

**Zmienne i ich typy**

Programy, których zadaniem jest tylko wypisanie jakiejś informacji na ekranie monitora (choćby te z poprzednich zajęć) są mało ciekawe i przydatne w praktyce (ale za to bardzo dobre na początku nauki programowania). W wielu przypadkach będziemy chcieli, aby programy pobierały od nas pewne dane i wykonywały na nich operacje. Są to działania dość łatwe (jak wyświetlenie rzeczonych danych w niezminionej postaci), jednak wymagają przechowania przez jakiś czas uzyskanej porcji informacji. W językach programowania służą do tego **zmienne**.

**Zmienna** (ang. variable) to miejsce w pamięci operacyjnej, przechowujące pojedynczą wartość określonego typu. Każda zmienna ma nazwę, dzięki której można się do niej odwoływać.

**Typ** określa nam rodzaj informacji, jakie można przechowywać w naszej zmiennej. Mogą to być liczby całkowite, rzeczywiste, pojedynczy znak i tak dalej. Możemy także sami tworzyć własne typy zmiennych. Na razie jednak powinniśmy zapoznać się z dość szerokim wachlarzem typów standardowych, które to obrazuje niniejsza tabelka. Wielkość i zakres liczb różnią się w zależności od systemu operacyjnego.

nazwa typu	opis	zakres	wielkość w bajtach
int	liczba całkowita (dodatnia lub ujemna)	Liczby całkowite o takim zakresie jak short albo long	2 lub 4
short	mała liczba całkowita (dodatnia lub ujemna)	-32768 do 32767 (ze znakiem) lub 0 do 65535 (bez znaku)	2
long	duża liczba całkowita (dodatnia lub ujemna)	-2147483648 do 2147483647 (ze znakiem) lub 0 do 4294967295 (bez znaku)	4
float	liczba rzeczywista (zmiennoprzecinkowa)	$3.4 \cdot 10^{-38}$ do $3.4 \cdot 10^{+38}$	4
double	liczba rzeczywista (zmiennoprzecinkowa podwójnej precyzji)	$1.7 \cdot 10^{-308}$ do $1.7 \cdot 10^{+308}$	8
long double	liczba rzeczywista (zmiennoprzecinkowa rozszerzonej podwójnej precyzji)	$3.4 \cdot 10^{-4932}$ do $1.1 \cdot 10^{+4932}$	10
bool	wartość logiczna (true lub false)	Cyfra 0 to false, każda inna to true	1
char	pojedynczy znak	-127 do 128 (ze znakiem kodu ascii) albo 0 do 255 (bez znaku kodu ascii)	1

Aby użyć zmiennych w programie, należy je najpierw zadeklarować. Robimy to według schematu:

```
typ_zmiennej nazwa_zmiennej;
```

Zmienną możemy deklarować w dowolnym miejscu w programie (oczywiście przed miejscem jej użycia), niemniej, ze względów estetycznych, najczęściej robimy to na początku funkcji `main()`. W powyższej deklaracji zmienna nie ma przypisanej wartości początkowej. Możemy to zrobić za pomocą operatora przypisania `=` w następujący sposób:

```
typ_zmiennej nazwa_zmiennej=wartosc_zmiennej;
```

lub

```
typ_zmiennej nazwa_zmiennej;  
nazwa_zmiennej=wartosc_zmiennej;
```

Przy tworzeniu nazw zmiennych musimy pamiętać o kilku zasadach:

1. Nazwa może zawierać jedynie cyfry, litery i znak podkreślenia
2. Nazwa nie może zaczynać się od cyfry
3. Rozróżnia się małe i wielkie litery

4. Nazwami nie mogą być słowa kluczowe języka c++ np. `int`, `return`, `include`, itp.
5. W nazwach nie stosujemy spacji i polskich liter.

