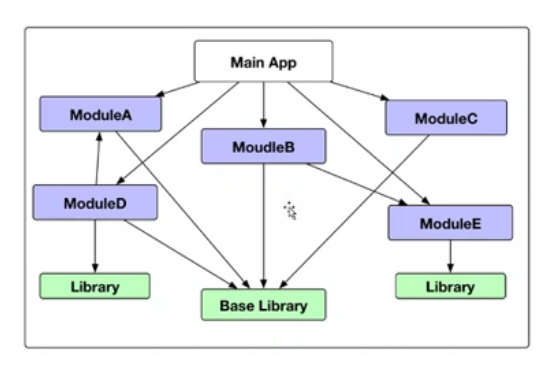
**Android 大型项目组件化架构**

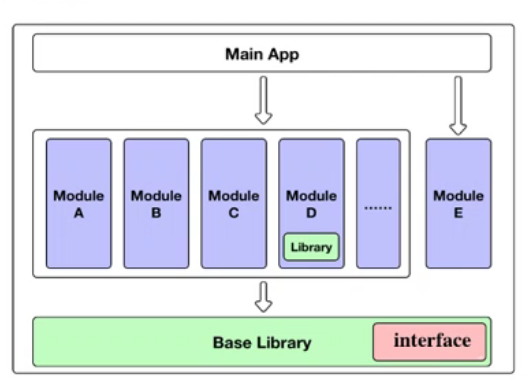
1. 一般的App项目架构



1. 什么是组件化架构
2. 项目发展到一定规模程度，随着业务规模和人员的扩张，代码会越来越多，越来越臃肿，维护难度越来越大，成本更高。这时候就必须将项目各个模块拆分成颗粒度更小的组件，达到降低耦合，代码复用，提高开发和维护效率等目的

2. 相比模块而言，组件偏向功能导向，颗粒度更小。模块偏向业务导向，一般情况，一个模块可以包含多个组件。

1. 组件化架构使用场景
2. 几乎所有大型项目都可以使用，尤其适合业务和功能复杂，模块众多的大型项目
3. 中大型互联网公司&企业，实现团队协作开发维护项目，提高研发效率和质量。
4. 组件化架构优点
5. 实现代码解耦，功能模块复用，代码架构整洁清晰
6. 赋能团队项目工程协作开发，研发人员之间职责明确，互不干扰，提高效率和质量
7. 加速编译和开发效率，降低项目维护难度和成本。
8. 可以灵活的对业务模块进行组装和拆分。
9. 组件化项目架构框架



1. 组件化需要解决的问题
2. 统一管理项目开发环境

使用gradle.properties配置编译环境属性，例如compileSdkVersion等

1. 组件需要满足单独运行与调试的要求

可以作为一个独立的App运行和调试，也可以作为lib库

1. 实现组件之间的揭秘跳转和数据传递

为组件创建中间层接口，每个组件需要提供各自接口的实现

1. 组件相互之间成调试，包括赖部分组件

在gradle.properties中配置组件是否独立运行属性，通过调整属性为false来实现集成调试

1. 组件中Application的使用

在app组件中通过反射实例化组件的Application对象

1. 组件之间实现代码隔离，资源文件隔离

在app组件中使用runtimeOnly依赖组件实现代码隔离，在组件中使用resourcePrefix ‘组件名\_’实现资源文件的隔离，解决资源冲突问题

1. 附录

目前gradle3.0.0的功能和代码隔离效果见下图

