funkcję o dowolnej ustalonej ilości argumentów:

```
In[3]:= KolejnaFunkcja[x_, y_, z_] := x ^ 2 + y ^ 2 - z ^ 2;  
(*definicja funkcji 3 zmiennych *)  
KolejnaFunkcja[1, 5, -2](* wywołanie funkcji dla x=1,  
y=5 i z=-2 *)
```

Out[4]= 22

---

## 2. Pure function - [link](#)

Składnia pakietu Mathematica pozwala na definiowanie funkcji użytkownika przy pomocy tzw. "pure function". Przeanalizujemy poniższy przykład.

```
In[5]:= MojaFunkcja1 = Function[x, x ^ 2 + 4 * x - 4]; (*definicja funkcji*)  
MojaFunkcja1[5](*wywołanie funkcji*)
```

Out[6]= 41

Możemy używać pure function w skróconej formie:

```
In[7]:= f = (#^2 + 1) &;
(*definiujemy funkcję,
która dla dowolnego argumentu x oblicza wynik działania x^2+1;
znak # oznacza argument funkcji;
skrótowy zapis pure function kończymy znakiem & *)
f@6
(*wywołanie funkcji następuje poprzez podanie argumentu po znaku @*)
```

Out[7]=

Out[8]= 37

```
In[9]:= g = (#1 + #2^2) &(* definiujemy funkcję dwóch zmiennych*)
g@@{3, 4}(*wywołujemy funkcję g z argumentami 3 i 4,
tzn w miejsce #1 podstawiamy 3, w miejsce #2 podstawiamy 4. W
przypadku wielu zmiennych używamy operatora @@ *)
```

Out[9]=  $3 + 4^2$  &

Out[10]=

19

Użycie pure function pozwala też na pisanie skomplikowanych funkcji wykonujących wiele działań na dowolnej liczbie argumentów.

```
In[95]:= RownanieKwadratowe = Function[{a, b, c}, (*podajemy listę argumentów*)
(* funkcja Print wyświetla tekst w oknie odpowiedzi *)
{(* otwieramy nawias klamrowy *)delta = b^2 - 4*a*c,
(* wyliczamy deltę *)x1 = (-b - Sqrt[delta])/(2*a),
(*obliczamy jeden pierwiastek*)x2 = (-b + Sqrt[delta])/(2*a),
(*obliczamy drugi pierwiastek*)
};
Print["Wyróżnik równania wynosi ", delta];
Print["Pierwsze rozwiązanie = ", x1];
Print["Drugie rozwiązanie = ", x2];
Return[{x1, x2}];
];
(* zamykamy nawias klamrowy
oraz kwadratowy kończący składnię *)
rozwl = RownanieKwadratowe[1, 5, 6];
Wyróżnik równania wynosi 1
Pierwsze rozwiązanie = -3
Drugie rozwiązanie = -2
Out[96]= {-3, -2}
```

```
In[97]:= rozwl
Out[97]=
{-3, -2}

In[98]:= RownanieKwadratowe[1, 2, 1]
Wyróżnik    równania    wynosi    0
Pierwsze    rozwiązanie   =      -1
Drugie      rozwiązanie   =      -1
Out[98]=
{-1, -1}
```

---

## Operacje na funkcjach - [link](#)

Operacje na pure function wspomaga szereg przydatnych funkcji:

**Map[f,lista]** – zastosowanie funkcji f do każdego elementu listy  
(skrót **/@**)

**Apply[f,lista]** - wywołanie funkcji f z argumentem w postaci listy  
(skrót **@@**)

```
In[52]:= Clear[a, b, c, d, e, f]

In[53]:= Reverse /@ {{a, b}, {c, d}, {e, f}}
Map[Reverse, {{a, b}, {c, d}, {e, f}}]
(*stosujemy funkcję Reverse do każdego elementu listy*)
Out[53]=
{{b, a}, {d, c}, {f, e}}

Out[54]=
{{b, a}, {d, c}, {f, e}}

In[55]:= Plus @@ Range[10](*sumujemy liczby od 1 do 10*)
Apply[Plus, Range[10]]
Out[55]=
55

Out[56]=
55

In[57]:= Select[RandomInteger[10, {20, 2}], Apply[CoprimeQ]]
Out[57]=
{{2, 3}, {7, 5}, {10, 3}, {10, 1}, {5, 1}, {7, 9}, {5, 4}, {3, 10}, {3, 2}, {9, 5}, {1, 2}}

In[58]:= X = RandomInteger[10, {20, 2}]
(*losujemy macierz wymiaru 20x2 spośród liczb całkowitych od 0 do 10*)
Out[58]=
{{7, 4}, {8, 7}, {1, 5}, {4, 6}, {0, 9}, {7, 3}, {10, 7}, {6, 2}, {1, 4},
{1, 10}, {4, 0}, {4, 8}, {0, 0}, {0, 0}, {8, 9}, {3, 9}, {3, 9}, {3, 0}, {1, 1}, {5, 7}}
```

```
In[59]:= Select[X, Apply[CoprimeQ]]
(*wybieramy tylko te wiersze, w których wartości są wzajemnie względnie pierwsze*)
Out[59]=
{{7, 4}, {8, 7}, {1, 5}, {7, 3}, {10, 7}, {1, 4}, {1, 10}, {8, 9}, {1, 1}, {5, 7}}
```

