Android技术路线学习路线

Part.1 Kotlin

- Tour of Kotlin!
  Kotlin官方文档引导介绍模块，快速了解语法，英文
- 中文翻译文档
  Kotlin文档翻译镜像
- [在线编辑器](https://play.kotlinlang.org/)
  Kotlin在线编辑器
- Java工程师Kotlin快速学习视频
  着重注意视频中Kotlin字节码反编译为Java代码的比对
- 《Kotlin核心编程》
  数据类型、Lambda、设计模式
-

Part.2 Google Developer ‘s Guides 1（@Feb 2024）

2.1. 官方文档指引

> **Important**
>
> 书籍、网络可能存在大量过期知识，开发必须以第一方文档作为主要知识源。

> Google第一方文档：基础知识、官方推荐最佳实践CodeLab：
> 开发者指南

2.2. 基础知识复盘

- **应用基础知识** 基础知识
  - 四种类型应用组件：（Activity/服务/广播接收器/Content provider）
  - intent（显式 intent/隐式 intent）
  - AndroidManifest.xml

2.3. 应用架构

- **应用架构，重要基础章节** 架构

2.3.1 架构最佳实践内容节选：

> 一种常见的错误是在一个 Activity 或 Fragment 中编写所有代码。这些基于界面的类应仅包含处理界面和操作系统交互的逻辑。

> 通过数据模型驱动界面（最好是持久性模型）,持久性模型是理想之选，原因如下：如果 Android 操作系统销毁应用以释放资源，用户不会丢失数据。当网络连接不稳定或不可用时，应用会继续工作。

> 基于架构原则，每个应用应至少有两个层：
> 界面层 - 在屏幕上显示应用数据。
> 数据层 - 包含应用的业务逻辑并公开应用数据。
> 您可以额外添加一个名为“网域层”的架构层，以简化和重复使用界面层与数据层之间的交互。

> **Important**
>
> ### 架构的优势
>
> 在应用中实现良好的架构会为项目和工程团队带来诸多好处：
>
> - 提高整个应用的可维护性、质量和稳健性。
> - 允许应用扩缩。尽可能减少代码冲突，使更多人和更多团队可以为同一代码库做贡献。
> - 有助于新手上手。架构能使您的项目保持一致性，让团队中的新成员可以快速上手，并在更短时间内提高效率。
> - 更易于测试。良好的架构鼓励使用更简单的类型，这些类型通常更易于测试。
> - 可以使用明确定义的流程有条理地调查 bug。

2.3.2 架构组件：

- 界面层View
  - 视图绑定viewBinding
  - 数据绑定dataBinding（XML）
  - 生命周期 androidx.lifecycle
  - Paging
- 数据层Model
  - DataStore
    - Preferences DataStore
    - Proto DataStore
  - WorkManager

2.3.3 应用入口：

- Activity
- Navigation
- fragment

2.3.4 依赖项注入 (DI)：难

Part.3 文章视频集合

https://www.bilibili.com/video/BV1kz4y1a7z5

https://www.bilibili.com/video/BV1K94y177nw

https://www.bilibili.com/read/cv19465431/?spm_id_from=333.999.0.0

https://juejin.cn/post/7022624191723601928

https://juejin.cn/post/7278498472860909605

Android的MVI架构页面如何驱动

倒计时示例

Part.4 项目

https://github.com/HuJianChong/MyWanAndroid.git
一个非标准项目，但是开发已经分层