

1. Pierwszy program

<pre>// mój pierwszy program w C++ #include <iostream> using namespace std; int main() { cout << "Witam" << endl; return 0; }</pre>	<pre>/* mój pierwszy program w C mogę go również skompilować jako program C++ */ #include <stdio.h> int main() { printf("Witam\n"); return 0; }</pre>
---	---

1.1 Program w języku C++

1.1.1 Komentarz

- Znaki `//` oznaczają początek komentarza. Komentarz rozciąga się od znaków `//` do końca wiersza. Znaki `//` pozwalają umieścić w programie komentarz jednowierszowy.
- Jeśli komentarz ma obejmować kilka wierszy, należy umieścić go pomiędzy znakami `/*` oraz `*/`, na przykład:

```
/* Autor: Jan Kowalski
   Data modyfikacji: 2.03.2001
   */
```

Kompilator ignoruje komentarze; zadaniem komentarza jest bowiem wyjaśnienie programu człowiekowi.

1.1.2 Dyrektywa `#include`

W języku C++ (za językiem C) przyjęto, że wprowadzanie danych i wyprowadzanie wyników będą realizowane za pomocą standardowej biblioteki. Biblioteka ta jest dołączana podczas kompilacji programu. W programie należy umieścić dyrektywę `#include`, która określa tę część biblioteki standardowej, z której będziemy korzystać:

```
#include <nazwa_pliku_nagłówkowego_do_włączenia>
```

Dla wejścia-wyjścia w stylu C++ będzie to plik nagłówkowy `iostream`.

1.1.3 Dyrektywa `using namespace`

- Każda nazwa w C++ musi być zadeklarowana w określonym obszarze programu, na przykład w funkcji.
- Deklaracja dostarcza kompilatorowi informacji jak ze zmiennej będzie się korzystało.
- Przestrzeń nazw jest to specjalny obszar w programie służący tylko do deklarowania zmiennych. Chcąc skorzystać z takiej zmiennej, trzeba wskazać obszar, z którego ona pochodzi.
- Nazwy wykorzystywane przez standardowe składniki kompilatora, na przykład biblioteki standardowe, umieszczane są w przestrzeni nazw o nazwie `std`.

Dyrektywa `using namespace <nazwa_przestrzeni_nazw>` udostępnia wszystkie nazwy z tej przestrzeni.

Funkcja `main`

- Funkcja jest to element składowy programu, który ma swoją nazwę i który może być wywoływany czyli uruchamiany (ang. *call*) z innej części programu. Każdy program w języku C++ posiada funkcję o nazwie `main`. Od niej rozpoczyna się wykonywanie programu. Funkcja składa się z instrukcji, z których każda jest zakończona średnikiem.
- Zapis `int main()` oznacza, że funkcja po wykonaniu zwraca wartość całkowitą (ang. *integer*), na podstawie której system operacyjny może stwierdzić, czy program zakończył się prawidłowo. Nawiasy `()` sygnalizują kompilatorowi, że nazwa `main` jest funkcją.
- Nawiasy klamrowe `{}` oznaczają w języku C++ jakąś jednostkę. W tym przypadku oznaczają, że wszystkie instrukcje w nich zawarte należą do funkcji `main`.

Korzystanie ze standardowej biblioteki do wyprowadzania wyników

- Instrukcja, która wyświetla tekst na ekranie ma postać:

```
cout << "Witam" << endl;
```

- Nazwa `cout` oznacza standardowy strumień wyjściowy (ang. *standard output stream*), który domyślnie jest powiązany z ekranem. Operator `<<` oznacza przesyłanie do strumienia wyjściowego. W tym przypadku przesyłany jest tekst oraz manipulator `endl`. Wysłanie manipulatora do strumienia wyjściowego oznacza dokonanie jakiegoś przekształcenia, w przypadku `endl` jest to przejście do następnego wiersza.

Instrukcja `return`

- Instrukcja `return` kończy wykonywanie funkcji i przekazuje określoną wartość do programu (lub innej funkcji), z którego funkcja ta została wywołana.
- Typ zwracanej wartości musi odpowiadać typowi, który został podany w nagłówku funkcji. Funkcja `main()` zwraca wartość systemowi operacyjnemu. Przyjmuje się, że 0 oznacza prawidłowe zakończenie programu.

1.2 Program w języku C

- Język C++ stanowi nadzbiór języka C. Większość prostych programów w języku C można bez problemu kompilować jako programy w języku C++.

```
/* mój pierwszy program w C                                komentarz
mogę go również skompilować jako program C++ */

#include <stdio.h>                                           dyrektywa preprocesora
int main()                                                  funkcja main
{
    printf("Witam\n");                                     instrukcje funkcji main
    return 0;
}
```

Komentarz

W języku C dostępny jest tylko komentarz `/* ... */`.

Dyrektywa `#include`

- Aby móc wyprowadzać wyniki na ekran, należy dołączyć plik nagłówkowy `stdio.h`.

Uwaga: Nazwy plików nagłówkowych bibliotek standardowych w języku C ANSI należy podawać w pełnej postaci, wraz z rozszerzeniem.

Funkcja `main`

- W języku C wykonywanie programu również rozpoczyna się od funkcji `main`.

