

# **Wprowadzenie, typy, definicje i deklaracje**

# Alfabet

Alfabet języka C, tj. zbiór znaków, za pomocą których zapisuje się programy w tym języku.

Alfabet zawiera:

- małe i duże litery alfabetu łacińskiego oraz "\_" (znak podkreślenia zaliczany do liter),
- cyfry arabskie,
- znaki specjalne: + - \* / = < > ( ) [ ] { } . , : ; ' ' " " ^ ! # & % | ~ ?

# Identyfikator

Ciągi znaków alfabetu zaczynające się od litery,  
dalej litery lub cyfry

Standardowo początkowe 32 znaki, odróżniając duże i małe litery

alfa	Alfa	AlfA	ALFA
Cena_Mleka	KosztTransportu		

Słowa kluczowe - identyfikatory zastrzeżone

# Komentarze

/\*

.....

bez zagnieżdżania

.....

\*/

//

.....

# Typy liczb całkowitych

char      short      int      long      long long

signed

unsigned

typ	signed	unsigned	bajty
char	- 128, + 127	0, 255	1
short	- 32 768, + 32767	0, 65535	2
int, long	- 2 147 483 648, + 2 147 483 647	0, 4 294 967 295	4
long long	- 9 223 372 036 854 775 808, + 9 223 372 036 854 775 807	0, 18 446 744 073 709 551 615	8

# Typy zmiennopozycyjne

float

double

long double

typ	zakres	cyfry	bajty
float	$\pm 3.4 \cdot 10^{\pm 38}$	7	4
double, long double	$\pm 1.7 \cdot 10^{\pm 308}$	15	8

# Liczby całkowite

```
12      154555      // dziesiętnie
012     03777453    // ósemkowo
0xAB    0x5c5d      0xffff45a // heksadecymalnie
```

Typ liczby :

- na podstawie wartości (domyślny `int`)

```
12      25467      // signed int
```

```
34760548093      // signed long long
```

- wskazany w zapisie liczby

15L	0777771	0xFF4FFFL	// signed long
25411	-457LL	0xAB56LL	// signed long long
45211u	0xffau		// unsigned int
3000000000u1	0xC56AFB44UL		// unsigned long
-120ULL	78u11		// unsigned long long



# Liczby zmiennopozycyjne

1.25	0.343	.5	2.
35.56E-12	0.34e2	5e3	17.18E+28

Typ liczby :

- na podstawie wartości (domyślny **double**)
- wskazany w zapisie liczby

12.545f	// float
0.2345676543F	// float
0.5e-31	// long double
0.9999998899E456L	// long double

# Znaki

Liczby całkowite typu: `char`

`'a'`

`'5'`

`'+'`

`'.'`

`'A'`

`'\071'`

`'\x41'`

`'\x5F'`

`'\n'`

`'\t'`

`'\r'`

`'\\'`

`'\"'`

# Łańcuchy

"Programowanie w języku C"

"Wynik : "

"\tImię\tNazwisko\tMiejsce zamieszkania\n"

"\x16\x16\x02" // znaki specjalne

"Początkujący programista, czytający kod programu w C może odnieść bardzo nieprzyjemne wrażenie, które można opisać cytatem \"ja nigdy tego nie opanuję\"."

(<https://pl.wikibooks.org/wiki/C/Podstawy>)

# Deklaracje zmiennych

< identyfikator, wartość >

char            signed char

int            signed            signed int

short          short int          signed short int

long           signed long          long int          signed

long long          signed long long

unsigned char

unsigned int            unsigned

unsigned short          unsigned short int

unsigned long            unsigned long int

unsigned long long

float

double

long double

# Deklaracje zmiennych

```
int      i;
```

```
char      a, b, c;
```

```
unsigned long      duza_odleglosc;
```

```
float      KursDolara;
```

```
double      masa, gestosc;
```

# Definicje zmiennych

```
int      licznik = 125,    suma = 0;
```

```
float    dokladnosc = 0.0005,  
  
         uchyb = 0.001;
```

```
double   moc = 15e6,      straty = 1500;
```

```
double   alfa = 3.34, beta, jota = 15.15, kappa;
```

# Stałe

```
const int      dni = 7,      tygodnie = 52;
```

```
const float    pi = 3.14159,    e = 2.71828;
```

```
const double   Avogadro = 6.022E23;
```

# Typy wyliczeniowe

```
enum id_typu { lista_stałych } id_zmiennej ;
```

```
enum dni {ni, po, wt, sr, cz, pi, so};  
    /* ni == 0, po == 1, ... , so == 6 */
```

```
enum dni {ni=1, po, wt, sr, cz, pi, so};  
    /* ni == 1, po == 2, ... , so == 7 */
```

```
dni Egzamin, Dobry = cz;  
Egzamin = Dobry;
```

```
enum TW1 {t1, t2, t3 = 0, t4, t5, t6 = 1, t7};  
    /* t4 == 1, t5 == 2, t7 == 2 */
```

```
enum {A = 0x41, B, C, X = 0x58} znak;  
    // zmienna typu wyliczeniowego
```

```
znak = C;        // poprawnie
```

```
znak = 0x41;     // błąd
```



# Typ logiczny

```
bool stan, obecosc = true;
```

```
stan = true;
```

```
obecosc = false;
```

```
enum BOOL {FALSE, TRUE}; // 0 , 1
```

```
BOOL flaga, status = TRUE;
```

```
flaga = FALSE;
```

```
status = flaga;
```

# Operator przypisania

```
int    i,    j,    k;
```

```
i = 1;      //zmienna i przyjmuje wartość 1
```

```
j = i;      //zmienna j przyjmuje wartość zmiennej i
```

```
k = dni;    //zmienna k przyjmuje wartość stałej dni
```

# Notacja węgierska

Przedrostek	Typ danych	Przykład
<b>b</b>	<b>bool</b>	<b>bJeszczeRaz</b>
<b>c</b>	<b>char</b>	<b>cKodPolecenia</b>
<b>l</b>	<b>long</b>	<b>lDuzyKaliber</b>
<b>n</b>	<b>int</b>	<b>nLicznikPierwszy</b>
<b>p</b>	wskaźnik	<b>pAdresNowejCeny</b>
<b>a</b>	tablica	<b>aDaneTestowe</b>
<b>s</b>	łańcuch znaków	<b>sStosownyNapis</b>

# Nazywanie typów danych

```
typedef      typ      nowy_identyfikator
```

```
typedef      char*      string;  
string  S1,  S2,  S3 = "napis";
```

```
typedef      int      num;  
num  k;  
int  l = 5;  
k = 1; // typ num jest równoważny z typem int
```

```
typedef      long      BIG;  
unsigned      BIG  ww;      // błąd
```