

GeekBand 极客班

互联网人才 + 油站!

# C++ 设计模式

[www.geekband.com](http://www.geekband.com)

## GeekBand 极客班 互联网人才+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](http://www.geekband.com)

C++设计模式

# Builder 构建器

李建忠

GeekBar 极客班

## “对象创建” 模式

- 通过 “对象创建” 模式绕开new，来避免对象创建（ new ）过程中所导致的紧耦合（依赖具体类），从而支持对象创建的稳定。它是接口抽象之后的第一步工作。
- 典型模式
  - Factory Method
  - Abstract Factory
  - Prototype
  - Builder

# Builder 构建器

## 动机 ( Motivation )

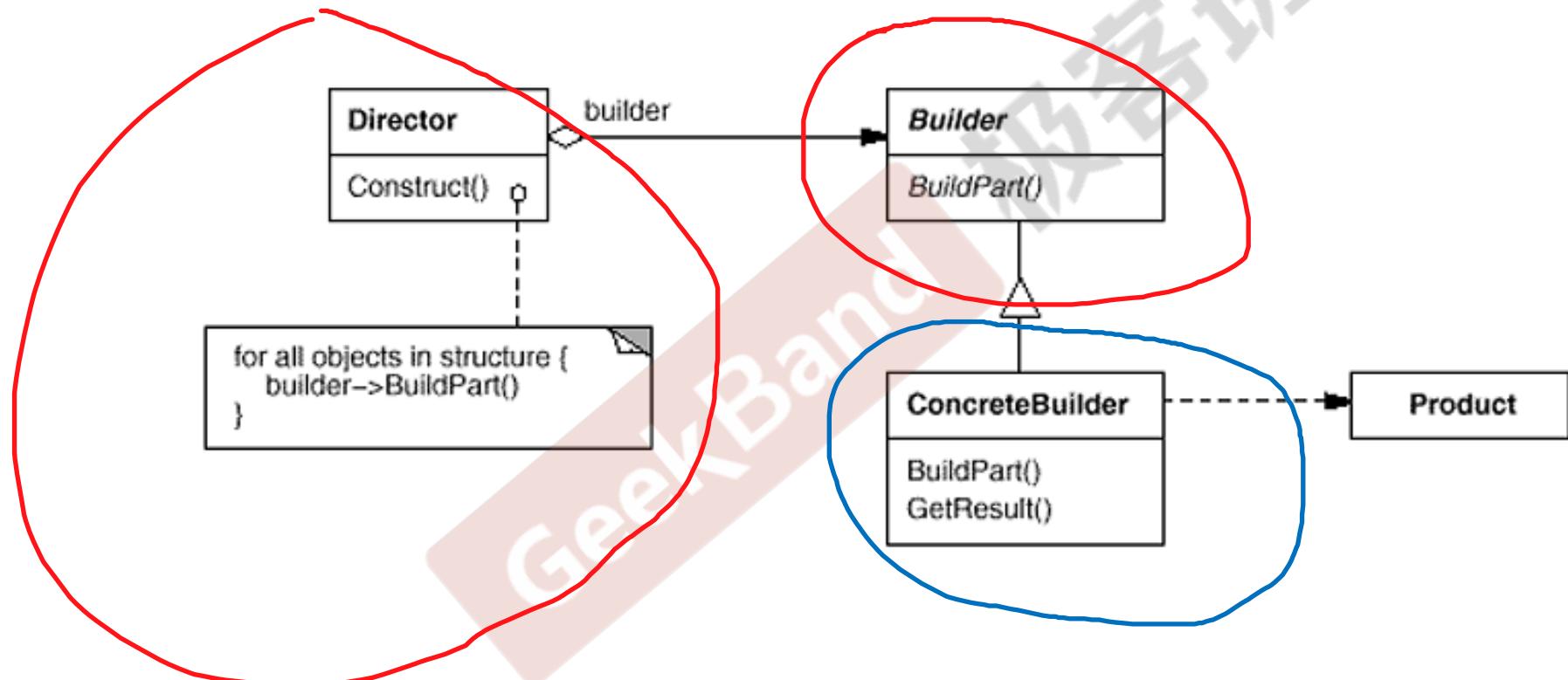
- 在软件系统中，有时候面临着“一个复杂对象”的创建工作，其通常由各个部分的子对象用一定的算法构成；由于需求的变化，这个复杂对象的各个部分经常面临着剧烈的变化，但是将它们组合在一起的算法却相对稳定。
- 如何应对这种变化？如何提供一种“封装机制”来隔离出“复杂对象的各个部分”的变化，从而保持系统中的“稳定构建算法”不随着需求改变而改变？

## 模式定义

将一个复杂对象的构建与其表示相分离，使得同样的构建过程(稳定)可以创建不同的表示(变化)。

——《设计模式》 GoF

## 结构 ( Structure )



## 要点总结

- Builder 模式主要用于“分步骤构建一个复杂的对象”。在这其中“分步骤”是一个稳定的算法，而复杂对象的各个部分则经常变化。
- 变化点在哪里，封装哪里—— Builder模式主要在于应对“复杂对象各个部分”的频繁需求变动。其缺点在于难以应对“分步骤构建算法”的需求变动。
- 在Builder模式中，要注意不同语言中构造器内调用虚函数的差别（C++ vs. C#）。