

二级C 类与对象：static静态数据成员静态成员函数 PDF转换
可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/134/2021_2022__E4_BA_8C_E7_BA_A7C___E7_c97_134477.htm

static 静态数据成员 静态成员函数 静态成员的提出是为了解决数据共享的问题。实现共享有许多方法，如：设置全局性的变量或对象是一种方法。但是，全局变量或对象是有局限性的。这里主要讲述类的静态成员来实现数据的共享。静态数据成员在类中，静态成员可以实现多个对象之间的数据共享，并且使用静态数据成员还不会破坏隐藏的原则，即保证了安全性。因此，静态成员是类的所有对象中共享的成员，而不是某个对象的成员。使用静态数据成员可以节省内存，因为它是所有对象所共有的，因此，对多个对象来说，静态数据成员只存储一处，供所有对象共用。静态数据成员的值对每个对象都是一样，但它的值是可以更新的。只要对静态数据成员的值更新一次，保证所有对象存取更新后的相同的值，这样可以提高时间效率。静态数据成员的使用方法和注意事项如下：1、静态数据成员在定义或说明时前面加关键字static。2、静态成员初始化与一般数据成员初始化不同。静态数据成员初始化的格式如下：`::=` 这表明：(1) 初始化在类体外进行，而前面不加static，以免与一般静态变量或对象相混淆。(2) 初始化时不加该成员的访问权限控制符private，public等。(3) 初始化时使用作用域运算符来标明它所属类，因此，静态数据成员是类的成员，而不是对象的成员。3、静态数据成员是静态存储的，它是静态生存期，必须对它进行初始化。4、引用静态数据成员时，采用如下格式：`::` 如果静态数据成员的访问权限允

许的话(即public的成员),可在程序中,按上述格式来引用静态数据成员。静态成员函数 静态成员函数和静态数据成员一样,它们都属于类的静态成员,它们都不是对象成员。因此,对静态成员的引用不需要用对象名。在静态成员函数的实现中不能直接引用类中说明的非静态成员,可以引用类中说明的静态成员。如果静态成员函数中要引用非静态成员时,可通过对象来引用。C会区分两种类型的成员函数:静态成员函数和非静态成员函数。这两者之间的一个重大区别是,静态成员函数不接受隐含的this自变量。所以,它就无法访问自己类的非静态成员。在某些条件下,比如说在使用诸如pthread(它不支持类)此类的多线程库时,就必须使用静态的成员函数,因为其地址同C语言函数的地址兼容。这种限制就迫使程序员要利用各种解决办法才能够从静态成员函数访问到非静态数据成员。第一个解决办法是声明类的所有数据成员都是静态的。运用这种方式的话,静态的成员函数就能够直接地访问它们,例如: `class Singleton{ public: static Singleton * instance().private: Singleton * p. static Lock lock.}.Singleton * Singleton::instance(){ lock.getlock(). // fine, lock is static if (!p) p=new Singleton. lock.unlock(). return p.}` 这种解决方法不适用于需要使用非静态数据成员的类。访问非静态数据成员将参照传递给需要考量的对象能够让静态的成员函数访问到对象的非静态数据: `class A{public: static void func(A amp. obj){ int n = obj.getval().}` 将一个参照或者指针作为静态成员函数的自变量传递,就是在模仿自动传递非静态成员函数里this自变量这一行为。本文部分来自作者Danny Kalev,他是一个系统分析家、软件工程师,在C和面向对象设计方面有

着14年的专业经验。其他部分来自网络，未能一一提及所有的作者，希望见谅。静态成员函数与普通成员函数的差别就在于缺少this指针，没有这个this指针自然也就无从知道name是哪一个人的成员了。根据类静态成员的特性我们可以简单归纳出几点，静态成员的使用范围：1.用来保存对象的个数。2.作为一个标记，标记一些动作是否发生，比如：文件的打开状态，打印机的使用状态，等等。3.存储链表的第一个或者最后一个成员的内存地址。100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com