



第8章 设计原则

8.8 接口隔离原则

刘其成 计算机与控制工程学院 ytliuqc@163.com 2018-09

软件设计与体系结构





8.8.1 概念

定义

■ 接口隔离原则

(Interface Segregation Principle, ISP)

- 定义
 - -客户端不应该依赖那些它不需要的接口。
 - -一旦一个接口太大,则需要将它分割成一些更细小的接口,使用该接口的客户端仅需知道与之相关的方法即可。

接口隔离原则分析

- 接口隔离原则是指使用多个专门的接口,而不使用单一的总接口。
 - -接口<mark>仅仅提供客户端需要的行为</mark>,即所需的方法,客户端不需要的行为则隐藏起来,应当为客户端提供尽可能小的单独的接口,而不要提供大的总接口。
- 从客户类的角度看,接口隔离原则是指一个类对另外一个类的依赖性应当是建立在最小的接口上的。

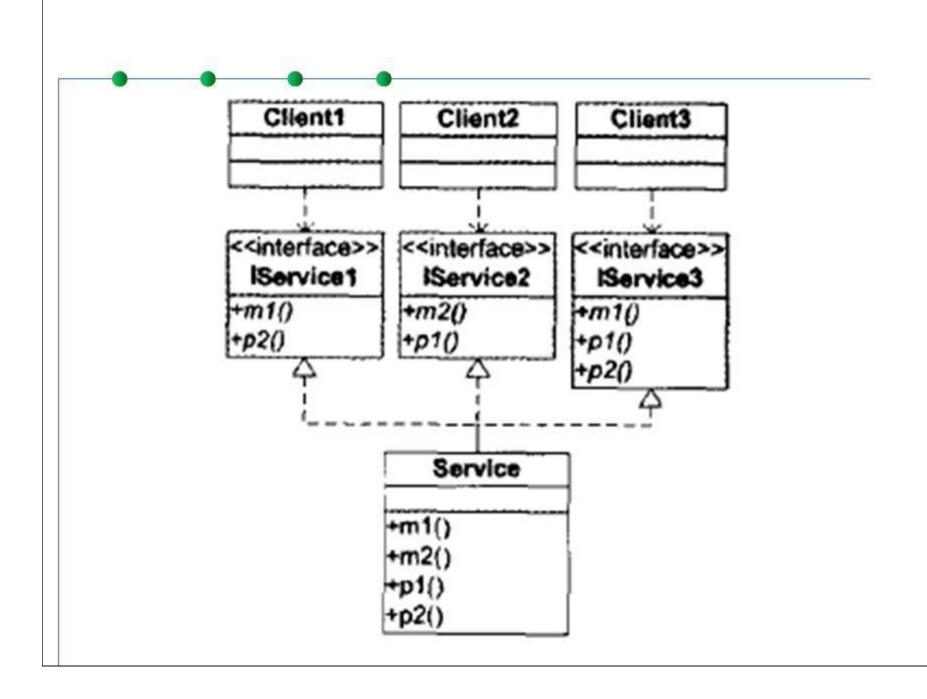
- 使用接口隔离原则拆分接口时,首先必须满足单一 职责原则,将一组相关的操作定义在一个接口中, 且在满足高内聚的前提下,接口中的方法越少越好。
- 可以在进行系统设计时采用定制服务的方式,即为不同的客户端提供宽窄不同的接口,只提供用户需要的行为,而隐藏用户不需要的行为。

角色隔离原则

- 接口通常指一个类所提供的所有方法的特征集合,这是一种在逻辑上才存在的概念。
 - 这样,接口的划分就直接带来类型的划分。
- 如果把一个接口看成电影中的一种角色,接口的实现就可以看成这个角色由哪一个演员来演。
 - 因此,一个接口应当只代表一个角色,而不能代表多个 角色。
 - 如果有多个角色,那么每一个角色都应当由一个特定的接口代表。
 - 这种角色划分的原则叫做角色隔离原则。
- 每一个接口应该承担一种相对独立的角色,不多不少,不 干不该干的事,该干的事都要干。

定制服务

- ■可将接口狭义地理解为Java语言的接口类型Interface,例如java.lang.Runnable就是一个Java接口
 - 这样,接口隔离原则就是指针对不同的客户端,为同一个角色提供宽、窄不同的接口。
- 这可以看作叫做定制服务(Customized Service)。
 - -例如,如果对于角色Service,三个不同的客户端需要不同的服务,系统应该针对三个客户端分别提供了三个不同的Java接口——IServicel,IService2和IService3
- 每一个Java接口都仅仅提供对应的客户端需要的行为, 而客户端不需要的行为没有放到接口中。这是适配器模式 的应用。



