

HenCoder Plus 讲义

常见项目架构的示例及培养自己的架构思路

架构的作用

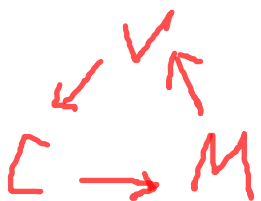
- 对于 Web 项目来说，他们的架构虽然大，但是架构设计也并没有太困难，因为有很多成熟的框架可以用。这些框架本身就提供了项目的架构，也就是说，你通过使用框架的方式，就把项目的架构搭建好了。
- 而对于 Android 开发来说，SDK 提供给我们的方案就已经可以写出完整的软件了。我们遵守 SDK 给我们定出的规则，已经能用比较稳定的代码写软件了。由于移动端的软件功能相对简单，所以写简单软件是不需要架构的。

但如果商业项目真的没有架构：

- 不同的人的风格略有不同，一个人写的代码结构，同事或者新人可能会看不懂；每个人的风格、代码结构不同，有必要进行定义一套规则。
- 当项目越做越大的时候，可能会面临「改不动」的问题：新功能加不进去，软件结构不好调整。
- 所以虽然移动端项目一般都很小，但是有一个稳定的架构，对商业项目还是有好处的。

MVC 和 MVP 到底有什么区别

- 实际上，MVP 和 MVC 并没有什么区别；而很多人所推崇的 MVP，其实才是 MVC。本身 MVC 就是一个宽松的概念，没有精确到分毫的严格规定。换个思路，我把 MVP 的 Presenter 这个类，改名叫 Controller，它应该叫 MVC 还是 MVP？（无所谓的啦）
- 换句话说，它们其实全都是 MVC。



一般我们在网文博客上，看到mvc的流程图都是像左边这个图那样，即View事件，调用C的逻辑，最终用M拿到数据，最后M去调用V更新。但是实际上，我们用所谓的mvc时，并没有M去调用V，而是M返回数据给C，C再去调用V更新的。其实这个就很像我们讲的MVP，就是了。

再说说那个所谓的MVC(其实也是MVP)吧，它的缺点可能就是责任划分不明确，xml当成是view，activity当成是Controller，同时能activity还有一部分view的功能，是这里导致了代码的耦合太严。

也就是要解耦了，xml + activity 兼顾了v和C的功能，从中抽取出了Presenter作为指挥者，然后xml和activity一起作为view作用了。也就是得到了我们的MVP。

世人对于这些框架的理解其实都不一样的，但对我们来说，最重要的是要知道人们的观点是什么样的（这个方便我们和别人讨论），以及 MVC 和 MVP 架构的优势和实现方式到底是什么（这个方便我们做事）。

优势：就是逻辑的独立；

其实，上面是从xml+activity中解耦出p。想想，还有另一个思路，就是从activity中抽出view也是可以的，然后activity作为P，xml + 新的模块作为view。也是可以，当然这个还是MVP，只是不同的实现罢了。

实现方式：课堂里已经演示过。

- 记住：把所有业务逻辑和界面更新操作都写进 Activity 的写法不叫 MVC。

用所谓的普遍的mvp架构模式，好处呢？面向接口IView，复用性好，可扩展性，也就方便测试。

MVVM 是什么

MVVM 就是一个加了数据的 MVP（或者 MVC）。

什么是数据绑定：

mvvm: Model + View + ViewModel。其实就是在MVP的基础上，加了个数据自动绑定的功能。是一个变相的mvp。

- 外部数据（数据库数据、网络数据）、内存数据（Java 代码中的变量）、表现数据（界面中展示的数据）中的外部表现数据和内存数据互相自动更新。
- 另外，MVVM 有时候还可以给你的内存数据和数据库数据做关联监听，让你的这三种数据实现进一步的联动。

MVVM 和 MVC、MVP 在定位上的区别

- MVC MVP 的架构性质更强：它提供设计规范
- 而 MVVM 是一个框架，像一个库：它提供数据绑定的功能特性

MVVM 和 Android Jetpack

和 MVVM 相关的 Jetpack 组件是 DataBinding，而不是 ViewModel

培养架构思路的关键

- 一个字：拆
- 要在项目里面培养，而不是看别人的文章，或者听别人讲课（哈哈，听我讲也不行，真的得练）。

客户端要架构师有点多余，但是架构思路一定要有。
架构就是拆，拆分成一个一个模块，达到解耦、模块化，解耦成模块 + 接口。

问题和建议？

课上技术相关的问题，都可以去群里和大家讨论，对于比较通用的、有价值的问题，可以去我们的知识星球提问。

具体技术之外的问题和建议，都可以找丢物线（微信：diuwuxian），丢丢会为你解答技术以外的一切。



觉得好？

如果你觉得课程很棒，欢迎给我们好评呀！<https://ke.qq.com/comment/index.html?cid=381952>

一定要是你真的觉得好，再给我们好评。不要仅仅因为对丢物线的支持而好评（报名课程已经是你最大的支持了，再不够的话 B 站多来点三连我也很开心），另外我们也坚决不做好评返现等任何的交易。我们只希望，在课程对你有帮助的前提下，可以看到你温暖的评价。

更多内容：

- 网站：<https://hencoder.com>；<https://kaixue.io>

- 各大搜索引擎、微信公众号、微博、知乎、掘金、哔哩哔哩、YouTube、西瓜视频、抖音、快手、微视：统一账号「扔物线」，我会持续输出优质的技术内容，欢迎大家关注。
- 哔哩哔哩快捷传送门：<https://space.bilibili.com/27559447>

大家如果喜欢我们的课程，还请去扔物线的哔哩哔哩，帮我素质三连，感谢大家！