函数的定义

2023年8月15日 9:41

6.2 函数的定义

函数的定义一般主要有5个步骤：

1、返回值类型

2、函数名

3、参数表列

4、函数体语句

5、return 表达式

语法：

```
返回值类型 函数名 （参数列表）
{

        函数体语句

        return表达式

}
```

- 返回值类型： 一个函数可以返回一个值。在函数定义中
- 函数名：给函数起个名称
- 参数列表：使用该函数时，传入的数据
- 函数体语句：花括号内的代码，函数内需要执行的语句
- return表达式： 和返回值类型挂钩，函数执行完后，返回相应的数据

函数的作用：实现特定的功能，并全部组装在一起形成整体

示例：定义一个加法函数，实现两个数相加

```
1  //函数定义
2  int add(int num1, int num2)
3  {
4      int sum = num1 + num2;
5      return sum;
6  }
```

函数的调用

2023年9月13日 19:48

6.3 函数的调用

功能： 使用定义好的函数

语法： 函数名（参数）

```
//函数定义
int add(int num1, int num2) //定义中的num1,num2称为形式参数，简称形参
{
    int sum = num1 + num2;
    return sum;
}

int main() {

    int a = 10;
    int b = 10;
    //调用add函数
    int sum = add(a, b);//调用时的a，b称为实际参数，简称实参
    cout << "sum = " << sum << endl;

    a = 100;
    b = 100;

    sum = add(a, b);
    cout << "sum = " << sum << endl;

    system("pause");

    return 0;
}
```

总结：函数定义里小括号内称为形参，函数调用时传入的参数称为实参

值传递

2023年9月13日 19:48

6.4 值传递

- 所谓值传递，就是函数调用时实参将数值传入给形参
- 值传递时，如果形参发生，并不会影响实参

> 总结：值传递时，形参是修饰不了实参的

```
void swap(int num1, int num2)
{
    cout << "交换前: " << endl;
    cout << "num1 = " << num1 << endl;
    cout << "num2 = " << num2 << endl;

    int temp = num1;
    num1 = num2;
    num2 = temp;

    cout << "交换后: " << endl;
    cout << "num1 = " << num1 << endl;
    cout << "num2 = " << num2 << endl;

    //return ; 当函数声明时候，不需要返回值，可以不写return
}
```

```
int main() {

    int a = 10;
    int b = 20;

    swap(a, b);

    cout << "mian中的 a = " << a << endl;
    cout << "mian中的 b = " << b << endl;

    system("pause");

    return 0;
}
```

函数的常见样式

2023年9月14日 10:28

6.5 函数的常见样式

常见的函数样式有4种

1. 无参无返
2. 有参无返
3. 无参有返
4. 有参有返

示例：

```
//函数常见样式
//1、 无参无返
void test01()
{
    //void a = 10; //无类型不可以创建变量,原因无法分配内存
    cout << "this is test01" << endl;
    //test01(); 函数调用
}

//2、 有参无返
void test02(int a)
{
    cout << "this is test02" << endl;
    cout << "a = " << a << endl;
}

```

```
16
17  //3、无参有返
18  int test03()
19  {
20      cout << "this is test03 " << endl;
21      return 10;
22  }
23
24  //4、有参有返
25  int test04(int a, int b)
26  {
27      cout << "this is test04 " << endl;
28      int sum = a + b;
29      return sum;
30  }
```

函数的声明

2023年9月13日 19:48

6.6 函数的声明

作用： 告诉编译器函数名称及如何调用函数。函数的实际主体可以单独定义。

- 函数的**声明可以多次**，但是函数的**定义只能有一次**

```
1  //声明可以多次，定义只能一次
2  //声明
3  int max(int a, int b);
4  int max(int a, int b);
5  //定义
6  int max(int a, int b)
7  {
8      return a > b ? a : b;
9  }
10
11 int main() {
12
13     int a = 100;
14     int b = 200;
15
16     cout << max(a, b) << endl;
17
18     system("pause");
19
20     return 0;
```

函数的分文件编写

2023年9月14日 10:35

6.7 函数的分文件编写

作用：让代码结构更加清晰

函数分文件编写一般有4个步骤

1. 创建后缀名为.h的头文件
2. 创建后缀名为.cpp的源文件
3. 在头文件中写函数的声明
4. 在源文件中写函数的定义

+

示例：

源文件中需要引用头文件：输入：#include” ……“ 这个表示自定义的头文件；
#include"……" 这个表示规定的头文件

```
#include<iostream>
using namespace std;
#include "swap.h"
```

1、头文件写函数的声明，若为自定义：语法为 - 返回值类型 函数名称（参数或不用写参数）；

```
#include <iostream>
using namespace std;

//函数的声明
void swap(int a, int b);
```

2、源文件写函数的定义，语法为-返回值类型 函数名称（参数或不用写）{函数的运算定义 }
注意：要引用头文件

```
#include "swap.h"
//函数的定义
void swap(int a, int b)
{
    int temp = a;
    a = b;
    b = temp;

    cout << "a = " << a << endl;
    cout << "b = " << b << endl;
}
```

Swap.h 指的是头文件

3、源文件中写具体的运算代码
注意：要引用头文件

```
int main() {

    int a = 10;
    int b = 20;

    swap(a, b);

    system("pause");

    return 0;
}
```