【C++】C++中函数重载的理解

原创 忽晴忽雨江湖 2018-06-19 17:37:05

我们在平时写代码中会用到几个函数但是他们的实现功能相同，但是有些细节却不同。例如：交换两个数的值其中包括（int, float,char,double)这些个类型。在C语言中我们是利用不同的函数名来加以区分。

```
void Swap1(int* a, int* b);
void Swap2(float* a, float* b);
void Swap3(char* a, char* b);

void Swap4(double* a, double* b);
```

我们可以看出这样的代码不美观而且给程序猿也带来了很多的不便。于是在C++中人们提出了用一个函数名定义多个函数，也就是所谓的函数重载。

一.函数重载定义

函数重载是一种特殊情况，C++允许在同一作用域中声明几个类似的同名函数，这些同名函数的形参列表（参数个数，类型，顺序）必须不同，常用来处理实现功能类似数据类型不同的问题。

在C++中不仅函数可以重载，运算符也可以重载。例如：

运算符<<,>>。既可以做移位运算符，也可以做输出，输入运算符。

注意：重载函数的参数个数，参数类型或参数顺序三者中必须有一个不同

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

int Add(int a, int b)
{

    return a + b;

}

double Add(double a, double b)
{

    return a + b;
}

float Add(float a, float b)
```

```
{
    return a + b;

}
int main()
{

    cout<<Add(1,2)<<endl;

    cout<<Add(3.5, 4.5)<<endl;


    cout << Add(2.22, 3.33) << endl;
    system("pause");
    return 0;
}
```

我们可以看到定义了一个Add函数来求三个不同类型数的和，在调用过程中系统会自动根据其实参的类型不同来实现准确调用。

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

int main()
{

    int max(int a, int b, int c);
    int max(int a, int b);
    int a = 10;
    int b = 20;
    int c = 30;


    cout << max(a, b, c) << endl;
    cout << max(a, b) << endl;
    system("pause");
    return 0;
}


int max(int a, int b, int c)
{
    if (b > a)
        a = b;
    if (c > a)
        a = c;
    return a;

}

int max(int a, int b)
{

    return (a > b) ? a : b;
}
```

从上边代码可以看出函数重载除了允许函数类型不同以外，换允许参数个数不同。

函数重载的规则：

- 函数名称必须相同。
- 参数列表必须不同（个数不同、类型不同、参数排列顺序不同等）。
- 函数的返回类型可以相同也可以不相同。
- 仅仅返回类型不同不足以成为函数的重载。

二、函数重载的作用：

重载函数通常用来在同一个作用域内 用同一个函数名 命名一组功能相似的函数，这样做减少了函数名的数量，避免了名字空间的污染，对于程序的可读性有很大的好处。

三、函数重载是一种静态多态：

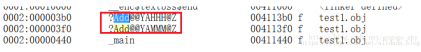
（1）多态：用同一个东西表示不同的形态；
（2）多态分为：
静态多态（编译时的多态）；
动态多态（运行时的多态）；
（3）函数重载是一种静态多态；

四.面试题

1.C语言中为什么不能支持函数重载?

从上图可知编译器在编译.c文件时，只会给函数进行简单的重命名；具体的方法是给函数名之前加上”_”;所以加入两个函数名相同的函数在编译之后的函数名也照样相同；调用者会因为不知道到底调用那个而出错；

2.C++中函数重载底层是如何处理的?



在.cpp文件中，虽然两个函数的函数名一样，但是他们在符号表中生成的名称不一样。

‘？’表示名称开始，‘？’后边是函数名“@@YA”表示参数表开始，后边的3个字符分别表示返回值类型，两个参数类型。“@Z”表示名称结束。

由于在.cpp文件中，两个函数生成的符号表中的名称不一样，所以是可以编译通过的。

3.C++中能否将一个函数按照C的风格来编译?

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


extern "C" int Add(int a, int b)
{

    return a + b;
}
int main()
{
    cout << Add(10, 20) << endl;
    system("pause");
    return 0;
}

```

可以按照C风格来编译，只需在函数名前加extern "C" 就可以完成按照C风格来编译