ПЗ-03: **Организация функций в языке Си**

* ***Функция*** – *это синтаксически выделенный именованный программный модуль, выполняющий определенное действие или группу действий*.

Каждая **функция** имеет свой **интерфейс** и **реализацию**.

**Интерфейс** **функции** – заголовок функции, в котором указывается название функции, список ее параметров и тип возвращаемого значения.

**Реализация** **функции** – тело функции, содержащее внутренние (локальные) данные функции и программный код, выполняющий действия согласно переданным в функцию параметрам и возвращающий значение, соответствующего интерфейсу функции типа.

С точки зрения программиста функции бывают:

* **библиотечные** – функции описанные в библиотеках функций языка СИ (как стандартных, так и не стандартных);
* **пользовательские** – функции реализованные программистом в процессе разработки программы.

Описание функции на языке Си осуществляется в любом месте программы вне описания других функций и состоит из трех элементов:

1. прототип функции;
2. заголовок функции;
3. тело функции.

**Прототип функции** – необязательная часть описания функции, предназначенная для объявления некоторой функции, интерфейс которой соответствует данному прототипу.

Объявление прототипа имеет следующий вид:

**тип имя(список типов формальных параметров);**

**Параметры функции** – значения, передаваемые в функцию при ее вызове.

Выделяют понятия: **формальные** и **фактические параметры**.

**Формальные параметры** – переменные, описываемые при объявлении функции в ее прототипе и заголовке и используемые в программном коде тела функции.

**Фактические параметры** – переменные, выражения, константные значения или вызовы других функций, указываемые при непосредственном вызове функции внутри другой функции.

**Прототип функции.**

В Си можно объявлять прототип функции, а уже после описывать ее реализацию. Примеры прототипов:

***int func(int, double, double);***

***void func(int, char \*);***

***double func(void);***

**Заголовок функции** – описание интерфейсной части функции, которая содержит: тип возвращаемого значения, имя функции и список формальных параметров функции.

Синтаксис объявления заголовка функции:

**тип имя(список формальных параметров)**

Каждый элемент (формальный параметр) имеет следующий формат объявления:

**тип имя**

Примеры заголовков функций:

**int func(int i, double x, double y)**

**void func(int ind, char \*string)**

**double func(void)**

**Тело функции** – часть-реализация, содержащая программный код, выполняемый при вызове функции.

Тело функции всегда следует сразу после заголовка функции (разделять их нельзя) и заключено в фигурные скобки.

**Пример**:

Реализация функции на Си для вычисления факториала числа.

***double factorial(unsigned);***

***...***

***double factorial(unsigned num)***

***{***

***double fact = 1.0;***

***for(unsigned i=1; i<=num; i++)***

***fact \*= (double)i;***

***return fact;***

***}***

Разбиение программ на **функции** дает следующие **преимущества**:

* Функцию можно вызвать из различных мест программы, что позволяет избежать повторения программного кода.
* Одну и ту же функцию можно использовать в разных программах.
* Функции повышают уровень модульности программы и облегчают ее проектирование.
* Использование функций облегчает чтение и понимание программы и ускоряет поиск и исправление ошибок.

**Задание 1:** Определить функцию, получающую целое число и определяющую, является ли оно простым. Входной параметр – число. Возвращаемое значение «1», если число простое, «0» – в противном случае.

**NB**: *Простые числа – это натуральные числа, имеющие только два положительных делителя.*

*Простые числа – это такие числа, которые делятся на себя и на единицу (например: 2, 3, 5, 7 и т. д.). Те числа, которые делятся не только на себя и на единицу, имеют больше двух делителей, называются составными. Есть интересное число, которое делится только на себя (имеет один делитель). Это единица, она не является ни простым, ни составным*.

**Задание 2:** Написать функцию для расчета наибольшего общего делителя двух натуральных чисел, используя алгоритм Евклида. Входные параметры – два числа, возвращаемое значение – НОД.

**NB**: *Наибольший общий делитель нескольких чисел – это наибольшее натуральное целое число, на которое эти числа делятся без остатка. Наибольший общий делитель обозначается следующим образом: НОД (18; 48) = 6.*

**Задание 3:** Написать функцию, определяющую, является ли целое число полным квадратом. Входной параметр – число. Возвращаемое значение - 1, если является, 0 - в противном случае.

**NB**: *Например, 9 — это квадратное число, так как оно может быть записано в виде 3 × 3, а также представляет площадь квадрата со стороной, равной 3*.

**Задание 4:** Определить функцию для расчета суммы цифр натурального числа. Входной параметр – число. Возвращаемое значение – сумма.

**Задание 5:** Определить функцию для расчета доли (в процентах) некоторой буквы в предложении. Входные параметры – строка, символ. Возвращаемое значение – доля.

**Пример**: Реализация функции на Си для вычисления факториала числа.

*#include <stdio.h>*

*#include <conio.h>*

*long int factorial(long int n)*

*{*

*if (n == 0 || n == 1) return 1;*

*return n \* factorial(n - 1);*

*}*

*int main ()*

*{*

*long int n;*

*printf("Calculation n!\nInput n: ");*

*scanf\_s("%d", &n);*

*if (n >= 0)*

*printf("%d! = %d\n", n, factorial(n));*

*else*

*printf("Error. n must be >= 0\n");*

*getch();*

*return 0;*

*}*