Korzystanie ze standardowej biblioteki do wyprowadzania wyników

- Instrukcja, która wyświetla tekst na ekranie ma postać:

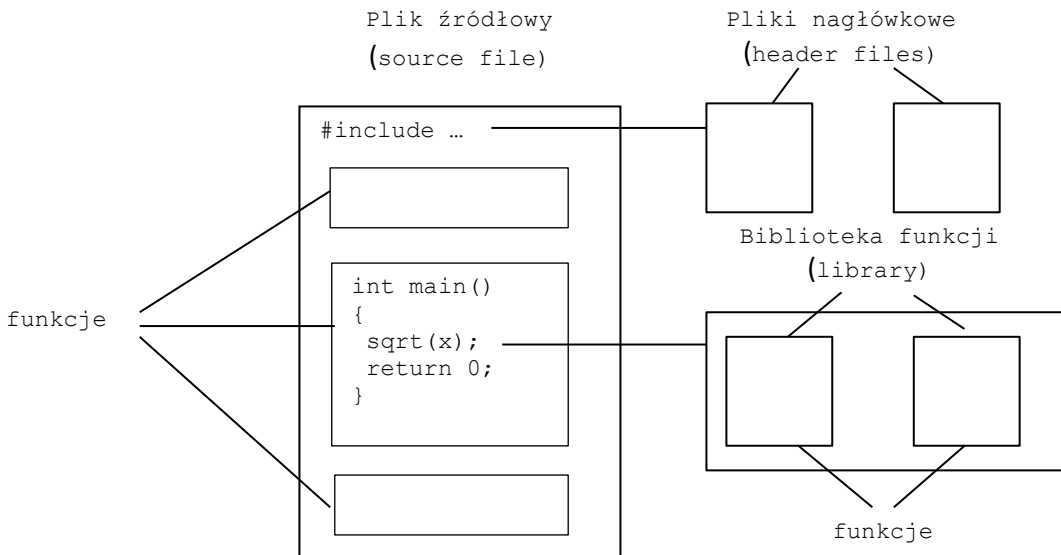
```
printf("Witam\n");
```

Wyprowadzanie wyników w języku C również realizowane jest za pomocą funkcji. W przykładzie użyta została funkcja `printf()`, której jako argument przesłano napis do wyświetlenia. Napis jest zakończony sekwencją `\n`, co oznacza, że po wyświetleniu tekstu należy przejść do następnego wiersza.

Instrukcja `return`

W języku C instrukcja `return` również kończy wykonywanie funkcji i przekazuje określoną wartość (w tym przypadku 0) do programu (lub innej funkcji), z którego funkcja ta została wywołana.

1.3 Struktura programu w języku C++



- Program w języku C++ składa się z jednego lub wielu plików, w każdym pliku znajduje się jedna lub wiele funkcji.
- Zawsze musi zawierać funkcję o nazwie `main()`, od której rozpoczyna się wykonywanie programu
- może zawierać odwołania do plików nagłówkowych i funkcji bibliotecznych
- nazwy potrzebnych plików nagłówkowych wskazywane są za pomocą dyrektywy `#include`
- nazwy bibliotek wskazywane są za pomocą parametrów kompilacji

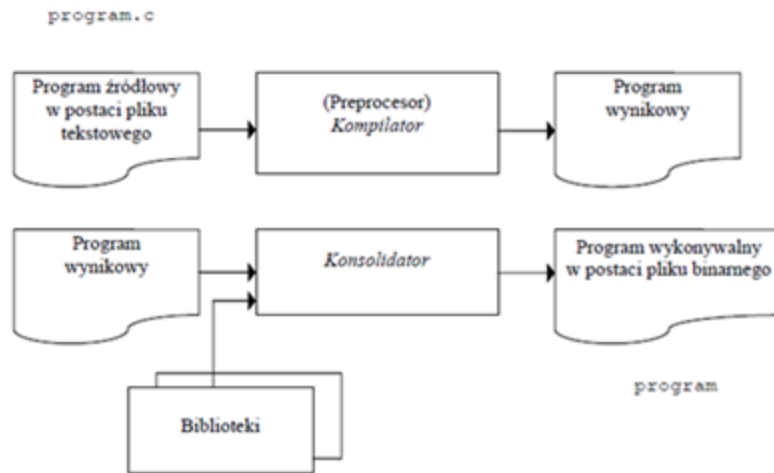
Pliki nagłówkowe

- zawierają deklaracje i definicje dla wszystkich funkcji, z których korzysta program i które nie znajdują się bezpośrednio w podstawowym pliku z tekstem programu
- są to pliki tekstowe

Biblioteki funkcji

- zawierają funkcje włączane do programu podczas konsolidowania (linkowania),
- każda biblioteka ma swój plik nagłówkowy (jeden lub wiele),
- są dostarczane przez producenta kompilatora (np. biblioteka run-time - czasu wykonania) lub tworzone przez użytkownika,
- są to pliki binarne,
- mogą mieć różne rozszerzenia, na przykład `.lib`.

1.4 Kompilacja programu



- Program źródłowy (ang. *source code*): dający się odczytać tekst programu.
- Kod wynikowy (ang. *object code*): przekład tekstu źródłowego programu na język komputerowy.
- Program wykonywalny (ang. *executable program*): program binarny gotowy do wykonania
- Kompilator (ang. *compiler*): program, który wczytuje cały tekst źródłowego i przekłada go na język komputerowy
- Konsolidator (ang. *linker*): program, który łączy oddzielnie skompilowane funkcje w jeden program; wiąże on kod wynikowy z funkcjami biblioteki C++
- Biblioteka (ang. *library*): plik zawierający funkcje standardowe, z których można korzystać w programie. Podczas wywołania takiej funkcji kompilator zapamiętuje jej nazwę, zaś konsolidator łączy kod wynikowy tekstu źródłowego z kodem istniejącym w bibliotece.

Przykład kompilacji w środowisku Linuks:

```
g++ -Wall program.cpp -o program
```