访问者模式的目的是封装一些施加于某些数据结构元素之上的操作，即使这些操作需要修改，接收这个操作的数据结构却可以保持不变。

一.访问者模式的现实场景

现实场景中两个访问者，药剂师和收银员。我们找医生生成病历单，就相当与一种封装，不管谁来看病这个封装过程都是类似的，都生成一个数据结构类似的病历单。

药剂师能根据这个封装的结果来获取到里面的信息。同样的，收银员也是，通过不同病人的病历单上的数据解析封装得到不同的账单，收银员通过访问账单就可以得到一些信息。

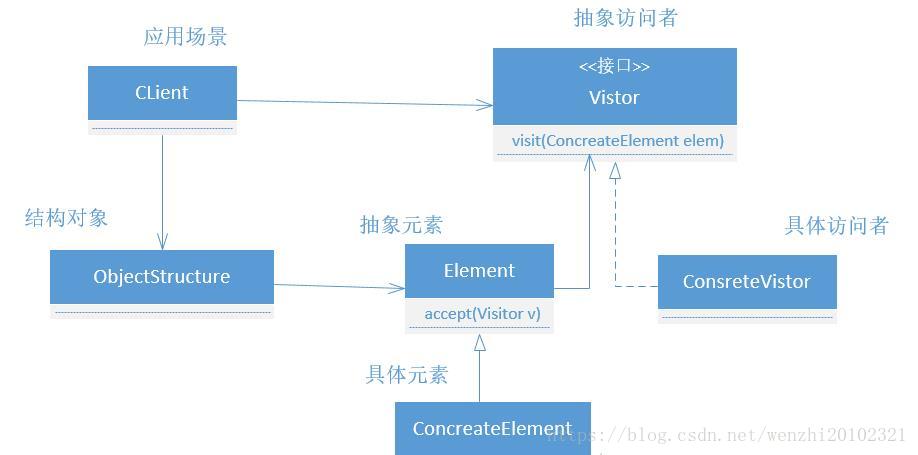
上面不同病人的病情不同，生成的病历单数据不同，但是药剂师都能读懂这个病历单。

不同病人的账单不同，收银员也是能够计算出具体收费情况。

二.访问者模式（Visitor Pattern）的定义

封装一些作用于某种数据结构的各种元素的操作，它可以在不改变数据结构的前提下定义作用于这些元素的新操作。

1. 访问者模式的类图



四.访问者模式的五个角色

1.抽象访问者（Visitor）角色

该角色声明一个或多个访问操作，定义访问者可以访问哪些元素。

2.具体访问者（Concrete Visitor）角色

该角色实现抽象访问者角色中的各个访问者操作。

3.抽象元素（Element）角色

该角色声明一个接收操作，接收一个访问者的对象。

4.具体元素（Concrete Element）角色

该角色实现抽象元素角色中的接收操作

5.结构元素（Object Structure）角色

该角色有以下责任： 可以遍历结构中所有的元素。

如果需要，该角色提供一个高层次的接口让访问者对象可以访问每一个元素，也可以设计一个复合对象或者一个集合，如List或Set。

五.访问者模式的优缺点

访问者模式中的优点：

1.访问者模式使得增加新的访问操作更容易。

2.访问者模式将有关元素对象的访问行为集中到一个访问者对象中，而不是分散到一个个的元素类中。

3.访问者模式可以跨过几个类的等级机构访问属于不同的等级结构的成员类。

4.累积状态。每一个单独的访问者对象都集中了相关的行为，从而分散到很多的元素对象中，易于系统的维护。

访问者模式的缺点：

1.增加新的元素类变得困难。每增加一个形的元素类就意味着要在抽象访问者角色中增加一个新的抽象操作，并在每一个具体访问者类中增加相应的具体操作。

2.破坏封装。访问者模式要求访问者对象访问每一个元素对象并调用每一个元素对象的操作，这隐含了一个对所有元素对象的要求，即访问者对象必须暴露一些自己的操作和内部状态，否则访问者的访问就没有意义。

3.违背了依赖倒置原则。访问者模式依赖的是具体的元素，而不是抽象的元素，这破坏了依赖倒置的原则，特别是在面向对象的编程中，访问者对象抛弃了对接口的依赖，而直接依赖实现类，扩展比较难。

六.访问者模式的使用场景

1、一个对象结构包含很多类型的对象，希望对这些对象实施一些依赖其具体类型的操作。在访问者中针对每一种具体的类型都提供了一个访问操作，不同类型的对象可以有不同的访问操作。

2、需要对一个对象结构中的对象进行很多不同的并且不相关的操作，而需要避免让这些操作“污染”这些对象的类，也不希望在增加新操作时修改这些类。访问者模式使得我们可以将相关的访问操作集中起来定义在访问者类中，对象结构可以被多个不同的访问者类所使用，将对象本身与对象的访问操作分离。

3、对象结构中对象对应的类很少改变，但经常需要在此对象结构上定义新的操作。