

Typy danych liczbowych

Język C++ zawiera pewną liczbę typów danych zdefiniowanych pierwotnie, dzięki którym można przedstawiać liczby całkowite, rzeczywiste i znaki.

| Typ | Długość | Zakres |
|---------------|---------|--|
| unsigned char | 8 bits | 0 .. 255 |
| char | 8 bits | -128 .. 127 |
| short int | 16 bits | -32,768 .. 32,767 |
| unsigned int | 32 bits | 0 .. 4,294,967,295 |
| int | 32 bits | -2,147,483,648 .. 2,147,483,647 |
| unsigned long | 32 bits | 0 .. 4,294,967,295 |
| long | 32 bits | -2,147,483,648 .. 2,147,483,647 |
| enum | 16 bits | -2,147,483,648 .. 2,147,483,647 |
| float | 32 bits | 3.4×10^{-38} .. $3.4 \times 10^{+38}$ |
| double | 64 bits | 1.7×10^{-308} .. $1.7 \times 10^{+308}$ |
| long double | 80 bits | 3.4×10^{-4932} .. $1.1 \times 10^{+4932}$ |

Do przedstawiania liczb całkowitych korzystamy z typów `char`, `short`, `int` i `long`. Typy całkowite są ze znakiem lub bez znaku. Do przedstawiania liczb rzeczywistych mamy do dyspozycji typy `float`, `double` i `long double`. Zdarza się, że typy `double` i `long double` są identyczne.

Wartości stałe

Literałami, czyli dosłownie stałymi, nazywamy pojawiające się w tekście programu dane zapisane w sposób dosłowny. Z każdym literałem związany jest typ. Literały są *nieadresowalne*.

Literały całkowite:

można zapisać w systemie dziesiętnym (tradycyjny zapis), ósemkowym (zapis zaczynamy cyfrą 0 zero) lub w szesnastkowym (zapis rozpoczynamy 0x lub 0X zero iks):

```
20          //dziesiętnie
024         //ósemkowo
0x14       //szesnastkowo
```

Domyślnym typem literałów całkowitych jest typ `int` ze znakiem. Jeśli stałą całkowitą zakończymy literą `L` lub `l` to literał będzie należał do typu `long`.

Podobnie zapisuje się literał typu `unsigned`, kończąc literą `U` lub `u`. Np.

`128u` `1034L` `1348UL` `124Lu`

Literały zmiennopozycyjne zapisujemy albo w postaci wykładniczej z literką `E` lub `e` albo w sposób tradycyjny z kropką oddzielającą część całkowitą od ułamkowej. Domyślnym typem jest typ `double`. Aby otrzymać literał należący do typu `float` należy na końcu literału dopisać literkę `F` lub `f`. Aby otrzymać literał typu `long double` na końcu literału trzeba dopisać `L` lub `l`.

Przykłady literałów zmiennopozycyjnych:

`3.14f` `0.3F` `45.7L` `2e-10L` `2.e-1`
`.2e-10F` `.0`

Nie są poprawnymi literałami

`-10F` (brak przynajmniej kropki)

`E10` (brak mantysy).

Ale jest przyjmowane bez błędu `-2U` co daje jako stałą `65534`.

Literały znakowe zapisuje się ujmując znak w pojedynczy cydzysłów:

`'a'` `'2'` `' '`

W pamięci przechowywane są jako liczby całkowite typu `char`. Zamiast znaku widocznego możemy również skorzystać z opisu znaku, rozpoczynającego się od backslasha po którym podajemy kod ASCII znaku będący liczbą ósemkową lub szesnastkową w postaci `***`, gdzie `***` oznacza zapis przy użyciu co najwyżej trzech cyfr liczby ósemkowej (zero wiodące można lub należy pominąć) lub liczby szesnastkowej (można zacząć od `x`):

`'\107'` `'\x47'`

opisują tą samą stałą a mianowicie literę `G`.

Można posługiwać się znakami sterującymi:

| | |
|-----------------|-----------------------|
| <code>\n</code> | nowy wiersz |
| <code>\t</code> | pozioma tabulacja |
| <code>\v</code> | pionowa tabulacja |
| <code>\b</code> | cofnięcie (backspace) |
| <code>\r</code> | do początku wiersza |
| <code>\f</code> | nowa strona |

| | |
|----|-----------|
| \\ | backslash |
| \? | znak ? |
| \' | apostrof |
| \" | cudzysłów |

Uwaga. Na ogół brak sygnalizacji dla błędnego zapisu stałej znakowej np.

```
cout << 'z1';
```

Łańcuchy (literały napisowe) składają się z ciągu znaków ujętego w podwójny cudzysłów. W przypadku gdy tekst nie mieści się w jednym wierszu, to na końcu takiego wiersza należy postawić \ (backslash) i w następnym kontynuować tekst (od początku linii bo wchodzi spacja). W łańcuchu mogą występować znaki sterujące i opisy znaków. Typem tych danych jest tablica znaków, która zawiera sam literał jak i znak pusty (null), dodany przez kompilator jako ostatni jej element. Zatem jest różnica między 'A' (jeden znak) a "A" (dwa znaki).