

GeekBand 极客班

互联网人才加油站!

C++设计模式

www.geekband.com

GeekBand 极客班 互联网人才+加油站：

极客班携手 网易云课堂，针对热门IT互联网岗位，联合业内专家大牛，紧贴企业实际需求，量身打造精品实战课程。

专业课程

+

项目碾压

+

习题&辅导

- | | | |
|------------|----------------|----------|
| • 顶尖大牛亲授 | • 紧贴课程内容 | • 学前导读 |
| • 贴合企业实际需求 | • 全程实战操练 | • 周末直播答疑 |
| • 找对重点深挖学习 | • 作品就是最好的PASS卡 | • 定期作业点评 |
| | | • 多项专题辅导 |



www.geekband.com

C++设计模式

Mediator 中介者

李建忠

GeekBand 极客班

“接口隔离” 模式

➤在组件构建过程中，某些接口之间直接的依赖常常会带来很多问题、甚至根本无法实现。采用添加一层间接（稳定）接口，来隔离本来互相紧密关联的接口是一种常见的解决方案。

➤典型模式

- Façade
- Proxy
- Adapter
- Mediator

Mediator 中介者

动机 (Motivation)

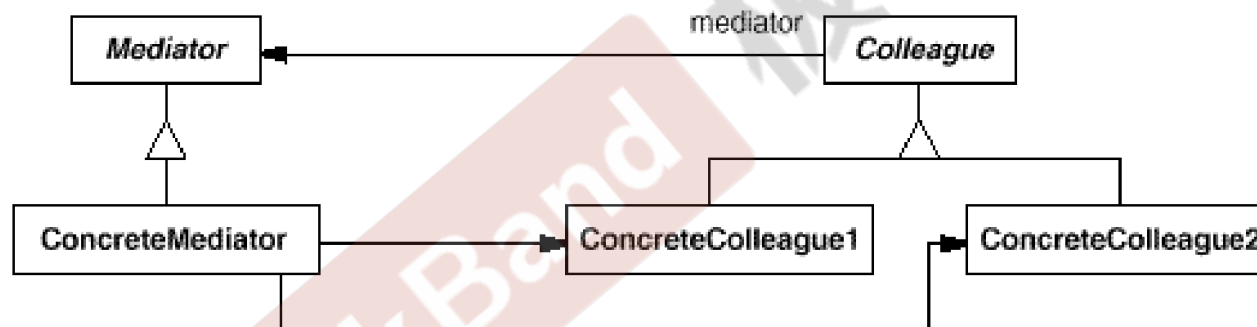
- 在软件构建过程中，经常会出现多个对象互相关联交互的情况，对象之间常常会维持一种复杂的引用关系，如果遇到一些需求的更改，这种直接的引用关系将面临不断的变化。
- 在这种情况下，我们可使用一个“中介对象”来管理对象间的关联关系，避免相互交互的对象之间的紧耦合引用关系，从而更好地抵御变化。

模式定义

用一个中介对象来封装（封装变化）一系列的对象交互。中介者使各对象不需要显式的相互引用(编译时依赖 → 运行时依赖)，从而使其耦合松散（管理变化），而且可以独立地改变它们之间的交互。

——《设计模式》GoF

结构 (Structure)



要点总结

- 将多个对象间复杂的关联关系解耦，Mediator模式将多个对象间的控制逻辑进行集中管理，变“多个对象互相关联”为“多个对象和一个中介者关联”，简化了系统的维护，抵御了可能的变化。
- 随着控制逻辑的复杂化，Mediator具体对象的实现可能相当复杂。这时候可以对Mediator对象进行分解处理。
- Façade模式是解耦系统间（单向）的对象关联关系；Mediator模式是解耦系统内各个对象之间（双向）的关联关系。