

Java私塾-最专业的Java就业培训专家，因为专业，所以出色！值得你的信赖！



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

10101010101010101010101010101

本周设计作业

n 本周设计作业的项目背景：

仓储堆位管理——来自中国邮政物流综合管理系统

n 学习目标：

(1) 在实战中练习设计的理念和方法

(2) 学习如何灵活的预留功能扩展

(3) 综合应用组合模式、访问者模式，包含着这些模式但不限于这些模式

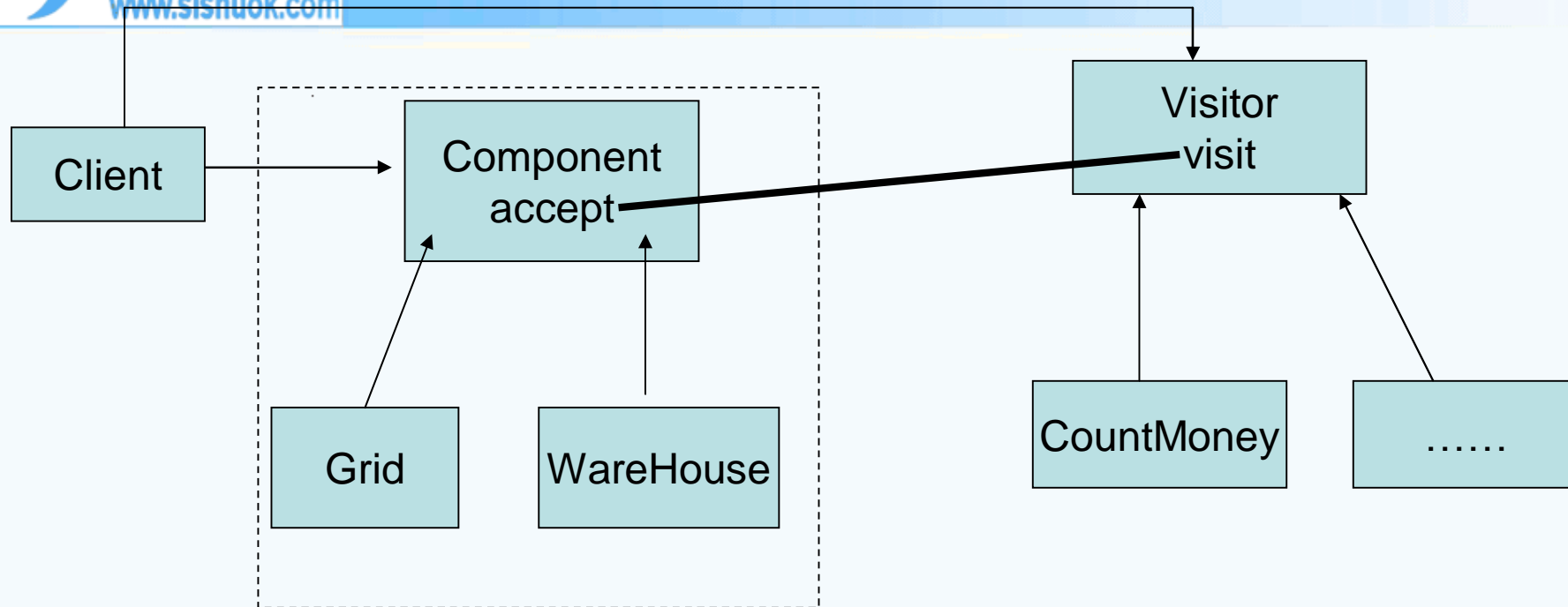
该设计方式可应用于多种有类似功能的系统，比如：各种类型树、类别树、组织结构树.....

n 基本功能

1：分成仓库、库区、堆位（实际业务分成：中心仓和区域仓库，每个仓库又分成库区、楼层、区域、储排、储架、层位、格位等）

2：要求能够统计每个仓库、每个库区下堆位的使用情况

3：要考虑到今后的扩展，比如：分析整个仓库的堆位的使用周转率、平均存放时间等，又比如统计某一个库区下的堆位存放货物的总金额等等.....



