

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 《研磨设计模式》 ——跟着CC学设计系列精品教程

10101010101010101010101010101

私塾在线<http://si.shuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送

本节课程概览

n 学习访问者模式

一：初识访问者模式

包括：定义、结构、参考实现

二：体会访问者模式

包括：场景问题、不用模式的解决方案、使用模式的解决方案

三：理解访问者模式

包括：认识访问者模式、操作组合对象结构、谁负责遍历所有元素对象、访问者模式的优缺点

四：思考访问者模式

包括：访问者模式的本质、何时选用

做最好的在线学习社区

网 址：<http://sishuok.com>

咨询QQ：2371651507

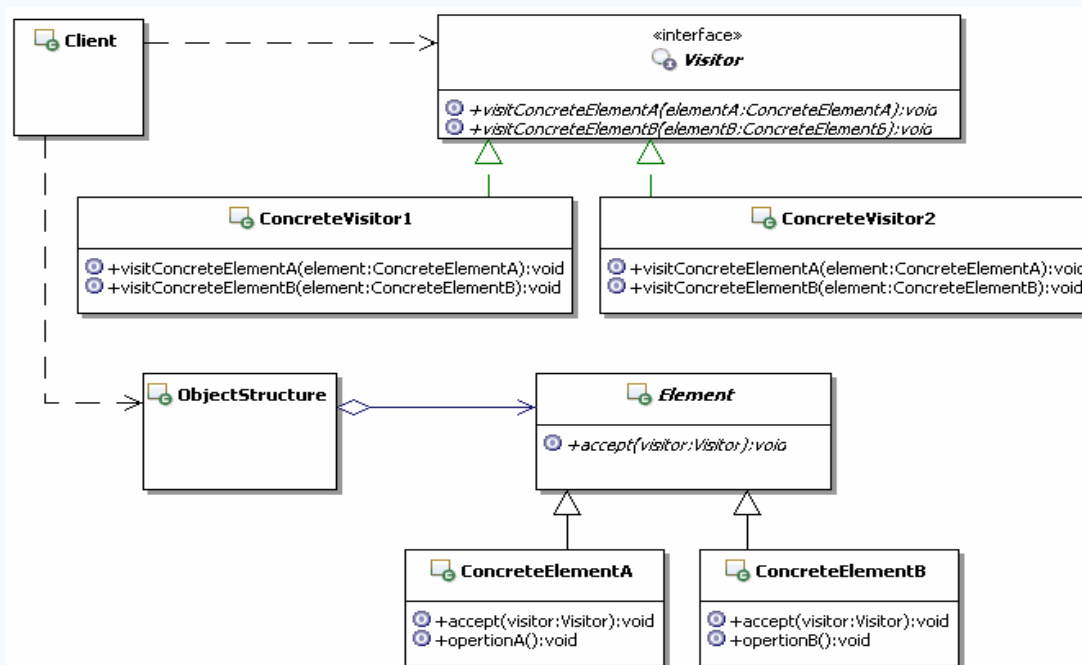
私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送

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



做最好的在线学习社区

网 址: <http://sishuok.com>

咨询QQ: 2371651507

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送

初识访问者模式

Visitor：访问者接口，为所有的访问者对象声明一个visit方法，用来代表为对象结构添加的功能，理论上可以代表任意的功能。

ConcreteVisitor：具体访问者实现对象，实现要真正被添加到对象结构中的功能。

Element：抽象的元素对象，对象结构的顶层接口，定义接受访问的操作。

ConcreteElement：具体元素对象，对象结构中具体的对象，也是被访问的对象，通常会回调访问者的真实功能，同时开放自身的数据供访问者使用。

ObjectStructure：对象结构，通常包含多个被访问的对象，它可以遍历这多个被访问的对象，也可以让访问者访问它的元素。可以是一个复合或是一个集合，如一个列表或无序集合。

但是请注意：这个ObjectStructure并不是我们在前面讲到的对象结构，前面一直讲的对象结构是指的一系列对象的定义结构，是概念上的东西；而ObjectStructure可以看成是对象结构中的一系列对象的一个集合，是用来辅助客户端访问这一系列对象的，所以为了不造成大家的困惑，后面提到ObjectStructure的时候，就用英文名称来代替，不把它翻译成中文。

