多线程下的单例模式：

// Copyright 2016.刘珅珅

// author：刘珅珅

// 单例模式

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace singleton\_test1

{

public sealed class Singleton

{

private static Singleton instance;

private static object \_lock = new object();

// 构造函数私有化

private Singleton() {}

public static Singleton GetInstance()

{

if (instance == null)

{

// 多线程下加锁

lock(\_lock)

{

if (instance == null)

{

instance = new Singleton();

}

}

}

return instance;

}

}

public sealed class Singleton2

{

// 不需要考虑多线程的问题

// 在运行时CLR自动创建这个实例

private static readonly Singleton2 instance = new Singleton2();

private Singleton2() { }

public static Singleton2 GetInstance()

{

return instance;

}

}

class SingletonTest

{

static void Main(string[] args)

{

}

}

}