

探讨C 实现一个不可被继承的类计算机等级考试 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/641/2021_2022__E6_8E_A2_E8_AE_A8C___E5_c97_641758.htm C#和Java都提供了一种机制

让一个类不能被继承，如C#中的sealed关键字和Java的final关键字，然而C 程序员就没这么好命了。不过C 也可以模拟出这种效果，原理基于：子类的构造函数会自动调用父类的构造函数，同理析构函数也是一样。如果父类的构造函数和析构函数被设为私有的话，那么子类就无法调用，也就达到了父类不可被继承的目的了。原理很简单，按此原理我也实作出一个自认为很实用的工具类，在此献下丑，欢迎大家批评

```
: template<T>. class CFobidDeriveProviderBase { friend TDerive.
friend TProvider. private: CFobidDeriveProviderBase(){}
~CFobidDeriveProviderBase(){} }. /* * 提供禁止派生的功能,需要此功能的类可以从CFobidDeriveProvider派生,并将类名作为模板参数传递 */ template<T>. class CFobidDeriveProvider : virtual
public CFobidDeriveProviderBase<T>.TDerive<T>. { public:
CFobidDeriveProvider(){} ~CFobidDeriveProvider(){} }. /* * 测试类,该类不可被继承 */ class CNoDerive : public
CFobidDeriveProvider<T>. { public: CNoDerive(){} ~CNoDerive(){}
void Alert() { AtlMessageBox( NULL, _T("Alert") ). } }. 之所以将继承的结构分为2层：CFobidDeriveProvider
和CFobidDeriveProviderBase，主要是方便使用，用户只需直接从CFobidDeriveProvider派生就可实现一个不可被继承的类，而不需要虚拟继承。若有类从CNoDerive派生： class
CSomeDerive : public CNoDerive { public: CSomeDerive(){}
```

~CSomeDerive(){} }. CSomeDerive的构造函数调用过程如下：
由于CFobidDeriveProvider是从CFobidDeriveProviderBase虚拟派生，在虚继承出现的继承层次中，总是在构造非虚基类之前构造虚基类，因而会跳过CNoDerive和CFobidDeriveProvider的构造函数而直接调用CFobidDeriveProviderBase的构造函数，但CSomeDerive不是CFobidDeriveProviderBase的友元，因此也无法调用CFobidDeriveProviderBase的私有构造函数。故而编译错误。 编辑特别推荐: 全国计算机等级考试（等考）指定教材 全国计算机等级考试学习视频 全国计算机等级考试网上辅导招生 全国计算机等级考试时间及科目预告 百考试题教育全国计算机等级考试在线测试平台 全国计算机等级考试资料下载 全国计算机等级考试论坛 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