做最好的在线学习社区

网 址：<http://sishuok.com>

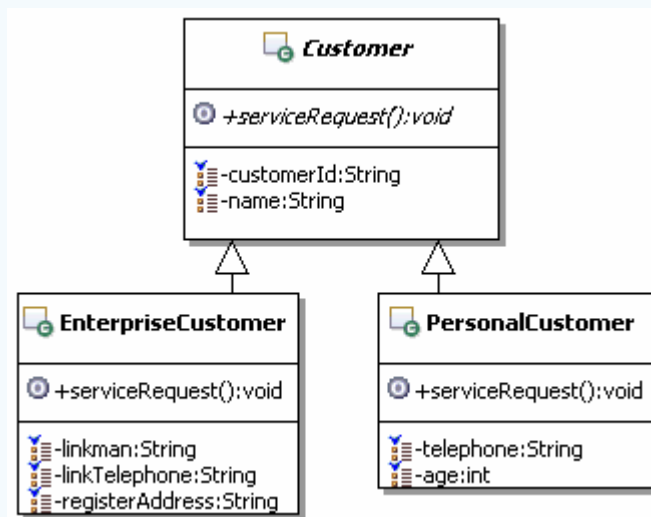
咨询QQ：2371651507

体会访问者模式

n 扩展客户管理的功能

考虑这样一个应用：扩展客户管理的功能。

既然是扩展功能，那么肯定是已经存在一定的功能了，先看看已有的功能：公司的客户分成两大类，一类是企业客户，一类是个人客户，现有的功能非常简单，就是能让客户提出服务申请。目前的程序结构如图



做最好的在线学习社区

网 址：<http://sishuok.com>

咨询QQ：2371651507

体会访问者模式

随着业务的发展，需要加强客户管理的功能，假设现需增加如下的功能：

- 1: 客户对公司产品的偏好分析，针对企业客户和个人客户有不同的分析策略，主要是根据以往购买的历史、潜在购买意向等进行分析，对于企业客户还要添加上客户所在行业的发展趋势、客户的发展预期等的分析。
- 2: 客户价值分析，针对企业客户和个人客户，有不同的分析方式和策略。主要是根据购买的金额大小、购买的产品和服务的多少、购买的频率等进行分析。

其实除了这些功能，还有很多潜在的功能，只是现在还没有要求实现，比如：针对不同的客户进行需求调查；针对不同的客户进行满意度分析；客户消费预期分析等等。虽然现在没有要求实现，但不排除今后有可能会要求实现。