接口的污染

- 总之,应当将多个不同的角色交给不同的接口,而不应当都交给同一个接口。
- 不要将一些看上去差不多的甚至是没有关系的接口 合并,这样会形成过于臃肿的大接口,称为接口的 污染。
- 所以,准确而恰当地划分角色以及角色所对应的接口,是面向对象的设计的一个重要的组成部分。

迪米特法则

- 迪米特法则要求任何一个软件实体尽量不要与外界通信,即使必须进行通信也应当尽量限制通信的广度和深度。
- 显然,接口隔离原则不向客户端提供不需要提供的行为,是符合迪米特法则的。

软件设计与体系结构

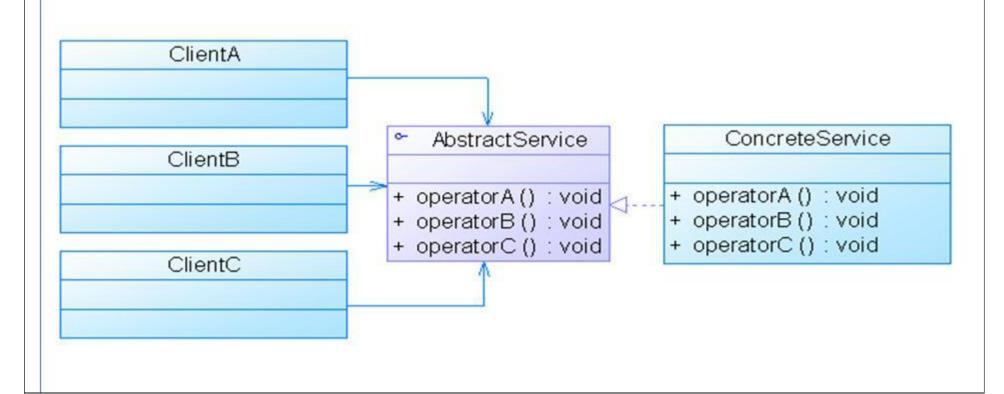




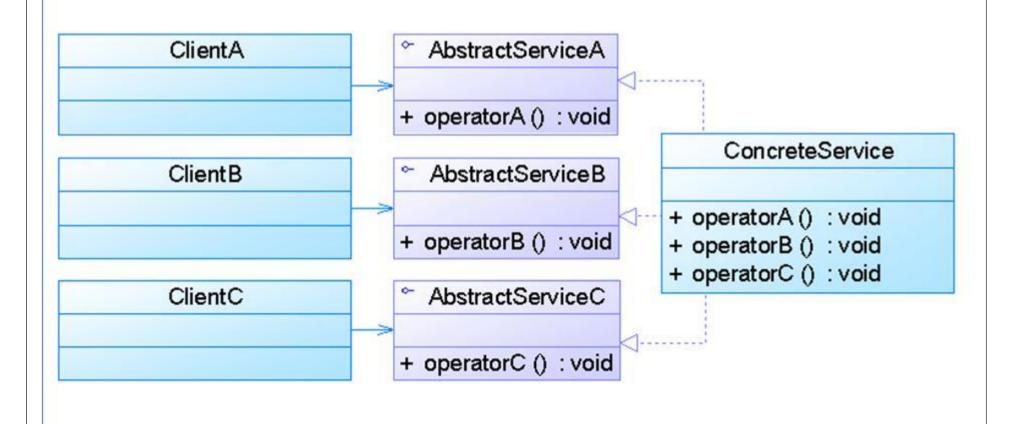
8.8.2 实例

多客户系统

■下图展示了一个拥有多个客户类的系统,在系统中定义了一个巨大的接口(胖接口) AbstractService来服务所有的客户类。



• 可以使用接口隔离原则对其进行重构。

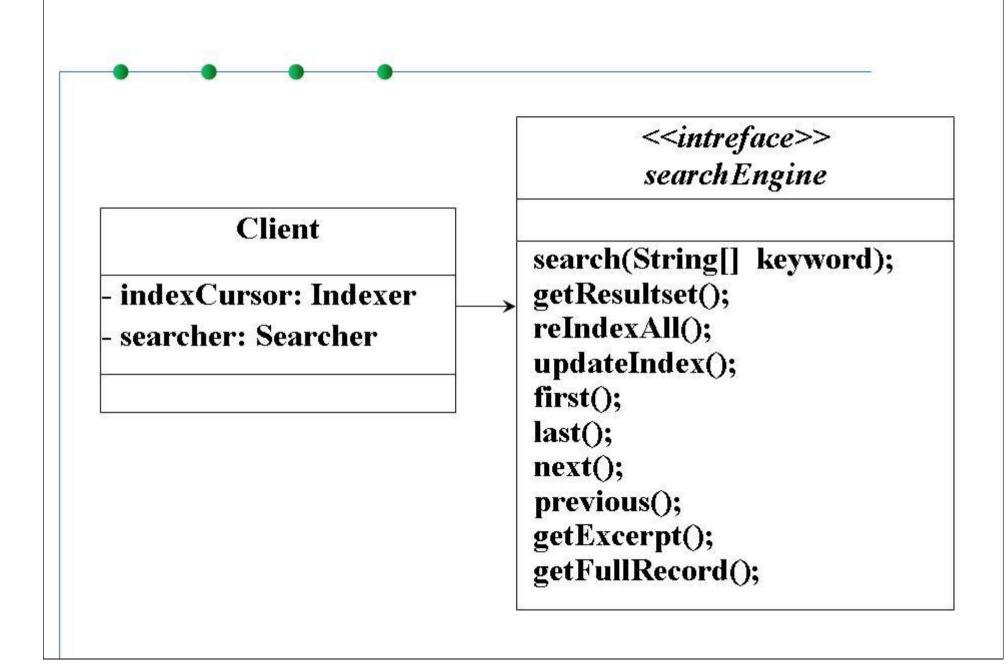


2. 全文查询引擎

- 从代码重构的角度讨论怎样将一个臃肿的角色重新 分割成更为合适的较小角色。
- (1) 全文查询引擎的系统设计
- 以一个网站的全文查询引擎的系统设计为例。
