

## Функции с параметрами

Для нормальной работы функции нужны исходные данные, и получить их она должна, согласно принципам структурного программирования, исключительно из своих параметров.

Обобщенно определение функции с параметрами выглядит так:

[спецификатор имя	класса функции([	памяти] список	[спецификатор формальных	типа] параметров]
{ тело функции }				

где [список-формальных-параметров] -- это перечисленные через запятую формальные параметры с указанием их типа.

**Параметр функции** — это принятый функцией аргумент, значение, переданное из вызывающего кода.

Различают:

- фактический параметр — аргумент, передаваемый в функцию при ее вызове;
- формальный параметр — аргумент, указываемый при объявлении или определении функции.

### Пример функции с одним параметром

```
// Объявление функции. int a - формальный параметр, имя
параметра может отсутствовать.
int myfunction(int a);

// Определение функции. int b - формальный параметр, имя
параметра может не совпадать с указанным при объявлении
функции.
int myfunction(int b)
{
    return 0;
}

int main()
{
    int c=0;
    myfunction(c); // Вызов функции. c - фактический
параметр.
    return 0;
}
```

## Передача параметра по значению

Параметры в Си передаются по значению, то есть вызывающая функция **копирует** в память, доступную вызываемой функции (сегмент стека), непосредственное фактическое **значение**.

Изменение копии переменной, соответственно, оригинал не затрагивает!

## Передача параметра по адресу

Если необходимо именно изменить переменную из внешней, по отношению к вызываемой функции, области видимости, можно копировать **адрес переменной**, подлежащей изменению. Соответственно при вызове функции **g(&x)** приходится использовать операцию взятия адреса.

Можно заметить, что передача параметра по адресу является частным случаем передачи по значению: передаваемым значением является адрес, по которому можно найти другое значение — значение переменной *x*.