# Singleton – Сек

**Сек** *(Singleton)* е създаващ [шаблон за дизайн](http://bg.wikipedia.org/wiki/%D0%A8%D0%B0%D0%B1%D0%BB%D0%BE%D0%BD%D0%B8_%D0%B7%D0%B0_%D0%B4%D0%B8%D0%B7%D0%B0%D0%B9%D0%BD), който се използва в [обектно-ориентираното](http://bg.wikipedia.org/wiki/%D0%9E%D0%B1%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%B0%D0%BD%D0%BE_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%B8%D1%80%D0%B0%D0%BD%D0%B5) [програмиране](http://bg.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%B8%D1%80%D0%B0%D0%BD%D0%B5). Този шаблон се използва обикновено в моделирането на обекти, които трябва да бъдат глобално достъпни за обектите на приложението (например обекта съдрържайки структурите с настройките на програмата ) или обекти, които се нуждаят от максимално късна инициализация за пестенето на ресурси от паметта.

// Пример за Сек

using System;

namespace SingletonDesignPattern

{

/// <summary>

/// главния клас за стартиране на примерното приложение

/// </summary>

class MainApp

{

/// <summary>

/// Главния метод за стартиране

/// </summary>

static void Main()

{

// Конструктора (Constructor) е защитен (protected) -- операторът new не може да бъде извикан

Singleton s1 = Singleton.Instance();

Singleton s2 = Singleton.Instance();

// Test for same instance

if (s1 == s2)

{

Console.WriteLine("Objects are the same instance"); //и двата обекта са една и съща инстанция (оплътворение) на този клас

}

// Изчакай удар по конзолата от потребителя

Console.ReadKey();

}

}

/// <summary>

/// The 'Singleton' class - класът Сек

/// </summary>

class Singleton

{

private static Singleton \_instance; //променлива за единствената инстанция на този клас

// Constructor is 'protected' - конструтора е защитен и не може да бъде извикан

protected Singleton()

{

}

//единтвения начин за инстанцииране е от тук

public static Singleton Instance()

{

// Uses lazy initialization. - използва късна инициализация

// Note: this is not thread safe. - Да не се използва в многонишкови приложения

if (\_instance == null)

{

\_instance = new Singleton();

}

return \_instance;

}

}

}

