

C++ 类 & 对象

C++ 在 C 语言的基础上增加了面向对象编程，C++ 支持面向对象程序设计。类是 C++ 的核心特性，通常被称为用户定义的类型。

类用于指定对象的形式，它包含了数据表示法和用于处理数据的方法。类中的数据和方法称为类的成员。函数在一个类中被称为类的成员。

C++ 类定义

定义一个类，本质上是定义一个数据类型的蓝图。这实际上并没有定义任何数据，但它定义了类的名称意味着什么，也就是说，它定义了类的对象包括了什么，以及可以在这个对象上执行哪些操作。

类定义是以关键字 **class** 开头，后跟类的名称。类的主体是包含在一对花括号中。类定义后必须跟着一个分号或一个声明列表。例如，我们使用关键字 **class** 定义 Box 数据类型，如下所示：

```
class Box
{
public:
double length; // 盒子的长度
double breadth; // 盒子的宽度
double height; // 盒子的高度
};
```

关键字 **public** 确定了类成员的访问属性。在类对象作用域内，公共成员在类的外部是可访问的。您也可以指定类的成员为 **private** 或 **protected**，这个我们稍后会进行讲解。

定义 C++ 对象

类提供了对象的蓝图，所以基本上，对象是根据类来创建的。声明类的对象，就像声明基本类型的变量一样。下面的语句声明了类 Box 的两个对象：

```
Box Box1; // 声明 Box1，类型为 Box
Box Box2; // 声明 Box2，类型为 Box
```

对象 Box1 和 Box2 都有它们各自的数据成员。

访问数据成员

类的对象的公共数据成员可以使用直接成员访问运算符 (.) 来访问。为了更好地理解这些概念，让我们尝试一下下面的实例：

实例

```
#include <iostream>
using namespace std;
class Box
{
public:
double length; // 长度
```

```
double breadth; // 宽度
double height; // 高度
};
int main( )
{
    Box Box1; // 声明 Box1, 类型为 Box
    Box Box2; // 声明 Box2, 类型为 Box
    double volume = 0.0; // 用于存储体积
    // box 1 详述
    Box1.height = 5.0;
    Box1.length = 6.0;
    Box1.breadth = 7.0;
    // box 2 详述
    Box2.height = 10.0;
    Box2.length = 12.0;
    Box2.breadth = 13.0;
    // box 1 的体积
    volume = Box1.height * Box1.length * Box1.breadth;
    cout << "Box1 的体积: " << volume <<endl;
    // box 2 的体积
    volume = Box2.height * Box2.length * Box2.breadth;
    cout << "Box2 的体积: " << volume <<endl;
    return 0;
}
```

当上面的代码被编译和执行时，它会产生下列结果：

```
Box1 的体积: 210
Box2 的体积: 1560
```

需要注意的是，私有的成员和受保护的成员不能使用直接成员访问运算符 (.) 来直接访问。我们将在后续的教程中学习如何访问私有成员和受保护的成员。

类 & 对象详解

到目前为止，我们已经对 C++ 的类和对象有了基本的了解。下面的列表中还列出了其他一些 C++ 类和对象相关的概念，可以点击相应的链接进行学习。

概念	描述
类成员函数	类的成员函数是指那些把定义和原型写在类定义内部的函数，就像类定义中的其他变量一样。
类访问修饰符	类成员可以被定义为 public、private 或 protected。默认情况下是定义为 private。
构造函数 & 析构函数	类的构造函数是一种特殊的函数，在创建一个新的对象时调用。类的析构函数也是一种特殊的函数，在删除所创建的对象时调用。
C++ 拷贝构造函数	拷贝构造函数，是一种特殊的构造函数，它在创建对象时，是使用同一类

	中之前创建的对象来初始化新创建的对象。
C++ 友元函数	友元函数 可以访问类的 private 和 protected 成员。
C++ 内联函数	通过内联函数，编译器试图在调用函数的地方扩展函数体中的代码。
C++ 中的 this 指针	每个对象都有一个特殊的指针 this ，它指向对象本身。
C++ 中指向类的指针	指向类的指针方式如同指向结构的指针。实际上，类可以看成是一个带有函数的结构。
C++ 类的静态成员	类的数据成员和函数成员都可以被声明为静态的。

← C++ 把引用作为返回值

C++ 继承 →

+

4 篇笔记

✎ 写笔记