体会访问者模式

n 不用模式的解决方案

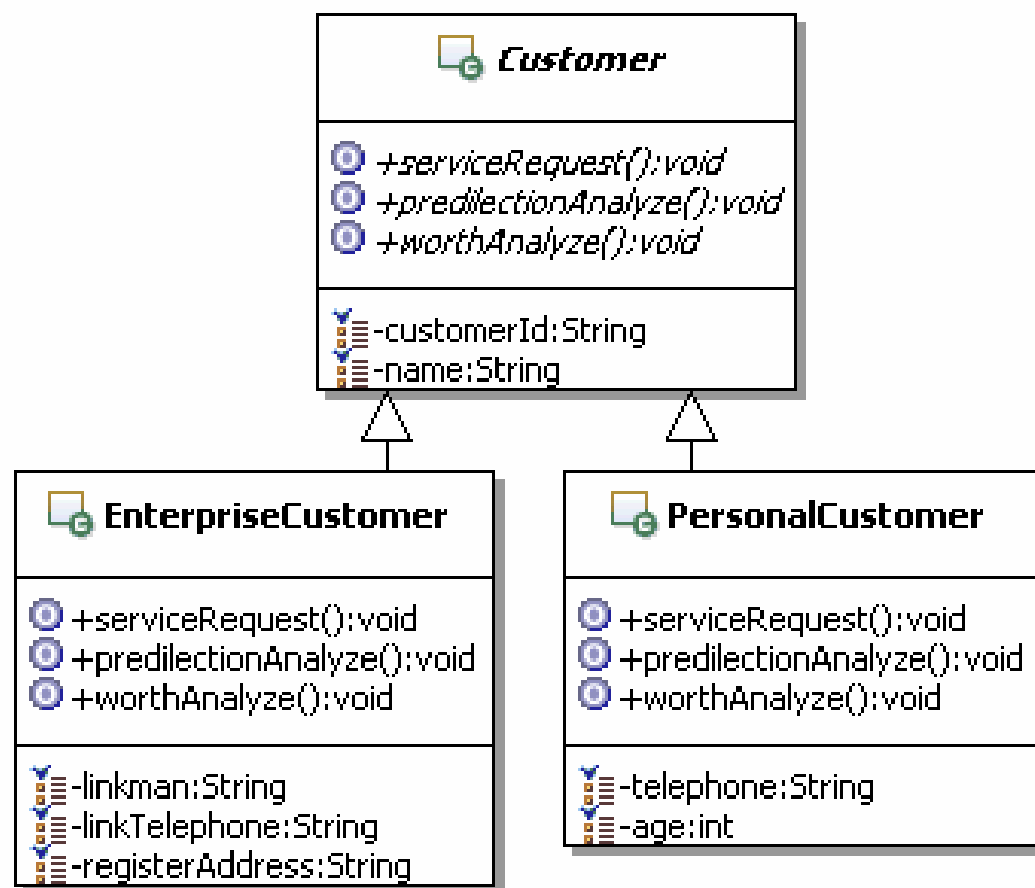
1: 实现思路

要实现上面要求的功能，也不是很困难，一个很基本的想法就是：既然不同类型的客户操作是不同的，那么在不同类型的客户里面分别实现这些功能，不就可以了。

由于这些功能的实现依附于很多其它功能的实现，或者是需要很多其它的业务数据，在示例里面不太好完整的体现其功能实现，都是示意一下，因此提前说明一下。

按照上述的想法，这个时候的程序结构如图

体会访问者模式



做最好的在线学习社区

网 址: <http://sishuok.com>

咨询QQ: 2371651507

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送

体会访问者模式

n 有何问题

以很简单的方式，实现了要求的功能，这种实现有没有什么问题呢？仔细分析上面的实现，发现有两个主要的问题：

- 1: 在企业客户和个人客户的类里面，都分别实现了提出服务请求、进行产品偏好分析、进行客户价值分析等功能，也就是说，这些功能的实现代码是混杂在同一个类里面的；而且相同的功能分散到了不同的类中去实现，这会导致整个系统难以理解、难以维护。
- 2: 更为痛苦的是，采用这样的实现方式，如果要给客户扩展新的功能，比如前面提到的针对不同的客户进行需求调查；针对不同的客户进行满意度分析；客户消费预期分析等等。每次扩展，都需要改动企业客户的类和个人客户的类，当然也可以通过为它们扩展子类的方式，但是这样可能会造成过多的对象层次。

那么有没有办法，能够在不改变客户这个对象结构中各元素类的前提下，为这些类定义新的功能？也就是要求不改变企业客户和个人客户类，就能为企业客户和个人客户类定义新的功能？

做最好的在线学习社区

网 址：<http://sishuok.com>

咨询QQ：2371651507

体会访问者模式

n 使用模式来解决的思路

仔细分析上面的示例，对于客户这个对象结构，不想改变类，又要添加新的功能，很明显就需要一种动态的方式，在运行期间把功能动态地添加到对象结构中去。

有些朋友可能会想起装饰模式，装饰模式可以实现为一个对象透明的添加功能，但装饰模式基本上是在现有的功能的基础之上进行功能添加，实际上是对现有功能的加强或者改造。并不是在现有功能不改动的前提下，为对象添加新的功能。

看来需要另外寻找新的解决方案了，可以应用访问者模式来解决这个问题。要使用访问者模式来重写示例，首先就要按照访问者模式的结构，分离出两个类层次来，一个是对应于元素的类层次，一个是对应于访问者的类层次。

对于对应于元素的类层次，现在已经有了，就是客户的对象层次。而对应于访问者的类层次，现在还没有，不过，按照访问者模式的结构，应该是先定义一个访问者接口，然后把每种业务实现成为一个单独的访问者对象，也就是说应该使用一个访问者对象来实现对客户的偏好分析，而用另外一个访问者对象来实现对客户价值分析。

