

# Приемы программирования на языках C и C++

## Содержание

<a href="#">1 Указатели на функции</a>	1
<a href="#">Список литературы</a>	2
<a href="#">Список листингов</a>	2

## 1. Ресурсы по языкам C и C++

<https://learnc.info/c/>

## 2. Указатели на функции

*Указатель* – это переменная, содержащая адрес другой переменной. Если одна переменная содержит адрес другой переменной, то говорят, что она *указывает* на ту переменную [1, стр. 95].

Синтаксис объявления указателя

```
тип *имя_указателя;
```

тип – это тип переменной, на которую будет ссылаться указатель.

Указатель, не ссылающийся на конкретную ячейку памяти, должен быть равен *нулю*. Использование нулевого указателя – это всего лишь общепринятое соглашение

```
// объявляем указатель на целочисленную переменную и инициализируем его с помощью NULL
int *p = NULL;
```

---

### Замечание

Указатель можно сравнивать с нулем или с NULL, но нельзя NULL сравнивать с переменной целого типа или типа с плавающей точкой [1, стр. 103]

---

Несмотря на то, что функция не является переменной, она располагается в памяти, и, следовательно, ее адрес можно *присваивать указателю*. Этот адрес считается точкой входа в функцию. Именно он используется при вызове. Поскольку *указатель может ссылаться на функцию*, ее можно вызывать с помощью этого указателя. Это позволяет также передавать функцию другим функциям в качестве аргументов [1].

*Адрес функции* задается ее *именем*, указанным без скобок и аргументов.

Пример

```
// Подключение заголовочных файлов с помощью инструкции препроцессора #include
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

// main - это точка входа приложения
```

```

int main() {
    char s1[10], s2[10]; // Объявление строк, как массива символов
    int (*p) (const char *, const char *); // Объявление указателя на функцию
    p = strcmp; // Инициализация указателя; указателю присваивается адрес функции strcmp

    printf("Enter first string: ");
    scanf("%s", &s1);

    printf("Enter second string: ");
    scanf("%s", &s2);

    void test(
        char *x, // Указатель на символьную переменную; ожидает получить адрес
        char *y, // Указатель на символьную переменную; ожидает получить адрес
        int (*cmp) (const char *, const char *) // Указатель на функцию!
    ) {
        // Сравнивает две строки
        printf("Comparison ... \n");
        if (!((*cmp)(x, y))) { // <==NB вызов функции strcmp как (*cmp)(x, y)
            printf("=> Equal!");
        } else {
            printf("=> Not equal.");
        }
    }

    // s1 и s2 это указатели на первый символ массива
    test(s1, s2, p);

    return 0;
}

```

Важный момент: имя массива является указателем на его первый элемент [1, стр. 78]. То есть

```

double *p; // Объявление указателя на вещественную переменную
double total[50]; // Объявление на массив вещественных чисел двойной точности
p = total; // Указателю p присваивается адрес первого элемента массива total

```

Поэтому в функцию `test` передаются не явные адреса `&s1` и `&s2`, а просто имена переменных `s1` и `s2`. Ведь имена переменных `s1` и `s2` связаны со строками (по сути массивами символов), а значит имена указывают на свои первые символы и таким образом в функцию на самом деле передаются адреса первых символов этих строк.

## Список литературы

1. Кольцов Д.М. Си на примерах. Практика, практика и только практика. – СПб.: Наука и Техника, 2019. – 288 с.

## Листинги