

Функции.

План ответа:

1. понятие «подпрограмма», виды подпрограмм, преимущества использования;
2. общая структура функции языка Си;
3. оператор return;
4. операция вызова функции;
5. объявление и определение функции;
6. передача аргументов в функцию;
7. рекурсия.

1. понятие «подпрограмма», виды подпрограмм, преимущества использования;

Подпрограмма - именованная часть программы, содержащая описание определённого набора действий.

Подпрограмма может быть многократно вызвана из разных частей программы.

Функция - это подпрограмма специального вида, которая всегда должна возвращать результат.

Вызов функции является, с точки зрения языка программирования, выражением, он может использоваться в других выражениях или в качестве правой части присваивания.

Процедура - это независимая именованная часть программы, которую после однократного описания можно многократно вызвать по имени из других частей программы для выполнения определенных действий.

Преимущества подпрограмм:

- 1.1 Уменьшение сложности программирования.
- 1.2 Модульность (использование подпрограмм позволяет разделить большую программу на части меньшего размера, которые легче понимать и изменять).
- 1.3 Закрытость реализации (изменение реализации подпрограммы не влияет на остальную часть программы, если интерфейс подпрограммы не изменился).
- 1.4 Расширение возможностей языков программирования (создание библиотек).

2. общая структура функции языка Си;

В языке Си подпрограммы представлены только функциями.

// заголовок функции

тип-результата имя-функции(список формальных параметров с их типами)

// тело функции

```
{
    определения
    выражения
}
```

```
float avg(float a, float b)
```

```
{
    float c;
    c = a + b;
    return c / 2.0;
}
```

Любая функция может принимать параметры.

Если список параметров содержит только ключевое слово void, у функции нет параметров.

Формальные параметры перечисляются через запятую.

Определение параметра начинается с указания его типа, за которым следует имя параметра.

Для каждого имени тип указывается отдельно.

У каждой функции есть исполнимая часть, которая называется телом функции и заключена в фигурные скобки, которые также являются частью тела функции.

Тело функции может содержать как объявление переменных, так и операторы.

Переменные, описанные в теле функции, «принадлежат» только этой функции и не могут быть ни прочитаны, ни изменены другой функцией. Тело функции не может содержать в себе определения других функций.

Если функция ничего не возвращает, ее тело может быть пустым.

Функция может вернуть значение любого типа кроме массива.

Если функция ничего не возвращает, то в качестве типа возвращаемого значения следует указать `void`.

Если тип возвращаемого значения не указан, то согласно стандарту C89, компилятор предполагает,

что возвращается значение целого типа.

Согласно стандарту C99 тип возвращаемого значения опускать нельзя (warning).

Для возврата значения используется оператор `return`.

3. оператор `return`;

`return` выражение;

Оператор `return` завершает выполнение функции и возвращает управление вызывающей стороне.

Функция может содержать произвольное число операторов `return`.

Оператор `return` может использоваться в функциях типа `void`. При этом какое-либо выражение не указывается.

4. операция вызова функции;

Для вызова функции необходимо указать ее имя, за которым в круглых скобках через запятую перечислить аргументы.

Значение, возвращаемое функцией, может быть проигнорировано.

```
a = avg(2.0, 5.0);
```

```
if (avg(a, b) < 0.0)
    printf("Average is negative!\n");
```

```
beep();
```

Если опустить скобки, функция не будет вызвана.

```
beep; // warning: statement with no effect
```

5. объявление и определение функции;

Объявление функции предоставляет компилятору всю информацию, необходимую для вызова функции: количество и типы параметров,

их последовательность, тип возвращаемого значения.

Объявление функции состоит из заголовка функции.

тип-результата имя-функции(список формальных параметров с их типами);

Объявление функции должно соответствовать ее определению.

Объявление функции может не содержать имен параметров. Однако их обычно ставят для большей наглядности.

Объявление `void f(void);` означает, что у функции нет ни одного параметра.

Объявление `void f();` означает, что у функции могут быть, и могут и не быть параметры.

Если параметры есть, мы не знаем ни их количество, ни их тип.

6. передача аргументов в функцию;

Параметры функции указываются при ее определении в списке параметров.

При этом в теле функции они представляют собой имена переменных, которые передаются в функцию при ее вызове.

Аргументы функции — это выражения, которые указываются при вызове функции.

В Си все аргументы функции передаются «по значению».

Авторы языка: «Благодаря этому свойству обычно удается написать более компактную программу, содержащую меньшее число посторонних переменных, поскольку параметры можно рассматривать как должным образом инициализированные локальные переменные.»

Это можно сделать, передав в функцию не значения переменных, а адреса, по которым эти переменные располагаются в памяти.

7. рекурсия.

Функция называется рекурсивной, если она вызывает саму себя.

Например, следующая функция рекурсивно вычисляет факториал, используя формулу $n! = n * (n - 1)!$

```
int fact(int n)
{
    if (n == 0)
        return 1;
    return n * fact(n - 1);
}
```

Отобразим последовательность запусков при вызове функции fact(3):

```
fact(3) = 3 * fact(2)      // приостановка выполнения fact(3)
fact(2) = 2 * fact(1)      // приостановка выполнения fact(2)
fact(1) = 1 * fact(0)      // приостановка выполнения fact(1)
fact(0) = 1
fact(1) = 1                // возобновление выполнения fact(1)
fact(2) = 2                // возобновление выполнения fact(2)
fact(3) = 6                // возобновление выполнения fact(3)
```