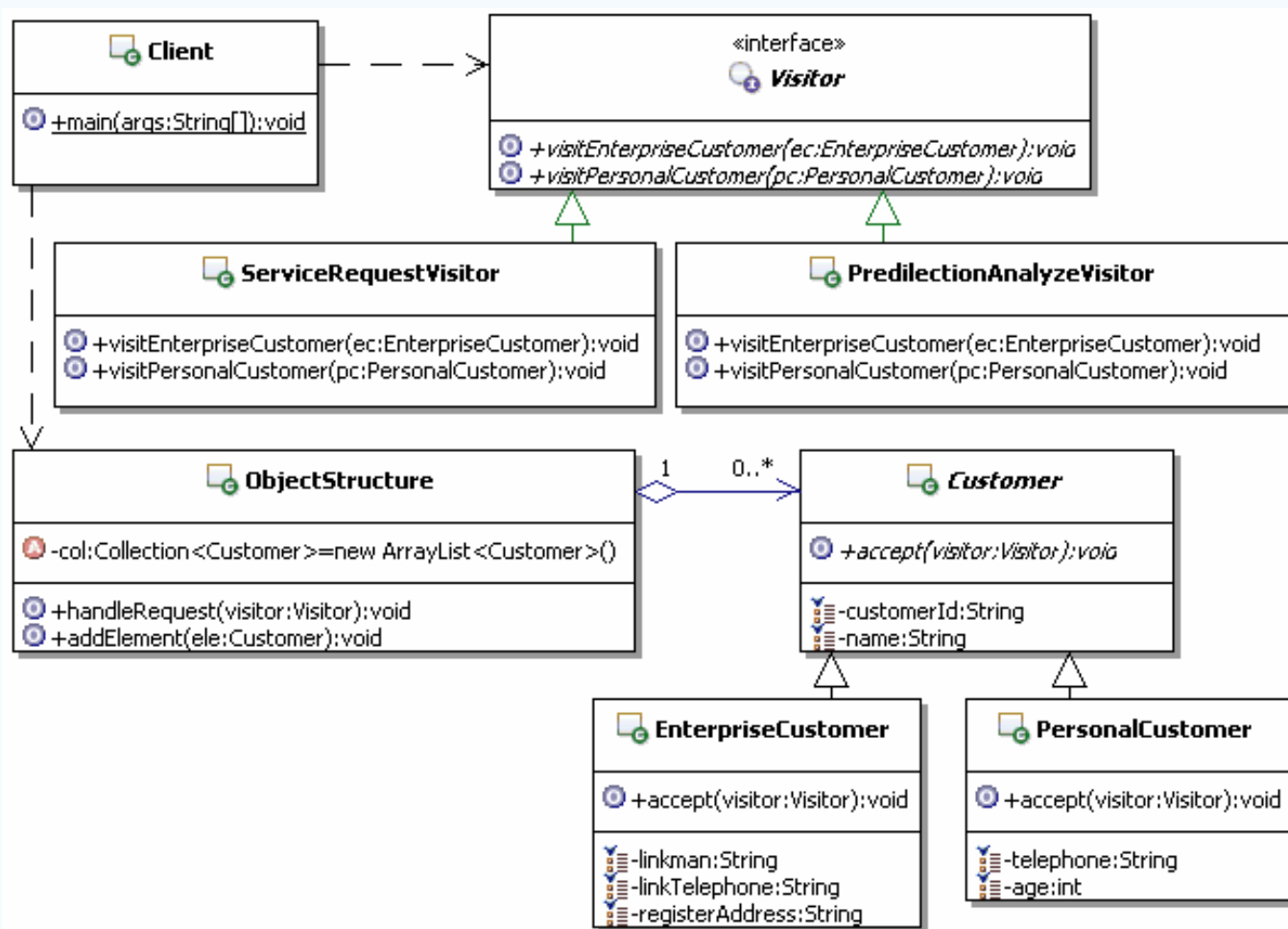
在分离好两个类层次过后，为了方便客户端的访问，定义一个ObjectStructure，其实就类似于前面示例中的客户管理的业务对象。新的示例的结构如图：

