PureMVC框架源码分析电子笔记

一： **PureMVC 目录结构。**

分三大目录，共21一个类（接口）。

Core: 包含三大核心类： Model 、View、Controller

Interface： 包含十个接口。

Pattern: 其他辅助类。

Mediator、Proxy、SimpleCommand(MarocCommand)。

Notification: 消息封装类：

包含 Name、Body 、Type 三个重要的字段，本身这个类的作用，就是一种消息的封装载体。

Notifer: 传递消息使用，里面存在SendNotification() 三个方法重载类，本质是引用 Façade 类中 SendNotification()的方法。

**\*Observer: 对指定对象调用指定方法。**

核心方法： NotifyObserver() 就是这个类的核心方法，我们在应用框架开发时候，使用SnedNotifacation() 方法的消息，最终消息的执行，就是由这个方法最终完成。

二**： 三大核心类：**

Core 文件夹下： Model、View、Controller

* 分析三大核心类共同特性：

1: 都具备注册、查询(Controller类除外）、是否存在、移除方法。

2： 都是“线程安全”、“延迟加载”的单例类。

3： 核心类，存在大量虚方法，方便我们不改变框架源代码基础上，重构进一步丰富框架实现。

4： View与Model 类都会执行 OnRegister() ,OnRemove（） 虚方法。

* Model 类不具备处理消息的能力。

所以不能给模型层实现类发消息。

* Controller 类，具备两大功能： 1： 缓存所有控制层实现类对象。

2： 可以接受“命令消息”，且自动执行（继承ICommand 的类）的Excute() 方法。

* View 类，

分两大功能：1： 整个框架的消息中心。

2： 对于Mediator 做存储，查询，以及处理。

三： **入口与辅助类。**

Façade 入口类，就是框架三大核心类的一个简化入口。

Mediator : 视图层需要继承的父类。

SimpleCommand: 命令类

MecroCommand: 多个命令类的集合。

Proxy : 模型代理类。

Notify: 是 Mediator、SimpleCommand、Proxy父类，目的是方便我们做消息传递。

四： **分析PureMVC 消息的流转：**

1： GameStart 脚本，执行Start() 方法。

2： ApplicationFacade 实例化（三层的初始化方法被执行）

[例如： InitializeView() 都执行]

3： 执行 StartupApplication 控制层，执行 Excute() 方法。

4： 执行SendNotifacation() 。

5： 执行父类SimplyCommand 的父类Notifier.cs 中的 SendNotifacation()

6： Notifer 是引用的Facade 类中的方法。

7： Facade 类是PureMVC 的一个“空壳”，聚集了三大核心类中主要的对外公布的方法。

8： Facade 最终是引用核心类：View 中“消息执行”NotifyObservers() 方法.