- 一个动态的资料网站将大量的文件资料存储在文件 中或关系数据库里面,用户可以通过输入一个和数 个关键词进行全网站的全文搜索。
- 这个搜索引擎需要维持一个索引库,在本例子里面索引库以文本文件方式存于文件系统中。
- 源数据被修改、删除或增加时,搜索引擎要做相应的动作,以保证引擎的索引文件也被相应地更新。

(2) 反面例子

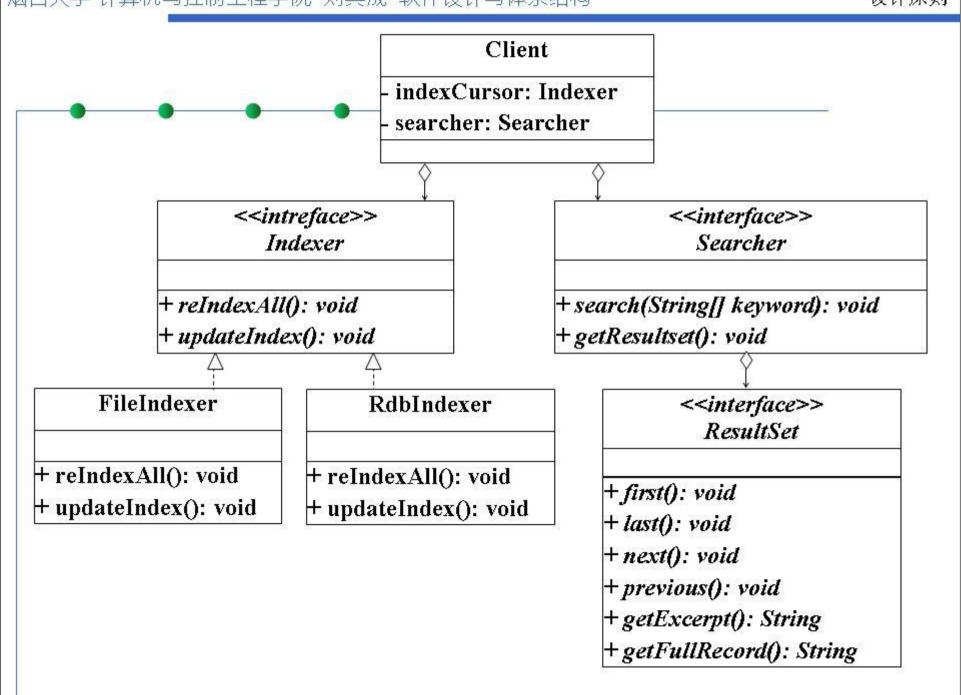
- 首先,下图所示为一个不好的解决方案。
- 一个接口负责所有的操作,从提供搜索功能到建立索引的功能,甚至包括搜索结果集合的功能均在一个接口内提供。
- 这个解决方案违反了角色分割原则,把不同功能的接口放在一起,由一个接口给出包括搜索器角色、索引生成器角色以及搜索结果集角色在内的所有角色。



```
interface search Engine {
  void search(String[] keyword);
  void getResultset();
  void reIndexAll();
  void updateIndex();
  void first();
  void last();
  void next();
  void previous();
  String getExcerpt();
  String getFullRecord();
```

