

Laboratorium nr 7

Temat: Funkcje cz.1.

Zakres laboratorium:

- deklaracja i definicja funkcji
- wywołanie funkcji
- funkcje **inline**
- pliki nagłówkowe biblioteki standardowej
- funkcje biblioteki standardowej C++
- zadania laboratoryjne

Deklaracja i definicja funkcji

Deklaracja funkcji (prototyp funkcji) – informuje kompilator jaką wartość funkcja będzie zwracała oraz liczbę i typ argumentów (w kolejności ich oczekiwanego pojawienia się)

```
typ_funkcji nazwa_funkcji(lista_typów_argumentów);
```

np.

```
int dodaj (int a, int b);           //deklaracja
```

Definicja funkcji – definicja jest deklaracją, w której przedstawiona jest treść funkcji (inaczej ciało funkcji)

np.

```
int dodaj (int a, int b)           //definicja  
{  
    return a+b;  
}
```

Wywołanie funkcji

Wywołanie funkcji – to napisanie nazwy funkcji z podaniem argumentów funkcji w nawiasie okrągłym () (jeśli funkcja takie argumenty posiada)

np.

```
int liczba1=5,liczba2=10,wynik;  
wynik=dodaj(liczba1,liczba2);           //wywołanie funkcji
```

```
...  
...  
...
```

```
int dodaj(int a,int b)                 //definicja funkcji  
{  
    return a+b;  
}
```

Argumenty formalne – to argumenty zdefiniowane w prototypie funkcji,
Argumenty aktualne – to argumenty użyte w wywołaniu funkcji.

Funkcje **inline**

Wywołania funkcji wymagają pewnego czasu wykonywania. Czas ten można zredukować poprzez zdefiniowanie funkcji jako **inline**, zwłaszcza dla małych, niedługich funkcji. Kwalifikator ten „doradza” kompilatorowi wygenerowanie kopii kodu funkcji w miejscach jej wywołań, aby uniknąć jej wywołania. Kompilator może ignorować kwalifikator **inline**, zwłaszcza dla większych funkcji, dlatego powinien być on używany tylko z małymi, często używanymi funkcjami.

np.

```
int liczba1=5,liczba2=10,wynik;  
wynik=dodaj(liczba1,liczba2); //wywołanie funkcji
```

...

...

...

```
inline int dodaj(int a,int b) //definicja funkcji inline  
{ return a+b; }
```

Pliki nagłówkowe biblioteki standardowej

Każda biblioteka standardowa zawiera odpowiedni **plik nagłówkowy** zawierający prototypy funkcji dla wszystkich funkcji w tej bibliotece i definicje różnych typów danych oraz stałych niezbędnych dla tych funkcji.

Programista może tworzyć **własne pliki nagłówkowe**, które powinny kończyć się rozszerzeniem `.h`. Plik nagłówkowy zdefiniowany przez programistę jest dołączany – podobnie jak pliki nagłówkowe bibliotek standardowych – przez użycie dyrektywy preprocesora `#include`, np.

```
#include "kwadrat.h"
```

UWAGA: Najnowszy projekt standardu ANSI/ISO określa nowe nazwy dla wielu starszych plików nagłówkowych C++. Większość nowych nazw plików nagłówkowych nie kończy się już rozszerzeniem `.h`.

```
#include <iostream.h>           //plik nagłówkowy starego stylu  
#include <iostream>            //plik nagłówkowy nowego stylu
```

UWAGA: Wiele kompilatorów nie obsługuje jeszcze ostatniego projektu standardu C++.

Pliki nagłówkowe biblioteki standardowej „starego i nowego stylu”

„stary styl”	„nowy styl”
<code><assert.h></code>	<code><cassert></code>
<code><ctype.h></code>	<code><cctype></code>
<code><float.h></code>	<code><cfloat></code>
<code><limits.h></code>	<code><climits></code>
<code><math.h></code>	<code><cmath></code>
<code><stdio.h></code>	<code><cstdio></code>
<code><stdlib.h></code>	<code><cstdlib></code>
<code><string.h></code>	<code><cstring></code>
<code><time.h></code>	<code><ctime></code>
<code><iostream.h></code>	<code><iostream></code>
<code><iomanip.h></code>	<code><iomanip></code>
<code><fstream.h></code>	<code><fstream></code>

Funkcje biblioteki standardowej C++

Biblioteka standardowa C++ zawiera bogaty zbiór funkcji do przeprowadzania powszechnych operacji matematycznych, manipulacji napisami i znakami, funkcji wejścia/wyjścia, sprawdzania błędów i wielu innych użytecznych operacji. To ułatwia pracę programisty, ponieważ funkcje te dostarczają wiele pomocnych mu i potrzebnych metod i narzędzi.

Funkcje i klasy biblioteki standardowej są udostępniane jako część środowiska programistycznego C++ i zazwyczaj dostarczane przez dostawców kompilatorów. Wiele bibliotek klas specjalnego przeznaczenia jest dostarczanych przez niezależnych dostawców oprogramowania.

Większość programistów korzysta z bogatej kolekcji istniejących klas i funkcji w bibliotece standardowej C++. Nauczenie się „świata C++” składa się wtedy głównie do: 1) poznania samego języka i 2) używania klas i funkcji z biblioteki standardowej.

Programy są zazwyczaj tworzone przez łączenie nowych funkcji i klas, napisanych przez programistę, z „zawartymi w pakiecie” funkcjami i klasami w bibliotece standardowej C++ i w różnych innych bibliotekach klas.

Funkcje matematyczne z biblioteki standardowej

FUNKCJE TRYGONOMETRYCZNE:

cos(x) **sin(x)** **tan(x)** **ctan(x)**

x w radianach

FUNKCJE WYKŁADNICZE I LOGARYTMICZNE:

exp(x) *funkcja eksponentcjalna e^x*
log(x) *logarytm naturalny z x (podstawa e)*
log10(x) *logarytm dziesiętny z x (podstawa 10)*

POTĘGI I PIERWIASTKI:

pow(x,y) *potęgowanie x^y*
pow10(x) *potęgowanie 10^x*
sqrt(x) *pierwiastek kwadratowy*
cbrt(x) *pierwiastek sześcienny*

INNE FUNKCJE:

hypot(a,b) *długość przeciwprostokątnej trójkąta*
ceil(x) *zaokrąglenie liczby x w górę*
floor(x) *zaokrąglenie liczby x w dół*
fabs(x) *moduł x (wartość bezwzględna x)*
fmod(x,y) *reszta z dzielenia x/y (modulo)*

Wybrane i przydatne funkcje z biblioteki standardowej

W pliku nagłówkowym **<cstdlib>**

rand() generacja liczb całkowitych pseudolosowych
srand() generacja liczb całkowitych losowych

W pliku nagłówkowym **<ctime>**

time() zwraca bieżący czas kalendarzowy w sekundach

W pliku nagłówkowym **<cstring>**

strcpy() do kopiowania napisów
strcat() do łączenia napisów
strcmp() do porównywania napisów
strlen() do określania długości napisów
size() do określania liczby znaków w napisie

W pliku nagłówkowym **<cctype>**

islower() do określania czy argument jest małą literą
isupper() do określania czy argument jest wielką literą
tolower() do zamiany wielkiej litery na małą
toupper() do zamiany małej litery na wielką

Zadania laboratoryjne