

MVP 设计模式 (Android- 浅谈)

作者: [MainTheme](#)

原文链接: <https://ld246.com/article/1516979945809>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

MVP(Model-View-Presenter) 是总所周知MVC模式的一个演变,他们的主要目的都是划分模块职责,低模块耦合,易测试,提高代码复用,这里主要针对Android平台来简单分析MVP.

1.层级责任

Model: 负责数据的检索,持久化等操作

View: 负责UI的绘制和用户的交互

Presenter: 作为Model和View的中间协调部分,负责两者之间的业务逻辑处理

2.与MVC模式的区别

MVP模式与MVC模式从层级数据流向上来说一个主要的区别应该就是:MVC模式允许View层和Model层直接通讯.从图1和图2可以看到MVP和MVC的区别.

图1MVC模式中Model可以直接update data 到View层。所以当某个View的功能很复杂的时候,View Model的耦合度可能会很高(并且在android的开发中Activity通常会充当controller&view的角色,结果Activity就很臃肿).而MVP模式就没有这个问题,View会抽象出来一系列操作UI的接口(Model层也可以),Presenter拿到的都是其他两个层级的接口来做业务逻辑的处理.这样不仅可以使View和Model之间的耦合度降低,还可以更易得进行单元测试.

3.MVP的优缺点

优点:降低耦合, 层级职责更明显, 易于单元测试

缺点:造成类数量爆炸, 代码复杂度和学习成本高, 在某些场景下presenter的复用会产生接口冗余

-----后续补充 网站申请空间中