- 1: 采用编码01 0101 010101
- 2: 采用 引用 01 aaa -1(puuid)
- 02 bbb 01
- 3: 存储成不同的表
- 4: 存储在一个表，通过类型等进行区分
- 5: 层次划分不确定性

本周设计作业

n 作业要求

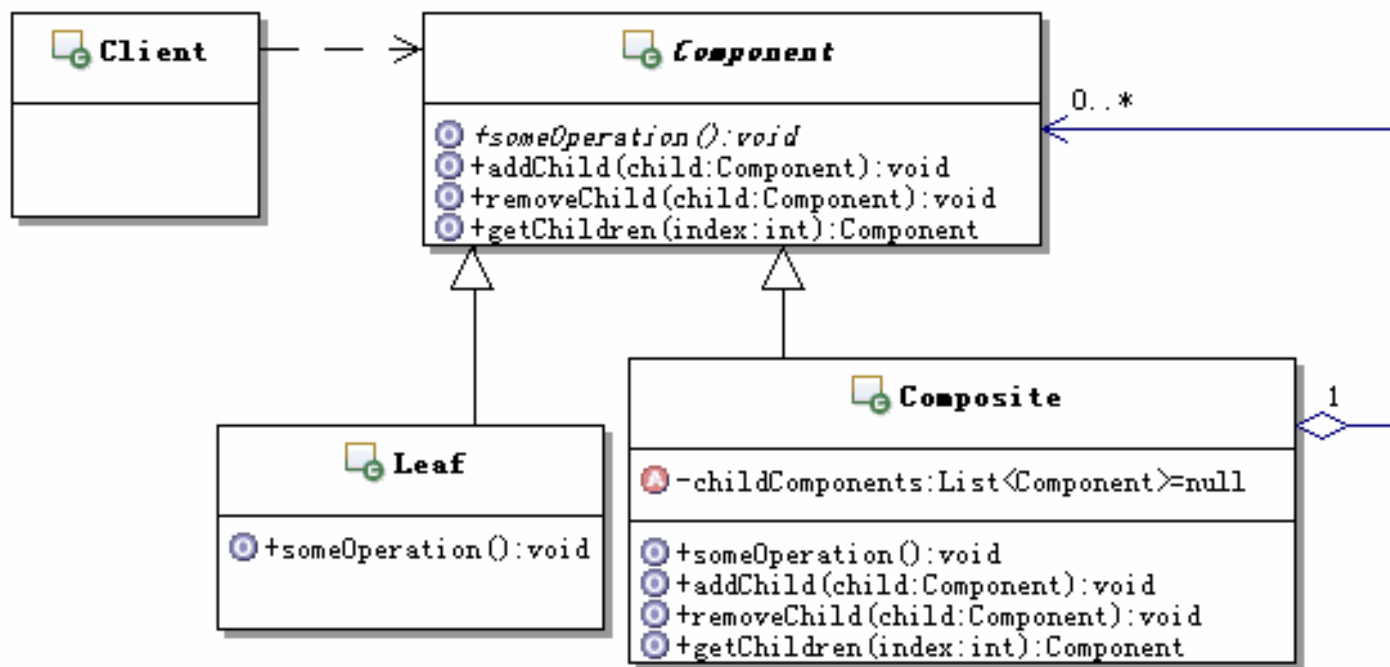
- 1: 在老师给定的概要代码基础上，实现上述基本要求的功能设计
- 2: 重点在功能实现的设计上，无须关注具体实现
- 3: 对于每个api，可以适当写点样例代码，能够调用运行更佳
- 4: 考虑合理的结构，职责的划分，以及设计模式的合理使用

初识组合模式

n 定义

将对象组合成树形结构以表示“部分-整体”的层次结构。组合模式使得用户对单个对象和组合对象的使用具有一致性。

n 结构和说明



组合模式的知识要点

n 组合模式的知识要点

- 1: 组合模式的目的是：让客户端不再区分操作的是组合对象还是叶子对象，而是以一个统一的方式来操作。
- 2: 组合模式设计的关键之处，是设计一个抽象的组件类，让它可以代表组合对象和叶子对象。
- 3: 组合模式通常会构成对象树，理论上可以包含无数个层次
- 4: 通常需要以对象递归的方式来访问对象树种的对象
- 5: 使用组合模式的时候，应该更关注于透明性而非安全性
- 6: 必要的时候，要考虑父子组件的引用关系，并避免环状引用，当然有意这样设计的除外

