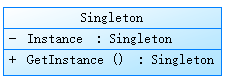
1、定义

单例模式是确保一个类只有一个实例，并提供一个全局访问方式的设计方法

2、UML类图



3、应用场景

  需要频繁实例化然后销毁的对象。

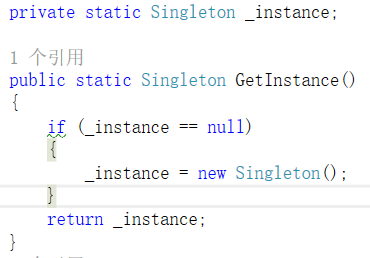
  创建对象时耗时过多或者耗资源过多，但又经常用到的对象。

  有状态的工具类对象。

  频繁访问数据库或文件的对象。

4、C#实现单例模式

4.1 单线程单例模式



代码解析:

Singleton的构造函数必须是私有的,确保外层无法通过new进行实例化对象

静态变量的生命周期同应用程序的生命周期相同，可以定义一个私有的全局静态变量\_instance来保存该类的唯一实例

提供一个可访问的静态方法访问类的实例，在该方法内通过判断\_instance是否为null，如果是null就创建一个实例,否则向外部返回实例

4.2 多线程单例模式

1) lock方式

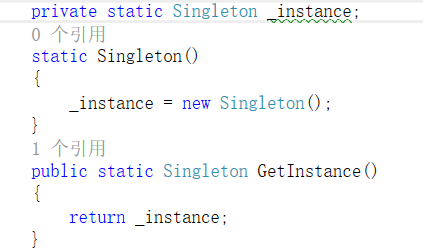


代码解析:

外层if判断\_instance不为null时，直接返回实例，避免每次获取实例时进行锁定，节省了 性能损耗

当外层判断成功时，使用lock锁定，保证只创建一个实例

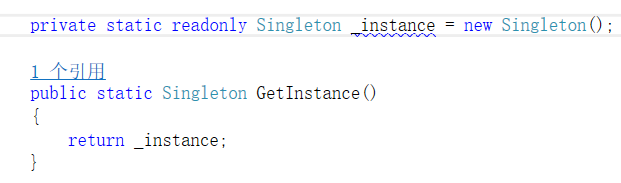
2) 静态构造函数



代码解析:

静态构造函数只会执行一次

3) 静态变量初始化



代码解析:

静态变量在初次运行时就会初始化，readonly确保变量运行时无法修改