

Функції в C/C++

Визначення (definition) функції включає наступні елементи (згідно затвердженому ANSI-стандарту мови C) :

- вказівку про можливість використання функції поза даним файлом (за допомогою одного з службових слів **static** або **extern**; останнє вживається за замовчуванням);
- тип результату, який повертає функція; відсутність результату позначається службовим словом **void**;
- ідентифікатор (ім'я) функції;
- кількість та типи *формальних параметрів* в круглих дужках; в разі їх відсутності вживається службове слово **void**;

важливо: всі параметри функції передаються за значенням

- тіло функції, яке взяте в фігурні дужки; складається з визначення внутрішніх (локальних) змінних функції та інструкцій, що мають виконуватись під час роботи функції. Якщо функція повертає результат, то серед цих інструкцій обов'язково має бути інструкція виду: **return <вираз>;**

причому тип виразу повинен збігатись з типом результату функції.

Відсутність результату функції також позначається службовим словом **void**;

Декларація функції в C/C++

Декларація (declaration, prototype) функції – це можливість зробити функцію видимою для компілятора. Декларація виглядає так, як і визначення функції, проте не містить тіла функції. В одному програмному файлі допустимо кілька декларації однієї функції і лише одне її визначення.

Приклад.

```
int strlen (char *); // декларація функції
int strlen (char *s) // визначення функції
{
    char *p = s;
    while (*p != '\0') p++;
    return p-s;
}
```

Зверніть увагу – в декларації функції ідентифікатори формальних параметрів необов'язкові.

Виклик (invocation) функції

Для виклику функції потрібно вказати її ідентифікатор та передати в дужках список фактичних параметрів в точній відповідності до списку формальних параметрів у визначенні та декларації функції. Якщо функція повертає результат, необхідно прийняти його присвоєнням результату виклику функції об'єкту відповідного типу.

Приклад.

```
int strlen (char *); // декларація функції
int main ()
{
    char * text = "Example";
    int len = strlen (text) // виклик функції
    cout << "Довжина тексту" << len << endl;
}
```

Класи пам'яті об'єктів мов C/C++

Класи пам'яті зовнішніх **глобальних** об'єктів (функцій та змінних)

extern

(за замовчуванням)

Доступ можливий з усіх файлів програми

static

Доступ можливий лише у файлі, де визначений об'єкт

Класи пам'яті внутрішніх **локальних** об'єктів (лише змінні)

auto

(за замовчуванням)

Доступ можливий лише у блоці, де визначений об'єкт

register

Компілятор намагатиметься розміщувати об'єкт на регістрах процесора

Важливі зауваження про класи пам'яті

1. Визначення змінної означає виділення під неї пам'яті, декларація змінної – це представлення її компілятору.
2. Всі змінні, визначені поза межами функцій, є глобальними (клас пам'яті **extern** за замовчуванням). Вони доступні у всіх функціях програми.
3. Всі змінні, визначені у функціях, а також параметри функцій, – локальні (клас пам'яті **auto** за замовчуванням). Доступні лише в тому блоці, де визначені.
4. Якщо біля глобальної змінної вказується службове слово **extern**, то це не визначення, а декларація змінної, яка буде визначена далі.
5. Локальна змінна з тим самим ідентифікатором “перекриває” глобальну змінну. Те саме стосується і локальних змінних різних рівнів.
6. Службове слово **register** біля локальної змінної – це лише побажання до системи розмістити її на регістрах процесора, щоб прискорити звертання до неї – компілятор в праві проігнорувати це прохання.

Деякі стандартні функції C/C++

Математичні функції містяться у файлі `<math.h>` (компілятори мови C) або у файлі `<cmath>` (компілятори мови C++). Аргументи **x** та **y** мають тип `double`, **n** – тип `int`; усі функції повертають значення типу `double`.

`sin (x)` – синус x ;

`cos (x)` – косинус x ;

`tan (x)` – тангенс x ;

`asin (x)` – арксинус x у діапазоні $[-\pi/2, \pi/2]$ ($x \in [-1, 1]$);

`acos (x)` – арккосинус x у діапазоні $[0, \pi]$ ($x \in [-1, 1]$);

`atan (x)` – арктангенс x у діапазоні $(-\pi/2, \pi/2)$;

`atan2 (y, x)` – арктангенс y/x у діапазоні $(-\pi/2, \pi/2)$;

`exp (x)` – e в степені x ;

`log (x)` – логарифм натуральний x ($x > 0$);

`log10 (x)` – логарифм десятковий x ($x > 0$);

`pow (x, y)` – x^y (помилка при $x=0, y \leq 0$ або при $x < 0, y$ – не ціле);

`sqrt (x)` – корінь квадратний з x ($x \geq 0$);

`fabs (x)` – модуль x ;

`ceil (x)` – мінімальне ціле у вигляді `double`, яке не менше за x ;

`floor (x)` – мінімальне ціле у вигляді `double`, яке не більше за x