**SingletonMono**

**设计初衷**

本类用于构建依赖 MonoBehaviour 生命周期的全局管理类，适合需要挂载到场景中的 GameObject，或使用 Unity 生命周期函数（如 Update、Start、OnEnable）的系统型组件，例如：音频管理器、UI 管理器、输入控制器等。

**文件路径**

文件路径：Frame/Scripts/SingletonMono

**使用方法**

继承该类，并实现 Reset() 方法。子类可在 Awake() 方法中进行初始化，并应调用 base.Awake() 以保证单例逻辑生效：

*public class AudioManager : SingletonMono<AudioManager>*

*{*

*public AudioSource Bgm;*

*protected override void Awake()*

*{*

*base.Awake();*

*// 初始化逻辑*

*Bgm = GetComponent<AudioSource>();*

*}*

*public override void Reset()*

*{*

*// 清空状态*

*Bgm.Stop();*

*}*

*}*

**公共方法**

**-**Instance: 获取当前类型的单例实例。若实例不存在，会尝试查找场景对象；若仍不存在，则动态创建 GameObject 并挂载组件。

-IsInstantiated: 判断是否已经实例化；

-Reset(): 单例重置接口。用于场景切换时清空状态或缓存，不销毁实例；

-Awake(): Unity 生命周期函数。用于确保只有一个单例实例，默认设置 DontDestroyOnLoad；

-OnDestroy(): 清理静态实例引用，防止残留；

**描述**

1. 泛型单例模式

通过 where T : SingletonMono<T> 的泛型约束，确保子类返回自身类型的单例实例。

1. 自动查找或创建实例

支持从场景中查找已有实例，若未找到则自动新建 GameObject 并附加组件，降低使用门槛。

1. 支持 Unity 生命周期函数

与 MonoBehaviour 保持兼容，子类可使用 Awake、Start、Update 等生命周期回调。

1. 跨场景持久化

默认使用 DontDestroyOnLoad 保证在多个场景间保持单例对象状态。

1. 生命周期接口支持

实现 IResetable 接口，便于在游戏状态切换或资源回收时统一重置管理类状态。

**优化方向**

1. 实例创建方式可扩展

当前自动创建的 GameObject 无法绑定到具体场景目录，可扩展支持资源预制体或父级设置。

1. Reset 调用权限未限制

同非 Mono 的 Singleton，Reset() 是 public 方法，容易被误用。可通过 Internal 接口或访问控制机制防止外部随意调用。

1. OnDestroy 与重载冲突风险

如果子类重写 OnDestroy() 且忘记调用 base.OnDestroy()，可能导致 \_instance 残留。