(3) 角色的分割

- 那么, 遵守接口隔离原则的做法是怎样的呢?
- 如图所示。
- ■可以看出,搜索引擎的功能被分割为三个角色:
 - -搜索器角色
 - -索引生成器角色
 - -搜索结果集角色



搜索器角色

- 搜索器角色是与索引生成器角色完全不同的角色 ,它提供用户全文搜索功能。用户传进一些关键 字,搜索器角色则返回一个ResultSet对象。
- 设计Searcher接口,对应搜索器角色。该接口提供用户全文搜索功能,用户输入关键字,返回一个搜索结果集对象。

```
interface Searcher{
  void search(String[] keyword);
  void getResultset();
}
```

索引生成器

- ■设计Indexer接口,对应索引生成器角色。
- 该接口针对文件类型的数据和关系数据库的数据 生成全文索引,并可以更新索引。

```
interface Indexer{
  void reIndexAll();
  void updateIndex();
}
```

- ■索引生成器角色,由于索引生成因数据的格式不同而不同,故分为RdbIndexer和FileIndexer两种实现。
 - -FileIndexer类代表对诸如*.txt, *.html, *.doc 以及*.pdf等文件类型的数据生成全文索引,
 - -RdbIndexer则针对关系数据库的数据进行全文 索引生成。
- 这两个实现扮演的同为索引生成器角色,就好像 扮演同样角色的两个不同演员一样。

搜索结果集角色

- ■搜索结果集角色就是ResultSet。
- 它给用户提供对集合进行迭代走访的功能
 - -first()将光标移到集合的第一个元素
 - -last()将光标移到集合的最后一个元素
 - -next()将光标移到集合的下一个元素
 - -previous()将光标移到集合的前一个元素
 - -getExcerpt()则返回当前记录的摘要
 - -getFullRecord()则将记录的全文返回。

■ 设计ResultSet接口,对应搜索结果集角色,该接口给用户提供访问结果集内容的功能。

```
interface ResultSet{
  void first(); //将光标移到集合的第一个元素
  void last(); //将光标移到集合的最后一个元素
  void next(); //将光标移到集合的下一个元素
  void previous();//将光标移到集合的前一个元素
  String getExcerpt(); //返回当前记录的摘要
  String getFullRecord(); //将记录的全文返回
}
```

思考题

- 1. 简述可维护性复用的概念。如何实现可维护性复用?
- 2. 开-闭原则的含义是什么?分析某系统是否符合开-闭原则,如果不满足则进行重构。
- 3. 里氏代换原则的含义是什么?分析某系统是否符合里氏代换原则,如果不满足则进行重构。
- 4. 依赖倒转原则的含义是什么?分析某系统是否符合依赖倒转原则,如果不满足则进行重构。
- 5. 合成/聚合复用原则的含义是什么?按照合成/聚合复用原则重构某系统。
- 6. 迪米特法则的含义是什么?在迪米特法则中,对于一个对象,其朋友包括哪几种情况?分析某系统是否符合迪米特法则,如果不满足则进行重构。
- 7. 单一职责原则的含义是什么?分析某系统是否符合单一职责原则,如果不满足则进行重构。
- 8.接口隔离原则的含义是什么?分析某系统是否符合接口隔离原则,如果不满足则进行重构。
- 9. 简单工厂模式和工厂方法模式等设计模式与主要的设计原则的关系是什么?通过实例进行说明。





谢谢

2018年11月6日