

Podstawy C++, część 3

Szymon Buchaniec

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

Akademia Górniczo-Hutnicza, Katedra Podstawowych Problemów Energetyki

9 stycznia 2019

Funkcje służą do wyodrębnienia pewnej funkcjonalności, w celu poprawy uniwersalności, testowania, zrozumienia i łatwości modyfikacji kodu. Prosty, często używany przykład: programowanie możemy rozumieć jako podanie przepisu komputerowi co powinien zrobić, tak samo jak przepisy kulinarne. Z oczywistych powodów w przepisie kulinarnym nie opisuje się dokładnie wszystkich czynności, np. dostajemy instrukcję, aby coś zmiksować mikserem lub usmarzyć na patelni. Nie jest podawany dokładny opis tego jak się miksuje, ani też jak smażyć, co najwyżej pewne “parametry” - czas, moc, ilość. My sami potrafimy to zrobić, mając te “funkcjonalności” w głowie. Dokładnie tak samo jest z programowaniem, pewne podstawowe funkcjonalności są wyodrębnione i tylko wywoływane w odpowiednim momencie.

```
void printHello()  
{  
    std::cout<<"Hello!";  
}
```

Tego typu prosta funkcja zawsze wykona dokładnie to samo działanie - wypisze na ekran napis "Hello". Możemy też zrobić coś bardziej skomplikowanego:

```
bool doYouLike(int a)
{
    if ( a%2 == 0)
    {
        return true;
    }
    else
    {
        return false;
    }
}
```

W tym przypadku funkcja “powie” czy lubi zmienną a.

```
int main()  
{  
    if (doYouLike(5))  
    {  
        printHello();  
    }  
    else  
    {  
        while(true) printHello();  
    }  
}
```

Przykład wykorzystania zdefiniowanych funkcji. Funkcje muszą być znane przed wywołaniem.

```
typ_zwracany nazwa_funkcji(argumenty_funkcji)
{
    //ciało funkcji
}
```

typ_zwracany - funkcje mogą zwracać wartości. Wartości te zostaną wstawione w miejsce wywołania funkcji w trakcie działania programu. Jeżeli funkcja nic nie zwraca, wpisujemy **void**.

argumenty_funkcji - możemy podać argumenty funkcji (typ oraz nazwa), które będą wymagane do jej wywołania.



Zapisywanie do plików w C++ jest bardzo podobne do korzystania z `std::cout`. Różnica polega na tym, że musimy najpierw otworzyć plik, do którego chcemy zapisać dane. Aby zapisywać do pliku potrzebujemy zmienną, która będzie go reprezentować (tzw. uchwyt pliku). Typ tej zmiennej - `ofstream` - jest dostępny w bibliotece `fstream`. Otwieranie pliku domyślnie powoduje jego utworzenie. Jeżeli plik istnieje, dane w nim zostaną nadpisane.



```
#include <fstream>
int main()
{
    int a = 5;
    //plik - uchwyt do pliku
    std::ofstream plik("file.txt");
    plik<<a<<" "<<1<<"\n";
    plik<<2*a<<" "<<2<<"\n";
    plik.close();
    plik.open("file2.dat");
    plik<<"OK";
    plik.close();
}
```