AKADEMIA GÓRNICZO-HUTNICZA
IM. STANISŁAWA STASZICA W KRAKOWIE

Podstawy C++, część 1

Szymon Buchaniec

http://home.agh.edu.pl/~buchanie/

Akademia Górniczo-Hutnicza, Katedra Podstawowych Problemów Enegetyki

12 grudnia 2018

Pierwszy program w c++

main.cpp

```
#include<iostream>
// Bibioteka I/O

int main() //Miejsce startu programu
{
    std::cout<<"Hello World\n";
    //Wypisanie na ekran Hello World
    return 0;
}
```

Pierwszy program w c++

Kompilacja

Do kompilacji naszego programu wykorzystamy program **g++**. Wykona on za nas wszystkie etapy potrzebne aby komputer mógł wykonac nasz program. Wynikiem działania g++ będzie nowy plik wykonywalny **a.out**.

```
szymon@szymon-GV62-8RC:~/test$ ls
main.cpp
szymon@szymon-GV62-8RC:~/test$ g++ main.cpp
szymon@szymon-GV62-8RC:~/test$ ls
a.out  main.cpp
szymon@szymon-GV62-8RC:~/test$ ./a.out
Hello World
```

Zmienne

Wewnątrz programu możemy używać zmiennych różnego typu. Zmienne reprezentują wartości i służą nam do opisania pewnej abstrakcji. Np. chcemy opisać ile mamy pieniędzy na koncie (zmienne przechowujące wartości liczbowe), czy drzwi w domu są otwarte (zmienne przechowujące wartości logiczne - prawda/fałsz) albo jak nazywa się nasza ulica (zmienne przechowujące napisy). Prykłady:

- int apples; - typ całkowity
- bool open = true; - typ logiczny (true/false)
- std::string name = "Szymon"; - typ napisowy

Typy danych

W języku c++ mamy dostęp do podstawowych typów danych:

- int - liczby całkowite, np. 2
- bool - wartości logiczne, np. false
- float - rzeczywiste o niskiej precyzji
- double - rzeczywitste o wysokiej precyzji, np. 2.1
- void - brak typu, nie można utworzyć takiej zmiennej
- char - pojedynczy znak, wewnątrz ' ', np 'i'.

Zmienne

Przykłady

Przykłady **definicji** (a zarazem **deklaracji**) zmiennych różnych typów:

- int count = 5; count = count-1;
- bool open = true; open = !open;
- double pi = 3.14;
- char znak = 'a'; znak = 'b';

Napisy oraz string

W języku c++ istnieje specjalny typ danych tekstowych (napisowych). Napisy oznaczamy poprzez znaki " ", wpisując wewnątrz nich tekst. Są to stałe napisowe. Do przechowywania oraz operacji na napisach służy dodatkowy typ danych string, zawarty w bibliotece standardowej string. Np. std::string name = "Ewa";. Możemy dodawać do siebie obiekty typu string za pomocą operatora +. Np. std::string a = "Ala", b = 'Kot'; std::string poemat = a + " i " + b;.

Stałe

W programie możemy używać wartości stałych. Są to wartości wpisane bezpośrednio w kodzie - tzw. literały, 5, 'z', "Napis" lub zadeklarowane jako const. Program nie może zmieniać ich wartości.

- const int apples = 3; - apples - stała typu całkowitego, 3 - literał
- const double PI = 3.14; - muszą być od razu zainicjalizowane

Operacje

Operacje arytmetyczne dla typu liczbowego

Operatory arytmetyczne służą do tworzenia wyrażeń zwracających wartość typu liczbowego. Załóżmy, że w programie mamy zdefiniowane 4 zmienne: double a; double b; oraz int c, d;.
Dostępne operacje arytmetyczne:

- a+b – dodawanie
- c-d – odejmowanie
- a*b – mnożenie
- a/b – dzielenie a przez b
- -a – liczba przeciwna do a
- d%c – operacja modulo - reszta z dzielenia a przez c, dostępna tylko dla liczb całkowitych

Bardziej skomplikowane operacje można znaleźć w bibliotece cmath. Zachowane są reguły kolejności wykonywania działań, można grupować wyrażenia w nawiasach (), np. (a+b)*(c+a).

