

C\_CLI中实现singleton模式 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/133/2021\\_2022\\_C\\_\\_\\_CLI\\_E4\\_B8\\_AD\\_c97\\_133654.htm](https://www.100test.com/kao_ti2020/133/2021_2022_C___CLI_E4_B8_AD_c97_133654.htm) 双重检测锁 ( Double-Checked

Locking ) 实现的Singleton模式在多线程应用中有相当的价值。在ACE的实现中就大量使用ACE\_Singleton模板类将普通类转换成具有Singleton行为的类。这种方式很好地消除了一些重复代码臭味，而且，优化后的性能较标准互斥版本提高15倍。最近在用C /CLI做一些工作，Singleton不可避免地需要用到，于是我又制造了一次车轮。

```
1 #pragma once 2 3 /** \class
sidle::Singleton 4 \brief Singleton (Double-Checked Locking) 5
\author 吴尔平 6 \version 1.0 7 \date 2005.02.08 - 8 \bug 9 \warning
10 */ 11 12 namespace sidle 13 { 14 using namespace System. 15
using namespace System::Threading. 16 17 template 18 ref class
Singleton 19 { 20 public: 21 static _T^ Instance() 22 { 23 if (_instance
== nullptr) 24 { 25 _mut->WaitOne(). 26 try 27 { 28 if (_instance
== nullptr) 29 { 30 _instance = gcnew _T(). 31 } 32 } 33 finally 34 {
35 _mut->ReleaseMutex(). 36 } 37 } 38 return _instance. 39 } 40
protected: 41 Singleton(){} 42 static _T^ _instance. 43 static Mutex^
_mut = gcnew Mutex(). 44 }. // ref class Singleton 45 46 }. //
```

namespace sidle 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](http://www.100test.com)