

Лекция 10. Функции

Программы на языке Си обычно состоят из большого числа отдельных функций (подпрограмм). Как правило, эти функции имеют небольшие размеры и могут находиться как в одном, так и в нескольких файлах. Все функции являются глобальными. В языке запрещено определять одну функцию внутри другой. Связь между функциями осуществляется через аргументы, возвращаемые значения и внешние переменные.

В общем случае функции в языке Си необходимо объявлять. Объявление функции (т.е. описание заголовка) должно предшествовать ее использованию, а определение функции (т.е. полное описание) может быть помещено как после тела программы (т.е. функции *main*), так и до него. Если функция определена до тела программы, а также до ее вызовов из определений других функций, то объявление может отсутствовать. Как уже отмечалось, описание заголовка функции обычно называют прототипом функции.

Функция объявляется следующим образом:

тип имя_функции(тип имя_параметра_1, тип имя_параметра_2, ...);

Тип функции определяет тип значения, которое возвращает функция. Если тип не указан, то предполагается, что функция возвращает целое значение (*int*).

При объявлении функции для каждого ее параметра можно указать только его тип (например: *тип функции (int, float, ...)*), а можно дать и его имя (например: *тип функции (int a, float b, ...)*).

В языке Си разрешается создавать функции с переменным числом параметров. Тогда при задании прототипа вместо последнего из них указывается многоточие.

Определение функции имеет следующий вид:

тип имя_функции(тип имя_параметра_1, тип имя_параметра_2,...)
{
тело функции
}

Передача значения из вызванной функции в вызвавшую происходит с помощью оператора возврата *return*, который записывается следующим образом:

***return* выражение;**

Таких операторов в подпрограмме может быть несколько, и тогда они фиксируют соответствующие точки выхода. Например:

```
int f(int a, int b)  
{  
    if (a > b)  
    {  
        a=2*a; return a;  
    }  
    b=2*b; return b;
```

}

Вызвать эту функцию можно следующим образом:

$c = f(15, 5);$

$c = f(d, g);$

$f(d, g);$

Вызвавшая функция может, при необходимости, игнорировать возвращаемое значение. После слова **return** можно ничего не записывать; в этом случае вызвавшей функции никакого значения не передается. Управление передается вызвавшей функции и в случае выхода "по концу" (последняя закрывающая фигурная скобка).

В языке Си аргументы функции передаются по значению, т.е. вызванная функция получает свою временную копию каждого аргумента. Вызванная функция не может изменить значение переменной вызвавшей ее программы.

Если некоторые переменные, константы, массивы, структуры объявлены как глобальные, то их не надо включать в список параметров вызванной функции.

Пример 1. Пример программы расчета периметра квадрата, площади квадрата или объема куба. Сторона куба и требуемый расчет вводится с клавиатуры.

```
#include <stdio.h>                                //подключение заголовочного файла
#include <stdlib.h>                                //для перехода на русский язык

int a,R;                                           //объявление переменных типа integer
char b;                                           //объявление переменных типа char
void vol(void);                                  //объявление функции vol
void area(void);                                 //объявление функции area
void perim(void);                                //объявление функции perim

void main()                                       //основной цикл
{                                                 //начало основного цикла
    system("chcp 1251");                          //переходим в консоли на русский язык
    system("cls");                                //очищаем окно консоли
    printf("Введите сторону квадрата и параметр расчета: ");
    scanf_s("%d,%c", &a,&b);                       //считывание

    if (b == 'S')                                 //проверка условий
    {
        area();                                  //вызов функции area
    }
    else
    {
        if (b == 'V')
        {
            vol();                               //вызов функции vol
        }
        else
        {
            perim();                             //вызов функции perim
        }
    }
}
```

```

    }
}

getchar(); getchar(); getchar();
//конец основного цикла

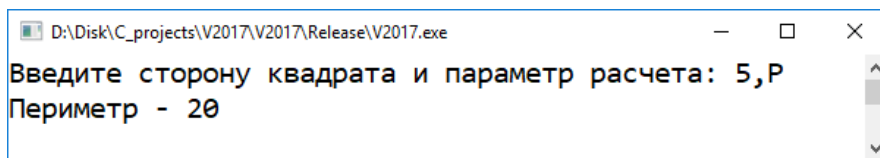
void vol(void) //функция vol
{
    R = a*a*a;
    printf("Объем - %d\n", R);
}

void area(void) //функция area
{
    R = a*a;
    printf("Площадь - %d\n", R);
}

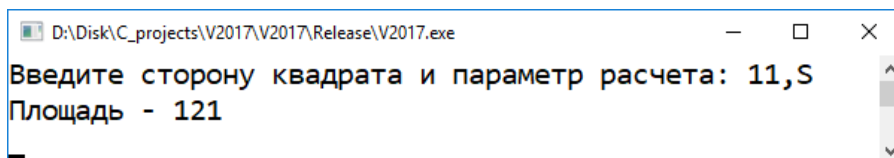
void perim(void) //функция perim
{
    R = a+a+a+a;
    printf("Периметр - %d\n", R);
}

```

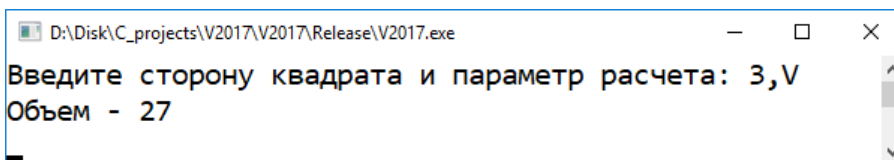
На рис. 3.26 представлен результаты выполнения данной программы.



a



б



в

Рис. 1. Программа расчета периметра квадрата, площади квадрата или объема куба: а) сторона – 5, расчет периметра; б) сторона – 11, расчет площади; в)) сторона – 3, расчет объема