

2.函数的定义和声明

杨振平

函数的定义

1. 函数定义的一般格式

＜函数值类型＞ ＜函数名＞（＜形式参数表 ＞） //函数头

{

 语句序列

}

称为函数体

一个函数由函数头和函数体两部分构成。

- 函数头中通常包括函数名，形式参数表以及函数的返回值类型，其中形式参数表必须包含在一对圆括号中。
- 函数体-由一对花括号扩起来的语句序列组成。

函数的定义（续）

2. 函数值类型-即函数的返回值类型

(1) 返回简单类型

如： `int`、`long`、`float`、`double`、`char`等

(2) 返回结构类型

(3) 返回指针类型

(4) 返回引用类型

注意：如果函数无任何返回值，这时函数的返回值类型应标记为`void`。`void`类型称为无类型或空类型。

函数的定义（续）

3. 形式参数表（即形参表）

函数的参数表用于实现函数间的数据联系。

形参的说明格式：**类型** 形参名

- ▶ 当有多个形参时，各参数说明之间用逗号分隔，如：

类型 形参名1, **类型** 形参名2, ..., **类型** 形参名n

- ▶ **注意：** 每个形参必须同时给出形参的类型和名称。

如：**int** max(**int** a,b)，作为函数首部定义是错误的。

因为形参b缺少类型说明符，即使与a类型相同，但作为形参说明也不能省略类型符**int**。

有关形参的进一步说明：

- ▶ 定义函数时需考虑设置形参，形参的个数及类型根据需要设定（也可以没有参数，称为无参函数）。
- ▶ 形参可以接收主调函数传递的实参的值。在函数中对形参进行处理，并将处理结果返回到主调函数。
- ▶ 形参是在函数调用时分配存储空间，同时接收实参的值。当函数执行结束后，系统将自动释放形参所分配的存储空间。因此，形参属于函数的局部变量，只能在该函数中使用。
- ▶ 当形参为引用类型或指针类型时，利用形参不仅可以访问实参的值，还可以改变实参的值。

函数的定义（续）

4. 函数体

- 由{ }括起来的语句序列构成，是实现函数功能的主体。
- 函数的编写过程类似于主函数。在函数中可以调用其它函数。
- 在函数体中，使用return语句返回函数执行的结果。

语句格式： return 表达式； //其中表达式类型应与函数返回值类型一致

- 对于无返回值的函数，也可使用return；将控制返回到主调函数。

说明： 在一个函数中允许出现多个return语句，但在函数执行期间只能有一个语句起作用。在函数体的最后位置，一个无返回值的return；语句可以省略。

函数定义举例：

例如：编写半径为 r 的圆面积的函数`area`。

这是一个求值的函数，函数中设置一个形参 r ，用于接收圆的半径。将参数类型和返回值类型定义为`double`。

```
double area(double r) //返回值类型为double
{
    double s;
    s=3.14*r*r;
    return s; //返回计算结果
}
```

函数定义举例：

例如：写一个函数，将给定的一个字符连续显示n次。

该函数实现一个操作功能，无须返回结果值；但需设置两个形参，一个接收显示的字符（使用char），另一个接收显示的次数（使用int）

```
void print_char(char c,int n) //函数无返回值
{
    for(int i=1;i<=n;i++)
        cout<<c;
    cout<<endl;
    return; //该语句可以省略
}
```