Operacje

Operacje arytmetyczne dla typu liczbowego - przykład

```
int a, c;
double b;
a = 5;
c = 2;
b = 2.0;

std::cout<<"Dodawanie; "<<a+b<<"\n";
std::cout<<"Odejmowanie: "<<a-b<<"\n";
std::cout<<"Dzielenie: "<<a/b<<"\n";
std::cout<<"Mnozenie: "<<a*b<<"\n";
std::cout<<"Dzielenie z reszta: "<<a%c<<"\n";
std::cout<<"Dzielenie calkowitych: "<<a/c<<"\n";
```

Operacje

Operacje arytmetyczne dla typu liczbowego - przykład

```
szymon@szymon-GV62-8RC:~/test$ ./a.out
Dodawanie; 7
Odejmowanie: 3
Dzielenie:  2.5
Mnożenie: 10
Dzielenie z resztą: 1
Dzielenie liczb całkowitych: 2
```

Operacje

Operacje relacyjne dla typu liczbowego

Operacje relacyjne służą do tworzenia wyrażeń logicznych, porównujących dwie liczby i zwracających wartość typu bool. Załóżmy, że w programie mamy zdefiniowane 2 zmienne: int a; oraz int b;.

- a == b – liczby równe, UWAGA: nie a = b
- a != b – liczby różne
- a >= b – a większe lub równe b
- a > b – a większe od b

Analogicznie do >= oraz > mamy operatory <= i <.

Zmienne

Inkrementacja i dekrementacja

W języku c++ mamy dostępne przydatne dodatkowe operatory jednoargumentowe służące do inkrementacji liczby całkowitej oraz dekrementacji liczby całkowitej. Są to operatory **++** oraz --. Pierwszy z nich zwiększa wartość zmiennej o jeden, drugi zminiejsza. Dla przykładu **a++** ustawi wartość **a** tak samo jak przy przypisaniu **a = a+1;**.
UWAGA: Co się stanie w przypadku przypisania **b = a++**?

Zmienne

Operacje dla typów logicznych

Tworzą wyrażenia o typie logicznym. Dwie zmienne: bool p; oraz bool q;.

- !p – negacja
- p && q – logiczne "i"
- p || q – logiczne "lub"

Biblioteki

Biblioteki zawierają w sobie przygotowane wcześniej funkcjonalności, które możemy wykorzystać w programie. Biblioteki wczytujemy do programu na samym początku pliku za pomocą instrukcji #include <nazwa_biblioteki> dla każdej wczytywanej biblioteki. W naszym przykładzie wczytaliśmy bibliotekę iostream odpowiedzialną za operacje wejścia/wyjścia (input/output - I/O). Dzięki niej mieliśmy dostęp do instrukcji **std::cout**<<

Biblioteki

Przydatne biblioteki

- iostream - komunikacja z użytkownikiem - wypisywanie na ekran oraz wczytywanie wartości od użytkownika.
- fstream - wczytywanie i zapisywanie do pliku
- cmath - funkcje matematyczne (pow, abs, sin, sqrt ...)
- vector - kontener typu vector
- cassert - asercje - pomoc w trakcie debugowania programu
- string - napisowy typ danych

Operacje wejścia/wyjścia

cin oraz cout

cin oraz **cout** są specjalnymi obiektami pozwalającymi na wczytywanie zmiennych z klawiatury (cin) oraz wypisywania na ekran (cout). Zdefiniowane w bibliotece iostream.

1. **cout** - używamy przekazując do tego obiektu zmienne lub stałe za pomocą operatorów <<. Zostaną one wypisane na ekran. Możemy oddzielać różne stałe oraz zmienne poprzez dodanie kolejnych operatorów << np.
   std::cout<<a<<" "<<b<<" i jeszcze trochę tekstu";
   Po << mogą występować podstawowe typy danych oraz napisy.
2. **cin** - Służy do wprowadzania danych przez uzytkownika. Potrzebujemy zadeklarowanych zmiennych które przechowają wczytaną wartość. Używamy z pomocą operatora >>. Np. **std::cin>>a;**.