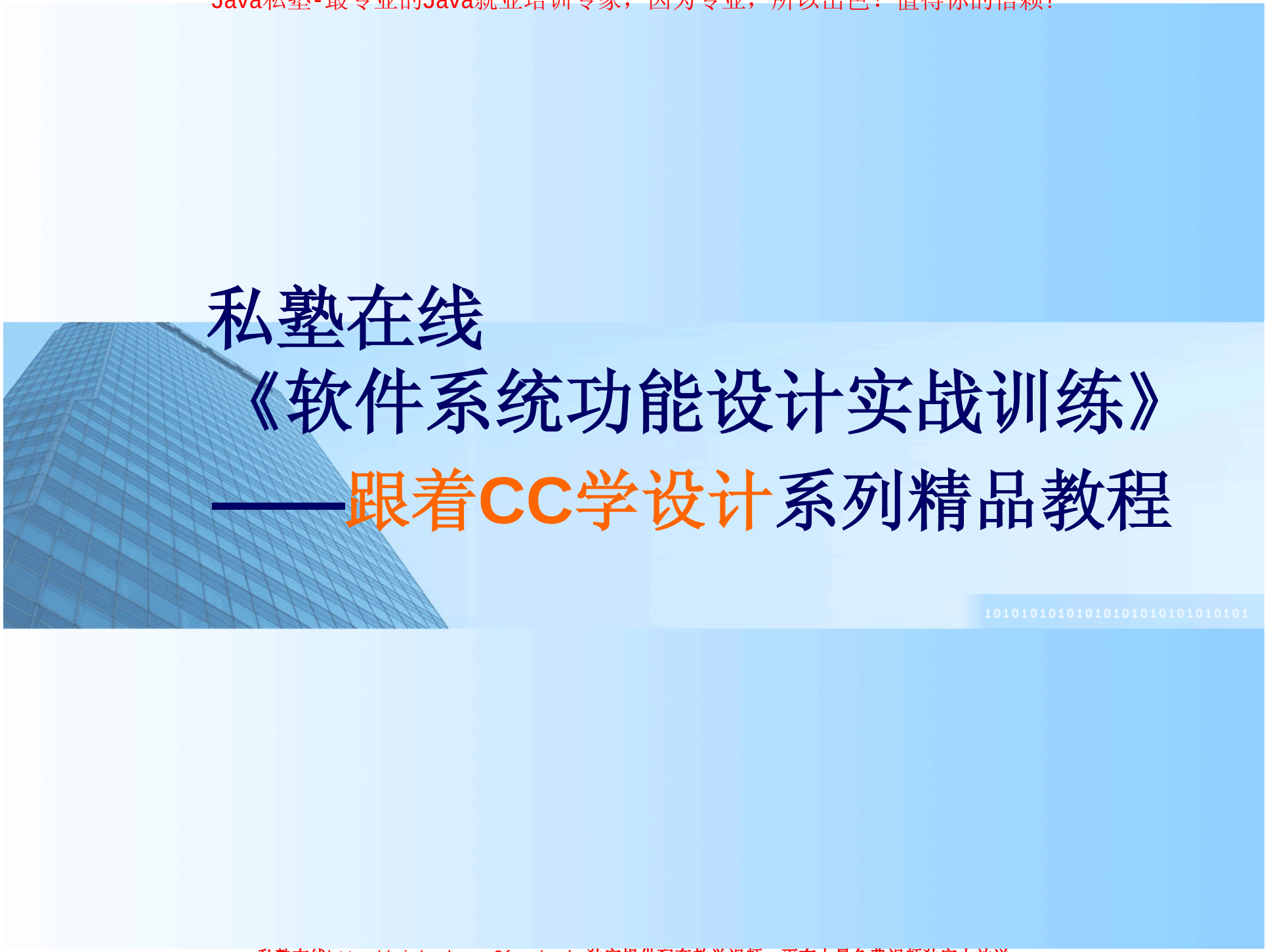


私塾在线
《软件系统功能设计实战训练》
——跟着CC学设计系列精品教程

本周设计作业

- 本周设计作业的项目背景：

  成本核算管理——来自某在线旅游系统

- 学习目标：

  （1）在实战中练习设计的理念和方法
  （2）学习如何动态的组装所需功能
  （3）学习基本的领域设计知识
  （4）综合应用享元模式、装饰模式，包含着这些模式但不限于这些模式