Oto kilka przykładów:

```
int liczba;

liczba=13;

float Liczba1, Liczba_2, b;

Liczba_1=3.34;

Liczba_2=5.456e3;

b=4567.78e-2;

double a=2.3444;

char znak='J';

char znak_2=74;

unsigned int dodatnia;

dodatnia=2;

unsigned char Symbol;
```

Kilka z nich wymaga wyjaśnienia. Po pierwsze, jeśli chcemy zadeklarować kilka zmiennych tego samego typu, możemy to zrobić tak jak w wierszu trzecim (poszczególne zmienne oddzielamy przecinkiem). Po drugie w liczbach zmiennoprzecinkowych do oddzielenia części ułamkowej stosujemy kropkę a nie przecinek. Ponadto liczbę taką możemy zapisać w tzw. **postaci naukowej** (patrz wiersz 5 i 6). Wtedy

```
5.456e3=5456,
4567.78e-2=45.6778.
```

Po trzecie wartości zmiennej typu `char` umieszczamy w apostrofach (patrz wiersz 8), a jeśli znak wypisujemy przy pomocy kodu ASCII, to wartością takiej zmiennej będzie liczba z tablicy kodu ASCII symbolizująca dany znak. W naszym przykładzie (patrz wiersz 9) zmienna `Symbol` symbolizuje literę J. Z kodem ASCII przy zmiennych typu znakowego związane jest również słowo kluczowe `unsigned`, które powoduje, że wartość zmiennej przy którym ono stoi musi być liczbą dodatnią. Jest to o tyle istotne, że znaki, których kod ASCII jest większy od 127 będą zwracały liczby ujemne zamiast faktycznego numeru kodu. Dla przykładu należy użyć deklaracji:

```
unsigned char Znak=145;
```

zamiast

```
char Znak=145;
```

Z powyższych rozważań na temat słowa kluczowego `unsigned` domyślamy się, że deklaracja

```
unsigned int dodatnia;
```

oznacza, że wartościami zmiennej `dodatnia` mogą być tylko liczby dodatnie.

Zajmiemy się teraz problemem wypisywania na ekranie monitora wartości zadeklarowanych zmiennych. W powyższych przykładach niektórym zmiennym przypisywaliśmy wartości od razu bez możliwości zmiany („na sztywno”). W C++ istnieje możliwość wczytywania danych z klawiatury. Prześledźmy poniższy kod

```
#include<iostream>
using namespace std;
int main()
{
    int liczbac;
    float liczbar=2.34;
    cout << "Podaj liczbe całkowita: ";
    cin >> liczbac;
    cout << "Twoje liczby to: "<<endl;
    cout << "\t liczba całkowita: " << liczbac << endl;
    cout << "\t liczba rzeczywista: " << liczbar << endl;
    system("pause");
    return 0;
}
```

Liczba rzeczywista dana jest „na sztywno”. Liczbę całkowitą wczytujemy z klawiatury za pomocą polecenia `cin>>`. Następnie wypisujemy dane liczby na ekran przy pomocy ich nazw (`\t` – tabulator). Aby użyć poleceń `cout` oraz `cin` należy dołączyć do programu bibliotekę `iostream`.

### Zadanie 1.

Napisać program (w pliku o nazwie `typy.cpp`), który wypisze na ekran liczby: 5676757567567, 4.3455, -3456 oraz literę B.

## Stałe

Stałe są w swoim przeznaczeniu bardzo podobne do zmiennych - tyle tylko że mają niezmiennie wartości. Używamy ich, aby nadać znaczące nazwy jakimś nie zmieniającym się wartościom w programie.

**Stała** to niezmienna wartość, której nadano nazwę celem łatwego jej odróżnienia od innych, często podobnych wartości, w kodzie programu. Jej deklaracja, na przykład taka:

```
const int STALA = 10;
```

przypomina nieco sposób deklarowania zmiennych – należy także podać typ oraz nazwę. Słowo `const` (ang. constant – stała) mówi jednak kompilatorowi, że ma do czynienia ze stałą, dlatego oczekuje również podania jej wartości. Wpisujemy ją po znaku równości `=`.

W większości przypadków stałych używamy do identyfikowania liczb - zazwyczaj takich, które występują w kodzie wiele razy i mają po kilka znaczeń w zależności od kontekstu. Pozwala to uniknąć pomyłek i poprawia czytelność programu. Stałe mają też tę zaletę, że ich wartości możemy określać za pomocą innych stałych, na przykład:

```
const int BRUTTO = 2000;
const int PODATEK = 22;
```

Jeżeli kiedyś zmieni się jedna z tych wartości, to będziemy musieli dokonać zmiany tylko w jednym miejscu kodu – bez względu na to, ile razy użyliśmy danej stałej w naszym programie. Poniżej jeszcze inne przykłady stałych:

```
const int DNI_W_TYGODNIU = 7;  
const float PI = 3.141592653589793;  
const int MAX_POZIOM = 50;
```