

JAVA基础：深入浅出Java的访问者模式 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/145/2021\\_2022\\_JAVA\\_E5\\_9F\\_BA\\_E7\\_A1\\_80\\_c104\\_145207.htm](https://www.100test.com/kao_ti2020/145/2021_2022_JAVA_E5_9F_BA_E7_A1_80_c104_145207.htm) 一、引子 对于系统中一个已经完成的类层次结构，我们已经给它提供了满足需求的接口。

但是面对新增加的需求，我们应该怎么做呢？如果这是为数不多的几次变动，而且你不用为了一个需求的调整而将整个类层次结构统统地修改一遍，那么直接在原有类层次结构上修改也许是个不错的主意。但是往往我们遇到的却是：这样的需求变动也许会不停的发生；更重要的是需求的任何变动可能都要让你将整个类层次结构修改个底朝天……。这种类似的操作分布在不同的类里面，不是一个好现象，我们要对这个结构重构一下了。那么，访问者模式也许是你很好的选择。

二、定义与结构 访问者模式，顾名思义使用了这个模式后就可以在不修改已有程序结构的前提下，通过添加额外的“访问者”来完成对已有代码功能的提升。《设计模式》一书对于访问者模式给出的定义为：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。从定义可以看出结构对象是使用访问者模式必须条件，而且这个结构对象必须存在遍历自身各个对象的方法。这便类似于java中的collection概念了。以下是访问者模式的组成结构：1) 访问者角色

( Visitor )：为该对象结构中具体元素角色声明一个访问操作接口。该操作接口的名字和参数标识了发送访问请求给具体访问者的具体元素角色。这样访问者就可以通过该元素角色的特定接口直接访问它。2) 具体访问者角色 ( Concrete

Visitor ) : 实现每个由访问者角色 ( Visitor ) 声明的操作。 3) 元素角色 ( Element ) : 定义一个Accept操作 , 它以一个访问者为参数。 4) 具体元素角色 ( Concrete Element ) : 实现由元素角色提供的Accept操作。 5) 对象结构角色 ( Object Structure ) : 这是使用访问者模式必备的角色。它要具备以下特征 : 能枚举它的元素 ; 可以提供一個高层的接口以允许该访问者访问它的元素 ; 可以是一个复合 ( 组合模式 ) 或是一个集合 , 如一个列表或一个无序集合。 来张类图就能更加清晰的看清访问者模式的结构了。 那么像引言中假想的。 我们应该做些什么才能让访问者模式跑起来呢 ? 首先我们要在原有的类层次结构中添加accept方法。 然后将这个类层次中的类放到一个对象结构中去。 这样再去创建访问者角色..... 三、 举例 本人阅历实在可怜 , 没能找到访问者模式在实际应用中的例子。 只好借《Thinking in Patterns with java》中的教学代码一用。 我稍微做了下修改。 import java.util.\*. import junit.framework.\*. //访问者角色 interface Visitor { void visit(Gladiolus g). void visit(Runuculus r). void visit(Chrysanthemum c). } // The Flower hierarchy cannot be changed: //元素角色 interface Flower { void accept(Visitor v). } 100Test 下载频道开通 , 各类考试题目直接下载。 详细请访问 [www.100test.com](http://www.100test.com)