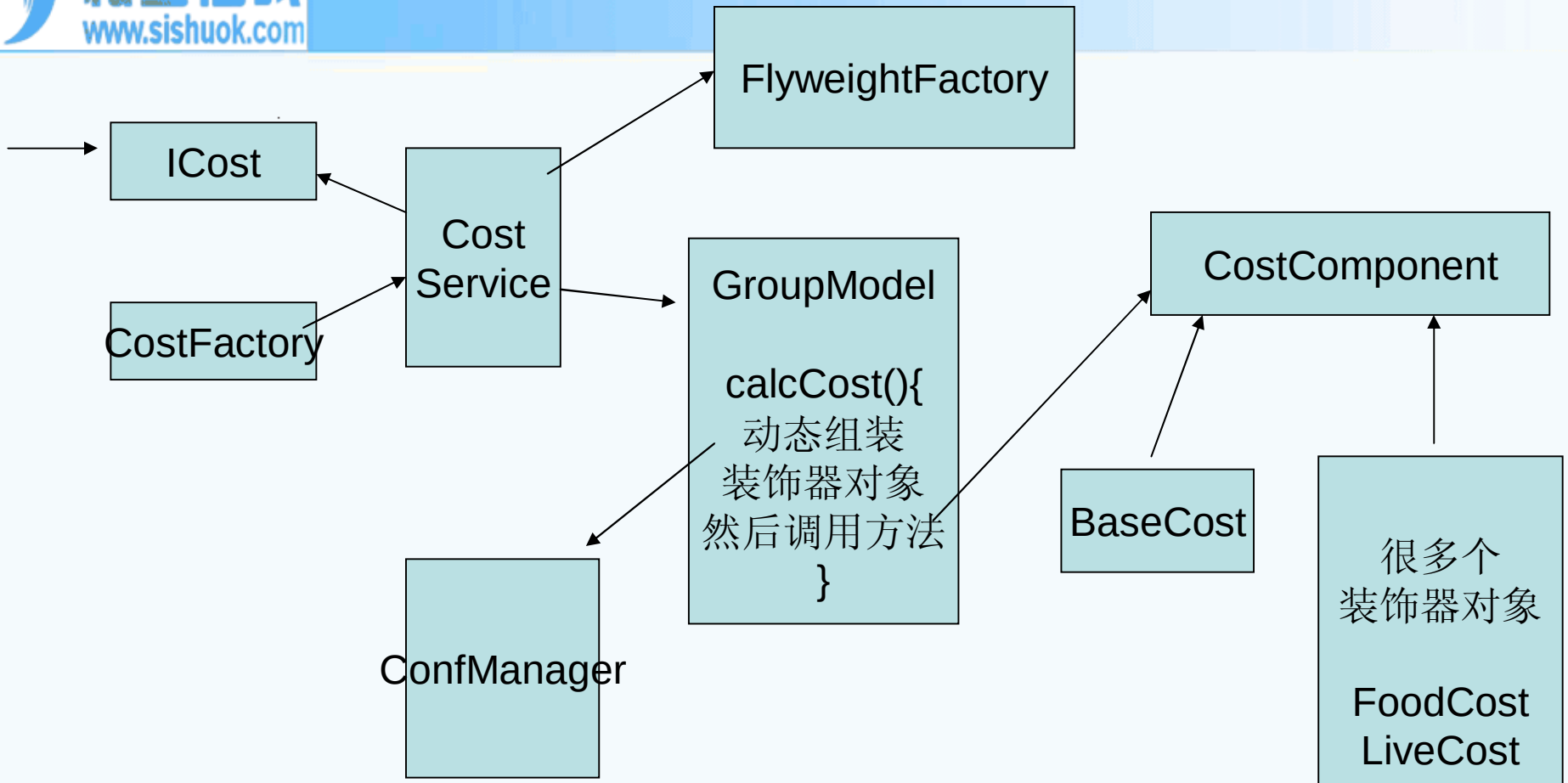
  该设计方式可应用于多种有类似功能的系统，比如：奖金计算、佣金计算、报价计算……

