

Java设计模式之Visitor模式Java认证考试 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/644/2021_2022_Java_E8_AE_BE_E8_AE_A1_c104_644812.htm Visitor 定义作用于某个对象群中各个对象的操作. 它可以使你在不改变这些对象本身的情况下,定义作用于这些对象的新操作. 在Java中,Visitor 模式实际上是分离了collection 结构中的元素和对这些元素进行操作的行为. 为何使用Visitor? Java 的Collection(包括Vector 和Hashtable)是我们最经常使用的技术,可是Collection好象是个黑色大染缸,本来有各种鲜明类型特征的对象一旦放入后,再取出时,这些类型就消失了.那么我们势必要用If 来判断,如:

```
Iterator iterator = collection.iterator() while (iterator.hasNext()) {  
Object o = iterator.next(). if (o instanceof Collection)  
messyPrintCollection((Collection)o). else if (o instanceof String)  
System.out.println(" " o.toString() "). else if (o instanceof Float)  
System.out.println(o.toString() "f"). else  
System.out.println(o.toString()). }
```

在上例中,我们使用了 instanceof 来判断 o 的类型. 很显然,这样做的缺点代码If else if 很繁琐.我们就可以使用Visitor 模式解决它. 如何使用Visitor?

针对上例,我们设计一个接口visitor 访问者: public interface Visitor { public void visitCollection(Collection collection). public void visitString(String string). public void visitFloat(Float float). } 在这个接口中,将我们认为Collection 有可能的类的类型放入其中. 有了访问者,我们需要被访问者,被访问者就是我们Collection 的每个元素Element,我们要为这些Element 定义一个可以接受访问的接口(访问和被访问是互动的,只有访问者,

被访问者如果表示不欢迎,访问者就不能访问),我们定义这个接口叫Visitable,用来定义一个Accept操作,也就是说让Collection 每个元素具备可访问性. `public interface Visitable {
public void accept(Visitor visitor). }`好了,有了两个接口,我们就要定义他们的具体实现(Concrete class): 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