

Podstawy języka C++, część pierwsza

Poznanie języka C++ rozpoczniemy od przeanalizowania prostego programu (plik *hello.cpp*):

```
/*  
  
    Wszystko, co tu jest, to komentarz.  
  
*/  
  
#include<iostream>  
using std::cout;  
  
// Tekst funkcji głównej - to też jest komentarz (ale tylko do końca linii)  
  
int main(int argc, char *argv[])  
{  
    cout<<"Cześć, użytkowniku!!!\n";  
}
```

Treść pliku rozpoczyna się od bloku tekstu ograniczonego za pomocą znaczników: `/*` na początku, oraz `*/` na końcu. Jest to jeden ze sposobów wprowadzenia komentarza – części tekstu programu, która nie programuje żadnej akcji komputera, a ma za zadanie wyjaśnić czytającemu np. znaczenie treści fragmentu programu. Innym sposobem wprowadzenia komentarza jest wprowadzenie znacznika `//` - taki komentarz jednak rozciąga się tylko do końca bieżącej linii. Następnym elementem programu jest linia

```
#include<iostream>
```

Nakazuje ona na dołączenie tzw. pliku bibliotecznego, tutaj jest to plik biblioteki standardowej języka C++ o nazwie *iostream*. W tym pliku są zdefiniowane komponenty języka, realizujące operacje wejścia – wyjścia.

Każdy program w C++ musi składać się z co najmniej jednej funkcji – musi zawierać funkcję o nazwie *main*. W nagłówku tej funkcji widzimy słowa:

1. *int* – które mówi, że funkcja zwraca wartość, będącą liczbą całkowitą
2. *main* – nazwę funkcji,

a następnie parę nawiasów, w której znajdują się dwie deklaracje argumentów, szczegóły deklarowania argumentów zostaną omówione później.

Po nagłówku funkcji *main* znajduje się blok tekstu, zamknięty w nawiasy klamrowe `{ }`. Wewnątrz nawiasów zawarta jest treść funkcji, zwana często ciałem funkcji. W omawianym programie wykorzystany został tzw. strumień wyjściowy, dostępny poprzez obiekt o nazwie *cout*, który jest udostępniany przez plik biblioteczny *iostream*. Symbole, udostępniane przez bibliotekę języka podzielone są na *przestrzenie nazw*. Aby skorzystać z nazwy *cout*, należy napisać *std::cout*, co oznacza kwalifikację, że nazwa *cout* należy do przestrzeni nazw *std*, zawierającej standardowe symbole języka. Aby nie być zmuszonym do kwalifikacji przestrzeni nazw za każdym razem, można zadeklarować w tekście programu, że pisząc *cout* mamy na myśli *std::cout*. Do tego celu służy deklaracja *using std::cout;*, zawarta w drugiej linii programu. Program ma za zadanie wypisać tekst, zawarty w cudzysłowach, na ekran. Wypisywanie na ekran zrealizowane jest poprzez wysłanie operatorem `<<` napisu zamkniętego w znaki cudzysłowu – jest to tzw. *stała tekstowa* (tekst o znanej z góry treści). W standardowych warunkach nasz program wypisze na ekranie tekst:

```
Cześć, użytkowniku!!!
```

a następnie przesunie kursor do następnego wiersza (w tym celu do tekstu jest wstawiona kombinacja znaków `\n` – jest to jedna z dostępnych w języku tzw. *escape sekwencji*). Ponieważ po zrealizowaniu instrukcji wypisywania na ekran natrafiamy na klamrę zamykającą funkcję *main*, kończy się funkcja główna programu. Zakończenie działania funkcji *main* oznacza zakończenie działania programu.

Zadanie:

Przygotować i uruchomić omówiony program dla systemu operacyjnego Linux

1. Wpisać treść programu do pliku o nazwie *hello.cpp*
2. Skompilować program, wydając komendę `g++ -o hello hello.cpp`
3. Uruchomić program wydając komendę `./hello`

Przeanalizujmy kolejny program (plik *wiek.cpp*):

```
#include<iostream>
using namespace std;

int main(int argc, char *argv[])
{
// deklaracje zmiennych
int wiek;           // zmienna typu "liczba całkowita"
char imie[100];    // zmienna typu "łańcuch tekstowy o długości 100 znaków"

    cout<<"Podaj swoje imię :";
    cin>>imie;

    cout<<"Podaj swój wiek :";
    cin>>wiek;

    cout<<"Informacje o Tobie:"<<endl;
    cout<<"Twoje imię: " << imie << endl;
    cout<<"Twój wiek: " << wiek << " lat" << endl;

// instrukcja warunkowa
if(wiek>=18) cout<< "Jesteś pełnoletni" << endl;
else
    cout << "Jesteś niepełnoletni"<< endl;
}
```

Poprzedni program nie używał żadnych zmiennych. Tutaj pojawiają się zmienne – obiekty, mające na celu przechowywanie informacji. W języku C++ zmienną deklaruje się, podając najpierw jej typ, a następnie wybrany dla niej *identyfikator* – nazwę, poprzez którą będziemy się odwoływać do zmiennej. Na przykład deklaracja:

```
int wiek;
```

oznacza, że deklarujemy zmienną o nazwie *wiek*, mającą typ *int*. Podstawowe typy danych w języku C++ to:

- *int* – typ przechowujący liczby całkowite (np. 0, 10, -2)
- *char* – typ znakowy (zmienna przechowuje pojedynczy znak – element tekstu, np. 'a', 'Z', '\n')
- *float* – typ zmiennoprzecinkowy (np. 1.2, 0.1, 1.5e3)
- *double* – typ zmiennoprzecinkowy o „podwójnej precyzji” (dokładniejszy)

W programie pojawia się deklaracja tzw. łańcucha tekstowego. Jest to zmienna przechowująca zadaną liczbę znaków, podaną w nawiasie klamrowym.