- 基本功能

  1：成本核算涉及到很多个方面，要能计算一个旅游团的总体成本，包括：交通成本、饮食成本、住宿成本、导游人员成本等

  2：要考虑扩展，因为还有很多不同的成本计算，比如：营销成本、前期的线路设计成本、税收成本等

  3：不同的团，需要计算的成本项是不一样的，比如：有些团是乘火车的，有些团是乘飞机的，那么交通成本的计算方式就不一样了，这次不引入这个复杂性。



本周设计作业

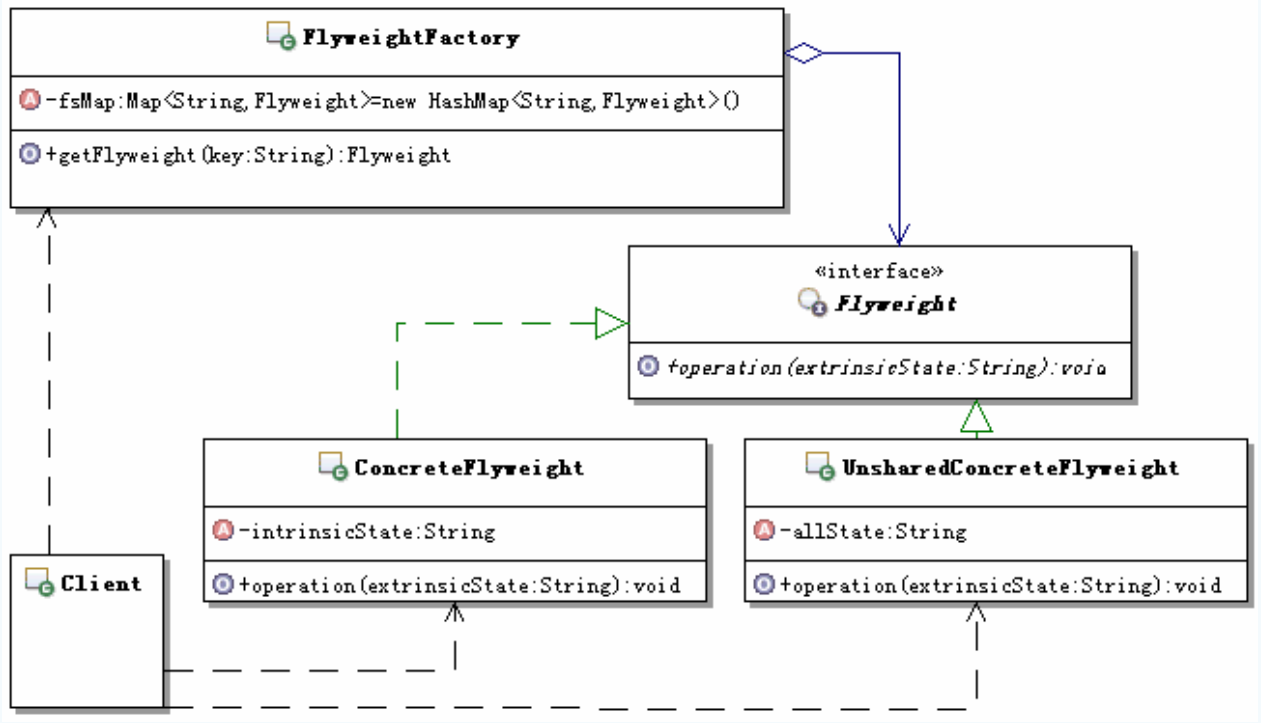
- 作业要求
  1：在老师给定的概要代码基础上，实现上述基本要求的功能设计
  2：重点在接口和功能实现的设计上，无须关注具体实现
  3：对于每个api，可以适当写点样例代码，能够调用运行更佳
  4：考虑合理的结构，职责的划分，以及设计模式的合理使用

初识享元模式

- 定义

  运用共享技术有效地支持大量细粒度的对象 。

- 结构和说明



享元模式的知识要点

- 享元模式的知识要点

1：享元模式设计的重点就在于分离变与不变，把一个对象的状态分成内部状态和外部状态，内部状态是不变的，外部状态是可变的。然后通过共享不变的部分，达到减少对象数量、并节约内存的目的