## Instrukcje warunkowe

```
In[60]:= abs[x_] := If[x < 0, -x, x]
abs /@ {-2, -1, 0, 1, 2}
Out[61]=
{2, 1, 0, 1, 2}

In[63]:= rownanie = Function[{a, b, c}, delta = b^2 - 4*a*c;
    If[delta < 0, (* warunek do sprawdzenia *)
        Print["Pierwiastki zespolone"], (* jeśli warunek spełniony *)
        Print["Pierwiastki rzeczywiste"] (* jeśli warunek fałszywy *)];
    (* koniec Ifa *)
];
rownanie[1, 2, -1](*wywołanie funkcji*)
rownanie[1, 0, 1](*wywołanie funkcji*)
Pierwiastki rzeczywiste
Out[64]=

Pierwiastki zespolone
Out[65]=
```

## Zadania

**Zadanie 1.** Zdefiniować funkcję **deMoivre[z,n]**, która oblicza wszystkie pierwiastki zespolone liczby **z** stopnia **n** zgodnie ze wzorem de Moivre'a.

**Zadanie 2.** Utworzyć tablicę wartości funkcji **Sin**, **Cos**, **Tan** dla argumentów generowanych przez polecenie **Pi/12Range[15]** (zob. tabela poniżej).

x	$\sin x$	$\cos x$	$\operatorname{tg} x$
$\frac{\pi}{12}$	$\frac{-1+\sqrt{3}}{2\sqrt{2}}$	$\frac{1+\sqrt{3}}{2\sqrt{2}}$	$2 - \sqrt{3}$
$\frac{\pi}{6}$	$\frac{1}{2}$	$\frac{\sqrt{3}}{2}$	$\frac{1}{\sqrt{3}}$
$\frac{\pi}{4}$	$\frac{1}{\sqrt{2}}$	$\frac{1}{\sqrt{2}}$	1
$\frac{\pi}{3}$	$\frac{\sqrt{3}}{2}$	$\frac{1}{2}$	$\sqrt{3}$
$\frac{5\pi}{12}$	$\frac{1+\sqrt{3}}{2\sqrt{2}}$	$\frac{-1+\sqrt{3}}{2\sqrt{2}}$	$2 + \sqrt{3}$
$\frac{\pi}{2}$	1	0	ComplexInfinity
$\frac{7\pi}{12}$	$\frac{1+\sqrt{3}}{2\sqrt{2}}$	$-\frac{-1+\sqrt{3}}{2\sqrt{2}}$	$-2 - \sqrt{3}$
$\frac{2\pi}{3}$	$\frac{\sqrt{3}}{2}$	$-\frac{1}{2}$	$-\sqrt{3}$
$\frac{3\pi}{4}$	$\frac{1}{\sqrt{2}}$	$-\frac{1}{\sqrt{2}}$	-1
$\frac{5\pi}{6}$	$\frac{1}{2}$	$-\frac{\sqrt{3}}{2}$	$-\frac{1}{\sqrt{3}}$
$\frac{11\pi}{12}$	$\frac{-1+\sqrt{3}}{2\sqrt{2}}$	$-\frac{1+\sqrt{3}}{2\sqrt{2}}$	$-2 + \sqrt{3}$
$\pi$	0	-1	0
$\frac{13\pi}{12}$	$-\frac{-1+\sqrt{3}}{2\sqrt{2}}$	$-\frac{1+\sqrt{3}}{2\sqrt{2}}$	$2 - \sqrt{3}$
$\frac{7\pi}{6}$	$-\frac{1}{2}$	$-\frac{\sqrt{3}}{2}$	$\frac{1}{\sqrt{3}}$
$\frac{5\pi}{4}$	$-\frac{1}{\sqrt{2}}$	$-\frac{1}{\sqrt{2}}$	1

**Zadanie 3.** Napisać funkcję, która dla dwóch 3-wymiarowych wektorów **w**, **v** liczy ich długość, iloczyn skalarny oraz kąt między nimi.

#### Zadanie domowe

uogólnij funkcję z zadania 3 dla dowolnych n-wymiarowych wektorów.

**Zadanie 4.** Obliczyć obwody okręgów i pola kół, jeżeli ich promienie zawarte są na liście **dane** wygenerowanej przez polecenie

**dane=Table[0.1 RandomInteger[100],[k,1,100]] .**

**Zadanie 5.** Napisać funkcję, która sprawdza, czy podane liczby całkowite **m** i **n** są względnie pierwsze. Jeżeli tak, funkcja wypisuje informację, że liczby **m** i **n** są względnie pierwsze, następnie:

- oblicza wartość **m/n** z dokładnością do 9 miejsc po przecinku

- wypisuje wszystkie liczby naturalne z przedziału  $[m,n]$  (lub  $[n,m]$ )
- wyświetla długość przedziału  $[m,n]$  (lub  $[n,m]$ )
- losuje liczbę całkowitą z przedziału  $[m,n]$  (lub  $[n,m]$ ).

Pamiętaj aby na początku sprawdzić, czy użytkownik podał liczby całkowite.