UWAGA!

Należy pamiętać, że w języku C++ łańcuch tekstowy kończy się zawsze znakiem o kodzie 0 (tzw. *NULL*), oznaczającym koniec napisu, przez co w łańcuchu może znaleźć się nie więcej użytecznych znaków niż ilość, podana w nawiasie klamrowym, zmniejszona o jeden. W naszym przykładzie program może pobrać od użytkownika imię o długości, wynoszącej maksymalnie 99 znaków.

Kolejnym nowym elementem języka jest wczytywanie danych ze strumienia wejściowego. Zadanie to realizuje instrukcja:

```
cin>>imie;
```

wczytująca dane wpisane z klawiatury (dokładniej: ze strumienia wejściowego) do zmiennej *imie*, akcję wczytywania programuje *operator >>*, stojący po prawej stronie obiektu *cin*, udostępniającego strumień wejściowy.

Instrukcja warunkowa *if* postaci ***if(warunek) instrukcja1; else instrukcja2;*** działa następująco:

- sprawdzane jest, czy warunek jest prawdziwy. Na przykład wyrażenie: *wiek* >= 18 jest prawdziwe, jeśli zmienna *wiek* zawiera np. liczbę 23, zaś fałszywe, jeśli zmienna *wiek* zawiera liczbę mniejszą od osiemnastu.
- jeśli warunek jest prawdziwy, wykonywana jest ***instrukcja1***
- jeśli istnieje część ***else instrukcja2;*** (jest to część opcjonalna, czyli nieobowiązkowa) i warunek *nie był* prawdziwy, wykonywana jest ***instrukcja2***

Program zawiera ponadto przykłady trzech typowych fraz, zawartych w programach w języku C++:

1. można „wstawić” do strumienia wyjściowego wartość *endl*, powodującą przejście kursora do nowego wiersza (takie działanie powoduje również dodanie *\n* do wypisywanego łańcucha tekstu)
2. „kaskadowe” użycie strumienia wyjściowego w wyrażeniu wypisywania na ekran: instrukcja ***cout<<"Twój wiek: " << wiek << " lat" << endl;*** spowoduje: wypisanie tekstu *Twój wiek:*, następnie wartości zmiennej *wiek*, potem napisu *lat*, i na koniec przesunie kursor do następnego wiersza.
3. użycie deklaracji *using std::cout* można zastąpić wygodniejszą dyrektywą użycia *using namespace std;* W programie użyte zostały nazwy *cout*, *cin* oraz *endl*, należące do przestrzeni nazw *std*. Aby móc korzystać z każdej z nich, należałoby utworzyć trzy deklaracje użycia lub za każdym razem pisać pełną nazwę z kwalifikacją przestrzeni nazw np. *std::endl*. Zamiast tego możemy poinformować kompilator, że chcemy, aby udostępnione zostały wszystkie nazwy z przestrzeni nazw *std*, do jakich uzyskaliśmy dostęp np. poprzez włączenie do programu plików bibliotecznych. W tym celu stosuje się dyrektywę użycia przestrzeni nazw *using namespace*.

Zadanie:

Proszę uruchomić omówiony program. Następnie na jego przykładzie napisać program, który wypisze w formie estetycznej tabeli imię, nazwisko, wiek osoby uruchamiającej program, oraz sprawdzi na podstawie wieku oraz wypisze na ekran, że użytkownik jest uczniem (wiek 7-13), gimnazjalistą (wiek 14-16), licealistą (wiek 17-19), lub że nie mieści się w tych kategoriach wiekowych. Do tego celu konieczne będzie utworzenie warunku złożonego z dwóch części.

Dla sprawdzenia, czy użytkownik jest dzieckiem (przedział lat 0-6) musimy sprawdzić, czy:

- wiek jest większy lub równy od 0 lat (zmienne typu *int* mogą być ujemne, a dowcipnisiów nie brakuje...) **i czy jednocześnie**
- wiek jest mniejszy niż 7 lat

Warunek ten zapisuje się następująco: ***wiek* >= 0 && *wiek* < 7**, przy czym operator ***&&*** oznacza logiczne I (logiczną koniunkcję). Innym operatorem tego typu jest logiczne LUB (logiczna

alternatywa), oznaczany w językach C++ i C za pomocą `||` (dwóch kresek pionowych).

Do rozwiązania zadania należy jeszcze zauważyć, że instrukcją stojącą po słowie kluczowym *else* może być kolejna instrukcja warunkowa *if(warunek) instr1; else instr2;*

```
if(wiek>100) cout << "Więcej niż 100 lat";  
else // nie więcej niż 100 lat  
if(wiek > 50) cout << „Więcej niż 50 lat (choć nie więcej niż 100)“;
```

Na następne zajęcia proszę zapoznać się z rozdziałami 2, 3 oraz 4 z książki „C++ dla każdego”, autorstwa Jesse Liberty.