

Sam dla siebie C++  
każdy programować może

sobota, dnia 01.11. 2025 r.

## *Funkcje w C++ nr 010/25*

### *część 1*

#### **Funkcje**

W programach utworzonym w języku C++, znajdują miejsce jako jedno z przodujących elementów – funkcje. Funkcje są to fragmenty kodu, umożliwiające utworzenie, zbudowanie, zaprogramowanie programu, w którym nie będzie potrzeby w czasie tworzenia tego programu, wielokrotnie zapisywać te same linijki w celu wykonania określonych operacji.

Funkcje umożliwiają podzielenie kodu programu na mniejsze programy, często nazywane programikami, i zarządzanie nimi w logiczny sposób.

Funkcje również umożliwiają, wręcz ułatwiają dokonania korekty w utworzonym programie bez jakiegokolwiek otwierania całego kodu w celu usunięcia usterki. Bowiem zdarzyć się może, że w czasie kompilacji tworzonego programu, kompilator będzie działać prawidłowo, a po wprowadzeniu do programu przed oddaniem do użytku, program, po wprowadzeniu danych, nie działa, lub się zawiesza. I pojawia się problem. Pytanie jaki? Program utworzony bez funkcji, wielokrotne powtarzanie tych samych linijek np. po 30 linijek w programie w którym jest około ponad 10000 linijek jest niemożliwe. Bo jak sprawdzić 10000 w ciągu jednego dnia?

Więc między innymi po to są właśnie funkcje - że jak coś nie zadziała prawidłowo, lub wadliwie, albo pomylił bym znaki plus na podzielić, to już całkiem problem – miałbym wynik niespodziewany wręcz nie taki co być powinien – abym wadę mógł usunąć szybko, sprawnie, bez jakichkolwiek trudności.

Bowiem funkcja – to nic innego jak zapisana część programu w kodzie wykonująca zapisane operacje w czasie jego użytkowania. Stąd też każda funkcja ma swoją nazwę i swoje argumenty, jak również ma prostą budowę.

#### **Ogólna budowa funkcji:**

##### Schemat 1.

```

typ nazwaFunkcji()
{
    ..... // instrukcja
    ..... // instrukcja
    ..... // instrukcja
    ..... // instrukcja
    return wartoscTypu Typ

```

}

Schemat 2.

```

    .typ nazwaFunkcji( ListaArgumentow )
    {
        ..... // instrukcja
        ..... // instrukcja
        ..... // instrukcja
        ..... // instrukcja
        return wartoscTypu Typ
    }

```

Powyższe przykłady wskazują na funkcje które coś zwracają.

Ale są też funkcje które utworzyć można żeby nic nie zwracały.

Schemat 1.

```

typ nazwaFunkcji()
{
    return wartoscTypu Typ
}

```

Schemat 2.

```

typ nazwaFunkcji( ListaArgumentow )
{
    return wartoscTypu Typ
}

```

## Definicja funkcji

Funkcja w C++ to blok kodu wykonujący w czasie użytkownika przez użytkownika swoje zadanie, pozwalający na przystosowanie go do wielokrotnego logicznego działania, unikania wielokrotnego

powtarzania w czasie projektowania programu tych samych linijek kodu, oraz może przyjmować dane wejściowe – parametry – na których wykonuje zadanie a także może zwracać wynik swojej pracy.

Składnia definicji:

```

typ_zwracany nazwa_funkcji( lista_parametrów)
{
    // ciało funkcji, tj. instrukcje do wykonania
    // opcjonalnie: instrukcja return
}

```

Ponadto proszę pamiętać i nie mylić definicji z deklaracją.

Deklaracja funkcji – jest to prototyp funkcji informujący kompilator o swoim istnieniu, nazwie, typie zwracanym, parametrach i nie posiada ciała, oraz kończy się średnikiem ( ; ).

Definicja funkcji – zawiera rzeczywistą implementację (ciało) funkcji i przydziela dla niej pamięć.

Funkcja może być zadeklarowana wielokrotnie, ale zdefiniowana tylko raz.

Podsumowując:

cecha	deklaracja funkcji	definicja funkcji
cel	informuje kompilator o istnieniu funkcji, jej typie i parametrach	dostarcza implementację ( ciało ) funkcji
składnia	kończy się średnikiem	zawiera kod funkcji w nawiasach klamrowych { }
pamięć	nie przydziela pamięci dla ciała funkcji	przydziela pamięć dla funkcji

Innymi słowy dla przykładu::

deklaracja:

```
void mojaFunkcja( int a);
```

podczas gdy definicja zawiera ciało funkcji – kod - w nawiasach klamrowych

```
void mojaFunkcja( int a) { /* ciało funkcji – kod funkcji }
```

Ćwiczenie przykładowe funkcji podaję na oddzielnym pliku: *Funkcje w C++ nr 010/25 - część 2*

**Tak dla przypomnienia krótka uwaga.:**

Niniejsze opracowanie, jak i następne opracowania celem nauki własnej którą kontynuuję, opieram na lekturach takich jak:

1. Język C++ Szkoła programowania Stephen Prata
2. Programowanie w języku C++ Wiesław Porębski
3. Podstawy języka C++ Stanley B. Lippman
4. Symfonia C++ standard Jerzy Grębosz
5. PRZEWODNIK DLA POCZĄTKUJĄCYCH C++ ALEX ALLAIN
6. Zadania z programowania z przykładowymi rozwiązaniami Mirosław J. Kubiak
7. i internet