做最好的在线学习社区

网 址：<http://sishuok.com>

咨询QQ：2371651507

体会访问者模式



做最好的在线学习社区

网 址: <http://sishuok.com>

咨询QQ: 2371651507

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送

理解访问者模式

n 认识访问者模式

1: 访问者的功能

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。

由于是针对一系列对象的操作，这也导致，如果只想给一系列对象中的部分对象添加功能，就会有些麻烦；而且要始终能保证把这一系列对象都要调用到，不管是循环也好，还是递归也好，总之要让每个对象都要被访问到。

2: 调用通路

访问者之所以能实现“为一系列对象透明的添加新功能”，注意是透明的，也就是这一系列对象是不知道被添加功能的。

重要的就是依靠通用方法，访问者这边说要去访问，就提供一个访问的方法，如 `visit` 方法；而对象那边说，好的，我接受你的访问，提供一个接受访问的方法，如 `accept` 方法。这两个方法并不代表任何具体的功能，只是构成一个调用的通路，那么真正的功能实现在哪里呢？又如何调用到呢？

很简单，就在 `accept` 方法里面，回调 `visit` 的方法，从而回调到访问者的具体实现上，而这个访问者的具体实现的方法才是要添加的新的功能。

做最好的在线学习社区

网 址: <http://sishuok.com>

咨询QQ: 2371651507

理解访问者模式

3: 两次分发技术

