

包 (Package)

» 包是什么？

- 包用来对类分组
- 系统中，有些类互相依赖即互相协作完成特定功能。
- 系统中，有些类不知道彼此

» 将有关联的类放在一个包里很自然

- 什么样的类应该放在一个包里？
- 包之间的依赖关系是什么样的？

6个设计原则

» 包的内聚性原则

- 重用发布等价原则
- 共同重用原则
- 共同封闭原则

解决什么样的类放到一个包里的问题

6个设计原则

» 包的耦合度原则

- 无环依赖原则
- 稳定依赖原则
- 稳定抽象原则

解决包之间关系的问题。

重用发布等价原则：REP

» REP (The Reuse-Release Equivalence Principle) :

- 重用的粒度与发布的粒度等价。

共同重用原则：CRP

» CRP (The Common-Reuse Principle) :

- 包中的类一起重用。如果你重用包中一个类，则要重用包中所有的类。

» 一个包中的类彼此依赖。

» 没有关系的类不应该放在同一个包中。

共同封闭原则：CCP

» CCP (The Common-Closure Principle) :

- 一个包中的所有类应该对同一种需求变化封闭。影响包的变化影响包中所有的类，但不影响其他包。

» 一个包不应有多个变化的原因。

无环依赖原则：ADP

- » ADP (The Acyclic-Dependencies Principle):
 - 在包依赖关系图中不允许环存在
- » 早后综合征 (Morning-after syndrome)
- » 2个解决方案:
 - 每周构建 (The Weekly Build)
 - ADP

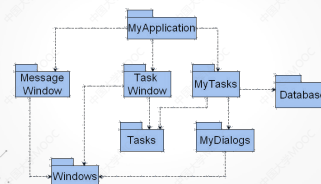
每周构建

- » 所有的开发人员在每周的前四天工作于自己的代码库，不需要对彼此的代码集成。
- » 周五，整合开发的所有代码并建立系统。
- » 适用于中型项目

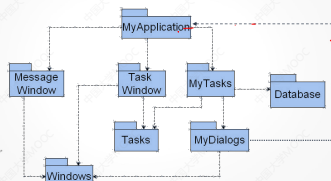
消除包依赖图中的环

- » 将开发环境划分为可发布的包。
- » 为了采用自底向上增量式开发，不允许环存在。
- » 有向无环图 (DAG, Directed Acyclic Graph) : 无论从哪个包开始，都不可能沿着依赖关系回到起点包。这种结构没有环。

有向无环图(DAG)



包依赖关系图有环



解除环

- » 解除环并将依赖关系图恢复为DAG的两种方案:
 - 应用依赖倒置原则
 - 创建一个MyDialog和MyApplication都依赖的新包。

