

# 构造函数为什么没有返回值?

2020-05-16

刚才同学问了我一个问题，C++里new一个类的成员函数是什么意思?

.....我心想，new成员函数? 还有这种操作?

后来我问，这个成员函数，名字是不是和类名一样?

他说一样，就是构造函数。

.....那不就是new一个对象吗?

然后他问了一个深刻的问题，为什么构造函数没有返回值?

我说new的是类，new后面指的是类名，不是成员函数。

可是类没有参数啊?

构造函数有参数。

但是构造函数没有返回值啊?

啊?



这是一段简单的代码:

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

class Test
{
    public:
        Test(int num);
};

// 构造函数，有参数没有返回值
Test::Test(int num)
{
    cout << num << endl;
}

int main()
{
    // 对象赋值给了变量
    Test* test = new Test(1);
    return 0;
}
```

其实不看代码也能想到这个场景。new一个对象的时候，new后面的变量，指的的类名，还是方法名?

如果指的是类名，类本身没有参数，也没有返回值，而且new这个关键字在代码的行为上也会特殊于其他语句。另外，类会有一个默认的无参构造方法，如果new的是类，要默认的构造方法干嘛？

如果指的是函数名，构造函数没有返回值，像上面的代码里明显就把new Test(1)赋值给了Test\* test。

构造函数为什么没有返回值？因为构造函数在执行的时候，告诉编译器在内存上开辟多大的空间，初始化了成员变量，确定了this的地址，然后干了这些事情之后，就不允许用户自定义返回值的类型了？因为构造函数的返回值一定且必须是它自己，所以就不需要对用户透明了（编译器为什么要“擅自”做这样的事情？）？

构造函数为什么没有返回值？我并不知道这个问题的答案，也不大有兴趣知道。但这个问题带给我很大的启发，也带给我很大的震撼，“new成员函数”这种说法真是思路清奇。我们对太多东西习以为常，司空见惯，觉得它应该就是那样，却很少问它为什么是那样，很少认真去思考和认识很多事物。