访问者模式能够实现在不改变对象结构的情况下，就能给对象结构中的类增加功能，实现这个效果所使用的核心技术就是两次分发的技术。

在访问者模式中，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。

两次分发技术具体的调用过程示意如图：

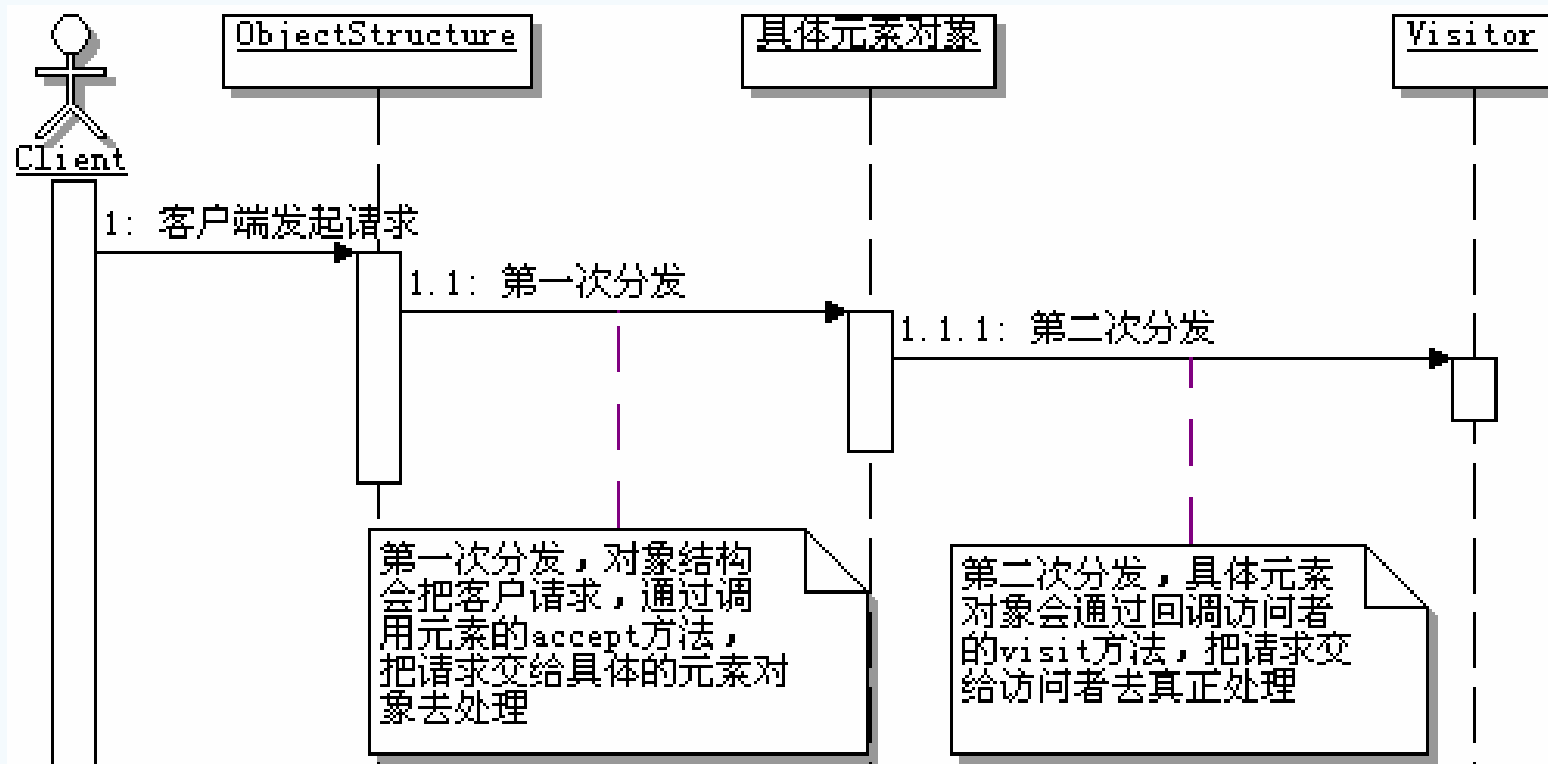
做最好的在线学习社区

网 址：<http://sishuok.com>

咨询QQ：2371651507

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送

理解访问者模式



做最好的在线学习社区

网 址：<http://sishuok.com>

咨询QQ：2371651507

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送

理解访问者模式

两次分发技术使得客户端的请求不再被静态的绑定在元素对象上，这个时候真正执行什么样的功能同时取决于访问者类型和元素类型，就算是同一种元素类型，只要访问者类型不一样，最终执行的功能也不会一样，这样一来，就可以在元素对象不变的情况下，通过改变访问者的类型，来改变真正执行的功能。

两次分发技术还有一个优点，就是可以在程序运行期间进行动态的功能组装和切换，只需要在客户端调用时，组合使用不同的访问者对象实例即可。

从另一个层面思考，Java回调技术也有点类似于两次分发技术，客户端调用某方法，这个方法就类似于accept方法，传入一个接口的实现对象，这个接口的实现对象就有点像是访问者，在方法内部，会回调这个接口的方法，就类似于调用访问者的visit方法，最终执行的还是接口的具体实现里面实现的功能。

