

🔗C#面向对象设计模式纵横谈(2): Singleton 单件(创建型模式) ---Level 300

活动日期: 2005-10-25 14:30 -- 16:00

主 讲: 李建忠

Q: 使用静态的计数器一样可以在单线程中实现只实例化一个对象的目的啊

A: 这个应该是不能的, 因为静态计数器的作用和 `if (instance == null)` 是一样的, 在多线程环境中都会有问题的。

Q: 多线程中的 lock 可以 lock(this)吗?

A: 因为是在静态属性中, 所以不能访问 this 指针。

Q: 为什么双检查?

A: 单检查也是可以的, 但是单检查的效率要比双检查低——因为同步控制的时间太长了。双检查能够最高效地实现多线程安全的访问。

Q: 为什么一定要加 readonly 关键字?

A: 这个 readonly 关键字只是不希望客户程序将 Instance 字段设置为 null 等不合理的值。

Q: remoting 里面的 Singleton 对象应该是使用了 Singleton 模式吧

A: 是的, .NET Remoting 中的服务器对象激活中就使用了 Singleton 模式

Q: 怎样获得类已经构造的实例的个数?

A: 可以在实例构造器中放一个静态的字段, 来表示计数器——在实例构造器中每次做 `count++` 即可。

Q: 怎样区分各个模式, 学了很久, 总是搞不清楚他们之间的区别, 经常性的搞混

A: 区分模式的最好办法是搞清楚为什么有这些模式，各个模式分别应对什么样的变化。

Q: 当好一个程序员必须要学好设计模式吗？它在代码编写过程中有什么好处？怎样可以学好设计模式？

A: 不一定，我了解的某些天才程序员对设计模式并不感兴趣——主要是因为他们首先不是面向对象程序员☺ 但是学好设计模式对于一个面向对象程序员有莫大帮助。学好设计模式的关键是深刻理解面向对象。

Q: lock 对于 singleton 本身的类使用 与 使用 helper 有什么区别？

A: 本质上没什么区别，但是别忘了这时候 Singleton 对象还没有创建☺ 所以这时候不可能 lock 一个 Singleton 对象。

Q: 我有一个疑问，在 singleton 设计模式下，什么时候，由谁来创建这个实例呢？

A: Singleton 模式中的“缓式加载”已经说明了 Singleton 的实例是在客户程序第一次调用 GetInstance 方法时才会被创建。

Q: 我大致的翻过设计模式这本书，我想请教下您，您认为在设计一个很好的面向对象的软件与程序语言的选择（比如 C#，C++，JAVA）二者之间怎么做到最好的搭配

A: 我个人认为这三门语言都是很好的面向对象语言，都能很充分地发挥面向对象的力量。在面向对象层次上，它们的差别并不大。

Q: 在多线程环境中，使用 Static 实例化一个对象后，那么它的实例的方法是否可以保证执行时不致冲突？

A: 实例方法在多线程环境中无所谓冲突，关键是实例方法操作的实例数据——如果有的话——有可能冲突。

