Данный паттерн гарантирует, что пользователь, сможет многократно вызывать метод Dispose().

using System;

namespace FinalizableDisposableClass

{

// Шаблон Microsoft для освобождения ресурсов.

// Данный паттерн гарантирует, что пользователь, сможет многократно вызывать метод Dispose().

public class ResourceWrapper : IDisposable

{

// Флаг показывающий вызов метода Dispose()

private bool disposed = false;

public void Dispose()

{

// Вызов вспомогательного метода.

// Если true, значит очистку инициировал пользователь объекта.

CleanUp(true);

// SuppressFinalize() - устанавливает флаг запрещения завершения для объектов

// которые в противном случае могли бы быть завершены сборщиком мусора.

// Отменяет работу деструктора для данного класса.

GC.SuppressFinalize(this);

}

// Сборщик мусора вызывает Деструктор, если пользователь объекта забудет вызвать метод Dispose().

~ResourceWrapper()

{

Console.WriteLine("Finalise!!!!!!!!!!!");

CleanUp(false);

}

// Метод для избежания дублирования кода в Деструкторе и методе Dispose().

private void CleanUp(bool clean)

{

// Проверка, что ресурсы еще не освобождены.

if (!this.disposed)

{

// Если clean равно true, освободить все управляемые ресурсы.

if (clean)

{

Console.Write("Освобождение ресурсов");

for (int i = 0; i < 40; i++)

{

Console.Write("F");

}

}

Console.WriteLine("Finalized");

}

this.disposed = true;

}

}

}