理解访问者模式

4：为何不在Component中实现回调visit方法

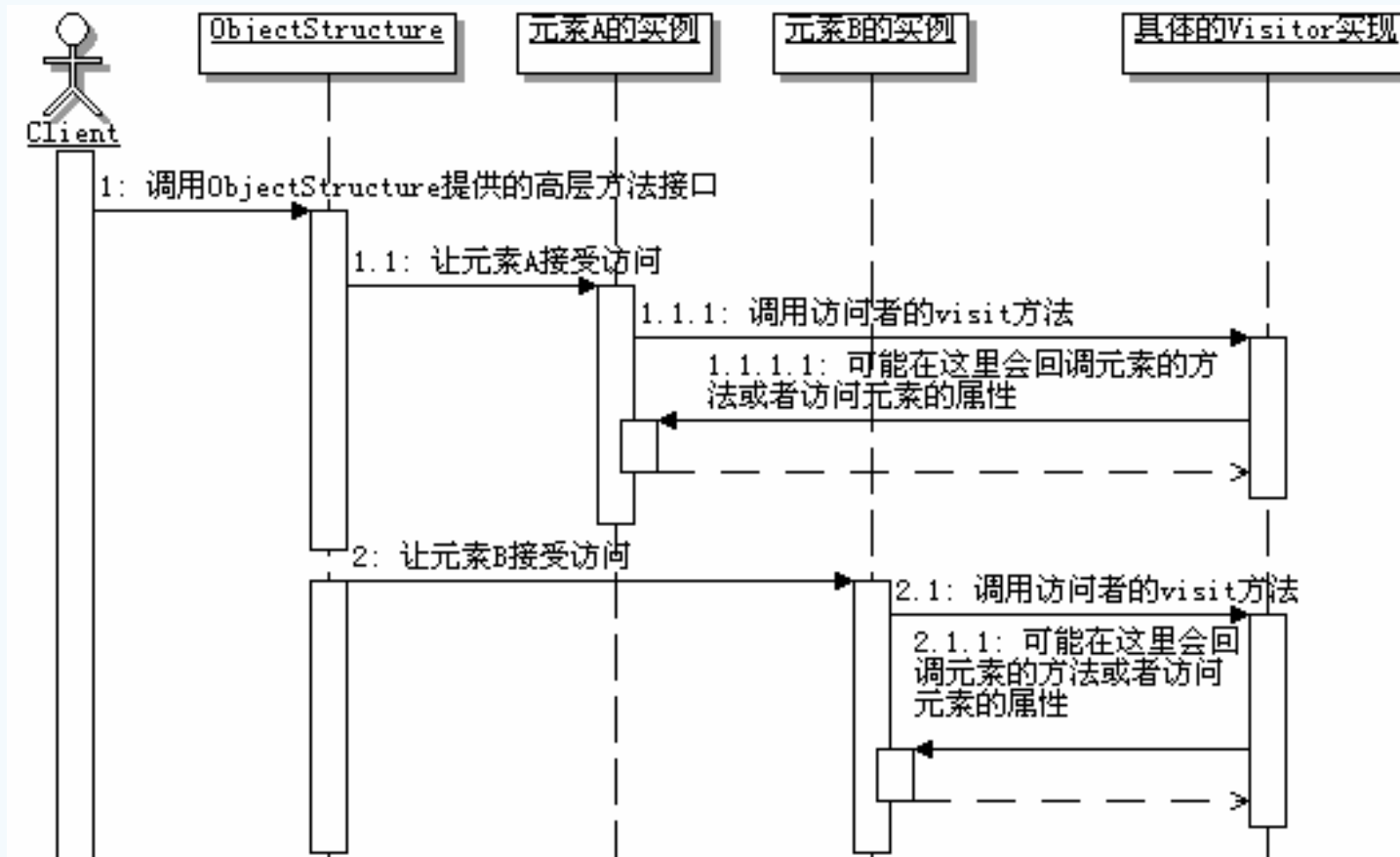
在看上面的示例的时候，细心的朋友会发现，在企业客户对象和个人客户对象中实现的accept方法从表面上看是相似的，都需要回调访问者的方法，可能就会有朋友想，为什么不把回调访问者方法的调用语句放到父类中去，那样不就可以复用了吗？

请注意，这是不可以的，虽然看起来是相似的语句，但其实是不同的，主要的玄机就在传入的this身上。this是代表当前的对象实例的，在企业客户对象中传递的就是企业客户对象的实例，在个人客户对象中传递的就是个人客户对象的实例，这样在访问者的实现中，就可以通过这不同的对象实例来访问不同的实例对象的数据了。

如果把这句话放到父类中，那么传递的就是父类对象的实例，是没有子对象的数据的，因此这句话不能放到父类中去。

理解访问者模式

5: 访问者模式的调用顺序示意图



做最好的在线学习社区

网 址: <http://sishuok.com>

咨询QQ: 2371651507

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送

理解访问者模式

6: 空的访问方法

并不是所有的访问方法都需要实现，由于访问者模式默认的是访问对象结构中的所有元素，因此在实现某些功能的时候，如果不需要涉及到某些元素的访问方法，这些方法可以实现成为空的，比如：这个访问者只想要处理组合对象，那么访问叶子对象的方法就可以为空，虽然还是需要访问所有的元素对象。

还有一种就是有条件接受访问，在自己的accept方法里面进行判断，满足要求的接受，不满足要求的，就相当于空的访问方法，什么都不用做。

理解访问者模式

n 操作组合对象结构

对于使用组合模式构建的组合对象结构，对外有一个统一的外观，要想添加新的功能也不是很困难，只要在组件的接口上定义新的功能就可以了，麻烦的是这样一来，需要修改所有的子类。而且，每次添加一个新功能，都需要这么痛苦一回，修改组件接口，然后修改所有的子类，这是相当糟糕的。

为了让组合对象结构更灵活、更容易维护和更好的扩展性，接下来把它改造成访问者模式和组合模式组合来实现。这样在今后再进行功能改造的时候，就不需要再改动这个组合对象结构了。

访问者模式和组合模式组合使用的思路：首先把组合对象结构中的功能方法分离出来，虽然维护组合对象结构的方法也可以分离出来，但是为了维持组合对象结构本身，这些方法还是放在组合对象结构里面；然后把这些功能方法分别实现成为访问者对象，通过访问者模式添加到组合的对象结构中去。

