visitor模式：抽象出**功能（一般和组合模式结合使用）**，将某个服务（如对一个节点的操作）包含的功能抽象成一个个的 visitor，方便调用方对功能的替换

场景

（1）组合模式，我们现在想象一下，我们之前做的是说递归删除部门自己本身

（2）但是我们现在要对他改造一下，将组合模式和访问者模式结合起来，就是说，我们可以让那颗树执行我们想要他做的任何事情

 【一个动作接口】

public **interface Visitor** {

   void visit(Element element);

}

【具体的操作动作】

public class **ConcreteVisitor implements Visitor** {

   public void visit(Element element) {

  }

}

public **abstract class Element** {

   public abstract void accept(Visitor visitor);

}

【根据不同的动作，执行不同的功能】

public class **ConcreteElement extends Element** {

   public void accept(Visitor visitor) {

       visitor.visit(this);

  }

}

访问者模式，是对目标对象，动态的执行某个功能，而不对目标对象有任何的改动

长什么样？如果不这么做有什么不好？这么做了有什么好处？ -> 大白话，没有任何理论的东西在里面，就是纯大白话，简单的代码，把模式的精髓给大家展现出来

如果真正要透彻的掌握，深刻的理解，自己以后可以用到自己的业务系统里去， 设计出来那种可扩展、可维护、结构清晰、优雅的代码，还需要实践，电商系统里重点去实践一下

软件工程、项目管理、设计模式，是架构师的基本功，spring boot比较流行，大路货，spring boot源码

制定系统的开发流程和规范，制定项目管理流程和规范，按照那套流程和规范去实际把控和完成整个项目，设计模式都是必用，代码质量

23种设计模式里，只有一种我这里没讲，解释器模式，真的太冷门了，几乎很少用

实际开始做项目了，结合我们的电商系统的需求文档 -> 需求评审 -> 需求分析 -> 概要设计 -> 详细设计 -> 工程初始化 -> 项目管理 -> 开发 -> 单元测试 -> 冒烟测试 -> 静态代码扫描 -> code review -> 集成测试 -> 系统测试 -> 验收测试 -> 系统上线 -> 线上运维

重点在业务里实践软件工程和项目管理，设计模式 -> spring boot源码分析 -> 交付出来真实的电商系统v1.0内测版

853769620