思考组合模式

n 组合模式的本质

组合模式的本质是：统一叶子对象和组合对象

n 何时选用组合模式

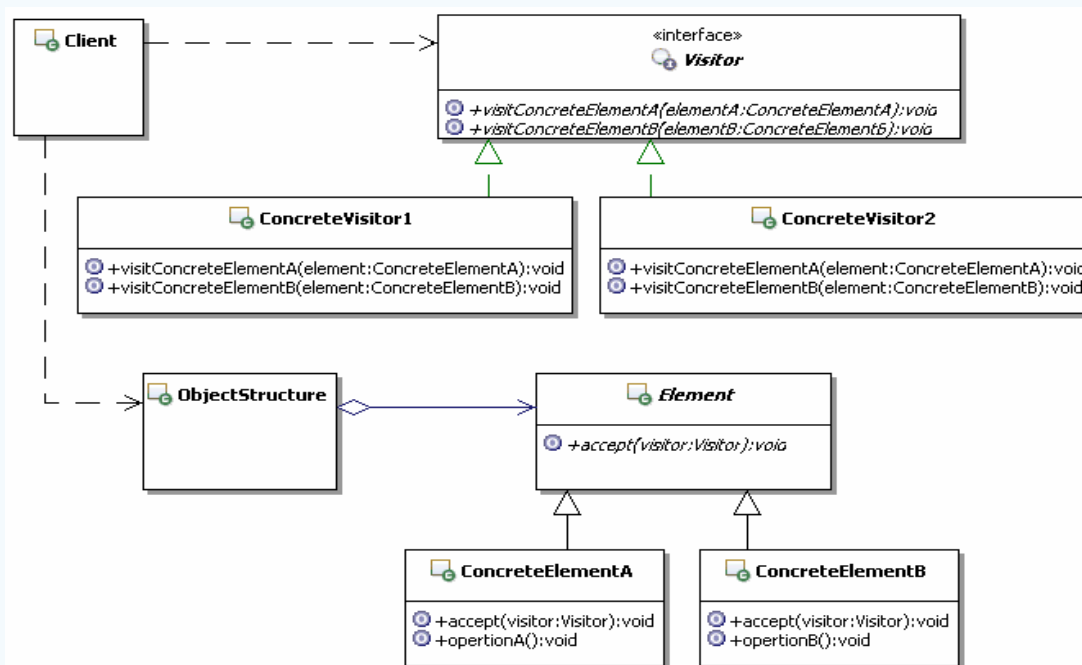
- 1: 如果你想表示对象的部分-整体层次结构，可以选用组合模式，把整体和部分的~~操作~~统一起来，使得层次结构实现更简单，从外部来使用这个层次结构也简单。
- 2: 如果你希望统一的使用组合结构中的所有对象，可以选用组合模式，这正是组合模式提供的主要功能

初识访问者模式

n 定义

表示一个作用于某对象结构中的各元素的操作。它使你可以在不改变各元素的类的前提下定义作用于这些元素的新操作。

n 结构和说明



访问者模式的知识要点

- 1: 访问者模式能给一系列对象，透明的添加新功能。从而避免在维护期间，对这一系列对象进行修改，而且还能变相实现复用访问者所具有的功能。
- 2: 访问者模式在实现的时候，通常都需要循环或是递归对象结构里面的所有对象，要保证每个对象都要被访问到
- 3: 访问者模式的accept方法和visit方法，构成一个调用的通路，从而使得新的功能能够通过这个通路添加到对象结构上
- 4: 访问者模式实现的核心技术就是两次分发的技术，当客户端调用ObjectStructure的时候，会遍历ObjectStructure中所有的元素，调用这些元素的accept方法，让这些元素来接受访问，这是请求的第一次分发；在具体的元素对象中实现accept方法的时候，会回调访问者的visit方法，等于请求被第二次分发了，请求被分发给访问者来进行处理，真正实现功能的正是访问者的visit方法
- 5: 通常不在父类中实现回调visit方法，因为不同的子类，visit方法返回的实例是不一样的
- 6: 在实现访问者模式的时候，要考虑让谁来负责遍历所有的元素对象，如果是集合结构，通常就在Obstructure里面循环访问；如果是组合模式构成的对象树，通常就在组合对象的accept方法里面，递归遍历它的子元素
- 7: 如果只有一个被访问对象，就不要循环和迭代了

思考访问者模式

n 访问者模式的本质

访问者模式的本质是：预留通路，回调实现

n 何时选用访问者模式

- 1: 如果想对一个对象结构，实施一些依赖于对象结构中的具体类的操作，可以使用访问者模式
- 2: 如果想对一个对象结构中的各个元素，进行很多不同的而且不相关的操作，为了避免这些操作使得类变得杂乱，可以使用访问者模式，把这些操作分散到不同的访问者对象中去，每个访问者对象实现同一类功能。
- 3: 如果对象结构很少变动，但是需要经常给对象结构中的元素对象定义新的操作，可以使用访问者模式