

GeekBand 极客班

互联网人才加油站!

# C++设计模式

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

C++设计模式

# Flyweight 享元模式

李建忠

Geek Band 极客班

## “对象性能” 模式

➤面向对象很好地解决了“抽象”的问题，但是不可避免地要付出一定的代价。对于通常情况来讲，面向对象的成本大都可以忽略不计。但是某些情况，面向对象所带来的成本必须谨慎处理。

- 典型模式
  - Singleton
  - Flyweight

# Flyweight 享元模式

## 动机 ( Motivation )

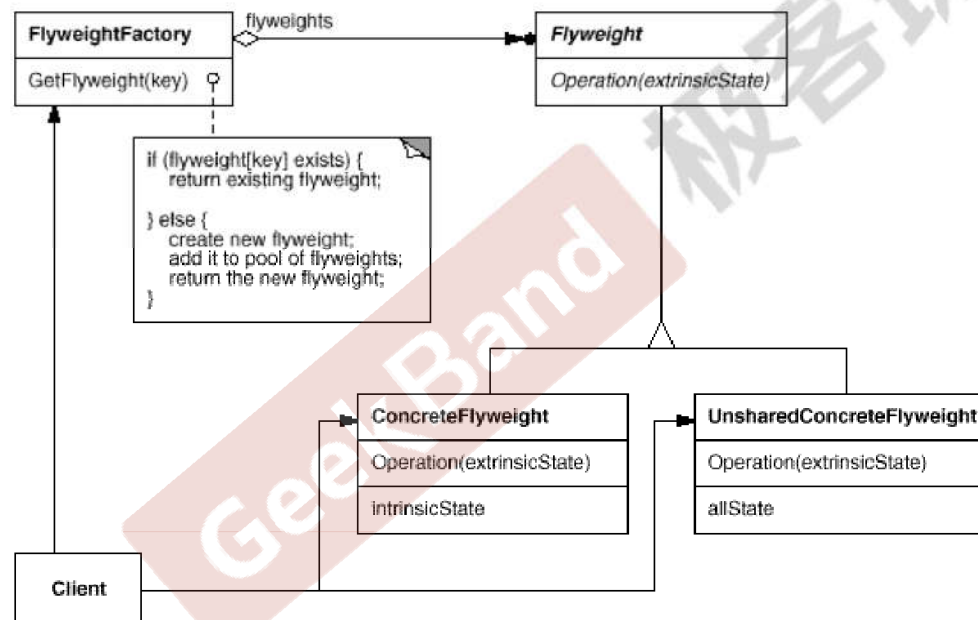
- 在软件系统采用纯粹对象方案的问题在于大量细粒度的对象会很快充斥在系统中，从而带来很高的运行时代价——主要指内存需求方面的代价。
- 如何在避免大量细粒度对象问题的同时，让外部客户程序仍然能够透明地使用面向对象的方式来进行操作？

## 模式定义

运用共享技术有效地支持大量细粒度的对象。

——《设计模式》GoF

## 结构 ( Structure )





## 要点总结

- 面向对象很好地解决了抽象性的问题，但是作为一个运行在机器中的程序实体，我们需要考虑对象的代价问题。Flyweight主要解决面向对象的代价问题，一般不触及面向对象的抽象性问题。
- Flyweight采用对象共享的做法来降低系统中对象的个数，从而降低细粒度对象给系统带来的内存压力。在具体实现方面，要注意对象状态的处理。
- 对象的数量太大从而导致对象内存开销加大——什么样的数量才算大？这需要我们仔细的根据具体应用情况进行评估，而不能凭空臆断。