

## Struktura programu w języku c++

Kod programu w języku c++ jest zwykłym tekstem napisanym w dowolnym edytorze w postaci tzw. **pliku źródłowego**. Plikowi takiemu nadaje się zwykle rozszerzenie **cpp** i kompiluje za pomocą kompilatora, aby utworzyć działającą aplikację (kompilator konwertuje plik źródłowy na plik wykonywalny). Dla ułatwienia wielu programistów wykorzystuje zintegrowane środowiska programistyczne, które zawierają niezbędne narzędzia do tworzenia programu m.in. edytor i kompilator. Takim środowiskiem jest np. **Dev c++**.

Poniżej przedstawiamy standardowy szablon programu c++:

```
#include<...>
...
#include<...>
using namespace std;
int main()
{
    Kod programu
    system("pause");
    return 0;
}
```

Omówimy teraz krótko poszczególne elementy tego szablonu.

Grupa wierszy `#include<...>` to grupa tzw. dyrektyw preprocesora. Każda z dyrektyw zawiera specjalne instrukcje dla kompilatora w postaci dołączonych plików (bibliotek), co rozszerza możliwości c++. Istnieje bardzo dużo dyrektyw (a właściwie bibliotek), a do najczęściej używanych należą: `#include<cmath>` - zawiera wiele przydatnych funkcji matematycznych m.in. `sqrt()`, `pow()`, itd., `#include<iostream>` - pozwala ona m.in. wyświetlać tekst na ekranie monitora oraz wczytywać dane z klawiatury. Inne biblioteki wprowadzimy przy okazji omawiania kolejnych zagadnień.

Wiersz `using namespace std;` jest poleceniem, które nakazuje użycia standardowej przestrzeni nazw `std`. Pojęcie przestrzeni nazw służy do określenia, które zmienne i funkcje można użyć w danym miejscu i zostało wprowadzone po to, aby zapobiegać nakładaniu się nazw zmiennych, funkcji itp. (na przestrzeni lat wraz z rozwojem języka powstawało coraz więcej bibliotek, przez co wzrastała liczba nazw zmiennych, funkcji itp. i w rezultacie zaczęło brakować tych nazw). Można tworzyć własne przestrzenie nazw, jednak my będziemy korzystać z przestrzeni standardowej.

Następnie definiujemy funkcję o nazwie `main()`, (jest to główna funkcja programu) wewnątrz której tworzymy kod źródłowy programu. Na końcu tej funkcji umieszczamy wiersze: `system("pause");` - polecenie to zatrzymuje wykonanie programu do momentu naciśnięcia jakiegoś klawisza (pozwala to zobaczyć efekt pracy programisty); oraz wiersz `return 0;` - polecenie to powoduje, że funkcja `main()` zwraca wartość 0, co oznacza, że program skompilował się pomyślnie.

## Zasady stosowane w pisaniu kodu źródłowego

1. W języku c++ rozróżnia się małe i wielkie litery (polecenia c++ piszemy małymi literami).
2. Większość instrukcji w kodzie źródłowym należy kończyć średnikiem.
3. W kodzie programu warto używać komentarzy dla poprawienia czytelności kodu szczególnie w obszernych programach. Komentarz jednowierszowy umieszczamy po znaku `//`, zaś wielowierszowy między znakami `/* ... */`. Komentarze są ignorowane przez kompilator.
4. W kodzie źródłowym nie używamy polskich liter.

**Zadanie 1.**

Stworzyć plik `pierwszy.cpp` i wpisać w nim następujący kod:

```
#include<iostream>
using namespace std;
int main()
{
    cout << "Witaj, to jest twoj pierwszy program w jezyku C++"<<endl;
    system("pause");
    return 0;
}
```

następnie skompilować plik oraz uruchomić program.

W powyższym kodzie należy wyjaśnić następujący fragment

```
cout << "Witaj, to jest twoj pierwszy program w jezyku C++"<<endl;
```

Polecenie `cout` (tzw. strumień wyjściowy) pozwala, przy pomocy operatora przypisania `<<`, wypisać tekst w cudzysłowie na ekranie monitora. Polecenie `endl` powoduje przejście kursora do nowej linii (pełni rolę entera).

**Zadanie 2.**

Napisać program, który wyświetli na ekranie monitora następującą ramkę:

```
*****
*      imię i nazwisko      *
*      Analiza finansowa    *
*      grupa 1              *
*****
```