**C/C++错题集之static相关**



[xuri8888](https://blog.csdn.net/m0_38099899) 2018-02-27 21:01:09  150  收藏

分类专栏： [C/C++](https://blog.csdn.net/m0_38099899/category_7471523.html)

版权

1. 静态数据成员只能在全局区域进行初始化，而不能在类体中进行（构造函数中初始化也不行），且静态数据成员不涉及对象，因此不受类访问限定符的限制。
2. 类方法是指类中被static修饰的方法，无this指针

    类方法是可以调用其它类的static方法的

    可以在类方法中生成实例对象再调用实例方法

    成员方法又称为实例方法

1. 局部变量说明为static存储数，其生存期将得到延长，离开局部时，局部变量的值会保存，下次再回到局部时可以读取保存的值；

     全局变量声明为static后，其作用域非但没有扩大，反而缩小，只能在所在的文件内使用；

     静态变量默认初始化为0；

     局部变量可以使用static，形参不能

1. 虚成员函数不可能是static成员函数
2. 外部变量，可供所以源文件使用

寄存器变量，放在寄存器而非内存中，效率更高，一般是临时变量

自动变量，所有未加 static 关键字的都默认是 auto 变量，也就是我们普通的变量

静态变量，在内存中只存在一个，可供当前源文件的所有函数使用

1. 静态存储变量具有局部的生存期.说法是否正确？错误

在程序运行期间都不释放  static不在栈区存储，所以不是局部生存期。

1. C++的多态性分为静态多态和动态多态。   
   静态多态性：编译期间确定具体执行哪一项操作，主要是通过函数重载和运算符重载来实现的；   
   动态多态性：运行时确定具体执行哪一项操作，主要是通过虚函数来实现的。
2. 类中的静态成员是被类所有实例所共享的，它不计入sizeof计算的空间   
   类中的普通函数或静态普通函数都存储在栈中，不计入sizeof计算的空间
3. 静态多态（早绑定），如函数重载

动态多态（晚绑定），如虚函数，继承

1. virtual 不能修饰静态成员函数、内联函数和构造函数
2. 类的静态成员函数可以调用静态数据成员，但是不能调用非静态数据成员和非静态成员函数

    静态数据成员必须单独初始化

    静态数据成员只有一份，且不依赖对象而存在

    静态成员函数不能加const

1. 静态联编与动态联编：如果是通过对象名调用虚函数，在编译期间就能确定调用的是哪一个类的虚函数，所以属于静态关联；如果是通过基类指针调用虚函数，在编译阶段无法从语句本身确定调用哪一个类的虚函数，只有在运行时，pt指向某一个类对象后，才能确定调用的是哪一个类的虚函数，故为动态关联
2. 静态存储变量具有局部的生存期：FALSE
3. 静态链表中指针表示的是下一个元素在数组中的位置
4. 通过一个对象调用虚函数时，C++系统对该调用采用静态联编
5. 静态函数属于类本身，不与类绑定，因此不能用this指针。

在类定义时，无法使用构造函数，因而无法完成对象的初始化

一般成员变量需要在类内进行初始化

静态成员变量必须在类外初始化，静态成员常量在类中初始化。

1. 静态分配是指在编译阶段就能确定大小，由编译器进行分配，堆不可以进行静态分配，堆的申请都是在执行过程中进行的。堆是使用malloc()、calloc()、realloc()等函数动态分配的，而使用alloca()函数可以动态分配栈的内存空间，释放的时候由编译器自己释放。