2：在享元模式中，享元对象又有共享与不共享之分，这种情况通常出现在跟组合模式合用的情况，通常共享的是叶子对象，一般不共享的部分是由共享部分组合而成的

3：享元的内部状态和外部状态是独立的，外部状态的变化不应该影响到内部状态，在需要的时候，可以把外部状态传递给享元对象使用

4：在享元模式中，通常是在第一次向享元工厂请求获取共享对象的时候，进行共享对象的初始化，而且多半都是在享元工厂内部实现，不会从外部传入共享对象

5：在实现享元模式的时候，通常会考虑垃圾清除的问题

思考享元模式

- 享元模式的本质

  享元模式的本质是：**分离与共享**

- 何时选用享元模式

1：如果一个应用程序使用了大量的细粒度对象，可以使用享元模式来减少对象数量

2：如果由于使用大量的对象，造成很大的存储开销，可以使用享元模式来减少对象数量，并节约内存

3：如果对象的大多数状态都可以转变为外部状态，比如通过计算得到，或是从外部传入等，可以使用享元模式来实现内部状态和外部状态的分离

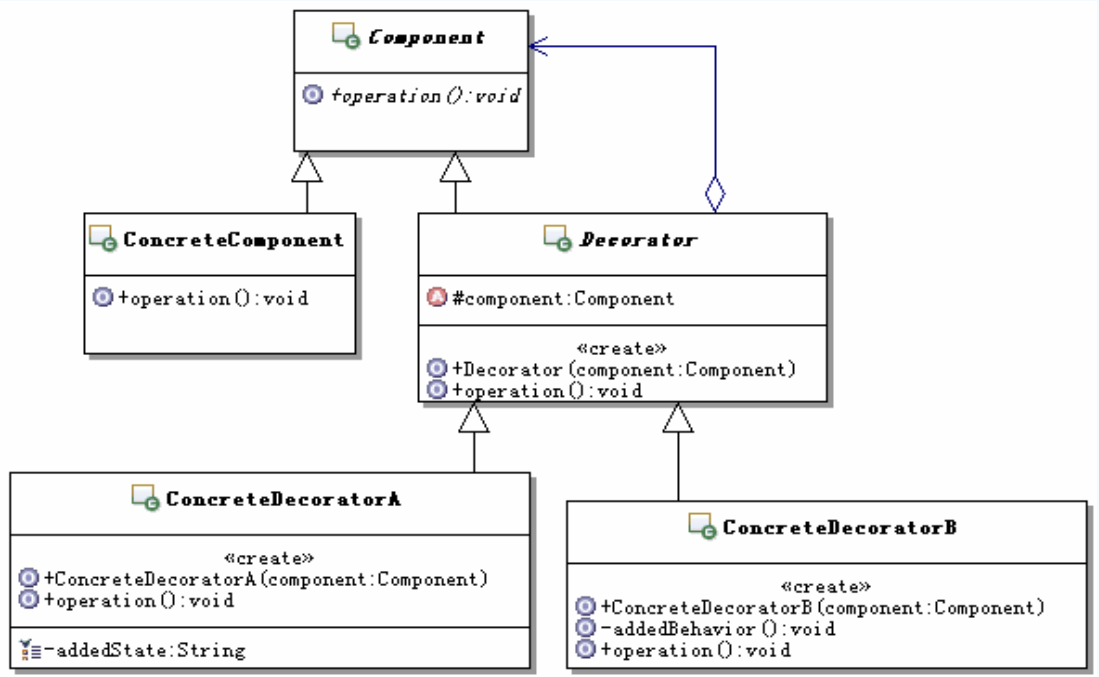
4：如果不考虑对象的外部状态，可以用相对较少的共享对象取代很多组合对象，可以使用享元模式来共享对象，然后组合对象来使用这些共享对象

初识装饰模式

- 定义

  动态地给一个对象添加一些额外的职责。就增加功能来说，装饰模式比生成子类更为灵活。

- 结构和说明



装饰者模式的知识要点

- 装饰者模式的知识要点

1：装饰模式能够实现动态的为对象添加功能，是从一个对象外部来给对象增加功能，相当于是改变了对象的外观，增强了系统的灵活性，而且也使得装饰器功能得到复用

2：装饰模式的思考起点是“尽量使用对象组合，而不是对象继承”来扩展和复用功能

3：装饰器是能够包含其它的装饰器对象的，相当于组合结构中的树枝对象，只不过装饰器对象只能包含一个装饰器对象，而不是多个。因此装饰模式的结构图根组合模式的结构图是相似的

4：装饰器对象的调用，也是要使用递归调用的

5：在装饰器里不仅仅是可以给被装饰对象增加功能，还可以根据需要选择是否调用被装饰对象的功能，如果不调用被装饰对象的功能，那就变成完全重新实现了，相当于动态修改了被装饰对象的功能

装饰者模式的知识要点

6：装饰器是用来装饰组件的，装饰器一定要实现和组件类一致的接口，保证它们是同一个类型，并具有同一个外观，这样组合完成的装饰才能够递归的调用下去

7：组件类是不知道装饰器的存在的，装饰器给组件添加功能是一种透明的包装，组件类毫不知情

8：装饰模式和AOP要实现的功能是类似的，只不过AOP的实现方法不同，会更加灵活，更加可配置；另外AOP**一个更重要的变化是思想上的变化——“主从换位”**，让原本主动调用的功能模块变成了被动等待，甚至毫不知情的情况下被织入了很多新的功能

思考装饰模式

- 装饰模式的本质

  装饰模式的本质是：**动态组合**

- 何时选用装饰模式

1：如果需要在不影响其它对象的情况下，以动态、透明的方式给对象添加职责，可以使用装饰模式，这几乎就是装饰模式的主要功能

2：如果不合适使用子类来进行扩展的时候，可以考虑使用装饰模式，因为装饰模式是使用的“对象组合”的方式。所谓不适合用子类扩展的方式，比如：扩展功能需要的子类太多，造成子类数目呈爆炸性增长。