下面通过访问者模式和组合模式组合来实现如下功能：输出对象的名称，在组合对象的名称前面添加“节点：”，在叶子对象的名称前面添加“叶子：”。

理解访问者模式

n 小结现在的程序结构

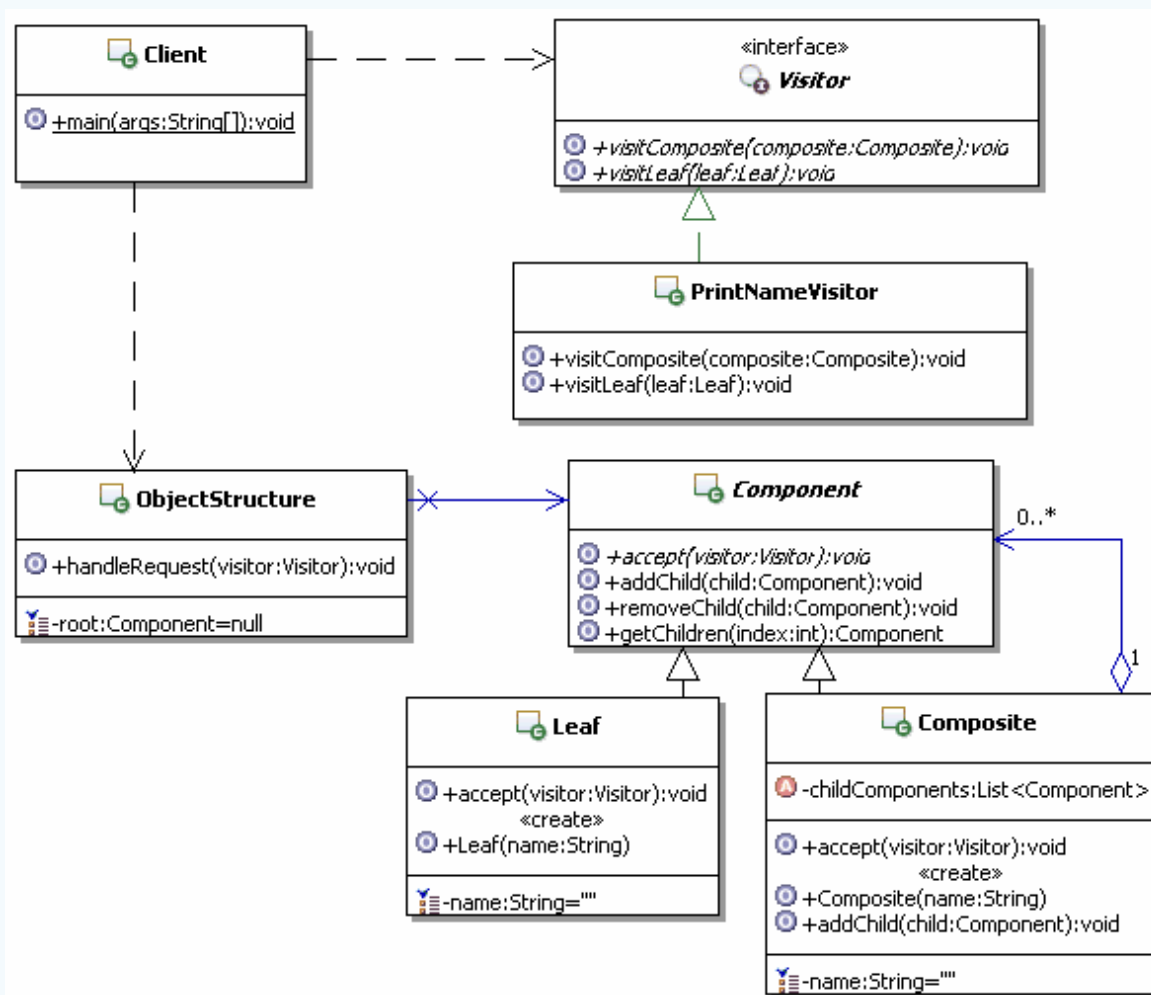
前面是分步的示范，大家已经体会了一番，接下来小结一下。

如同前面的示例，访问者的方法就相当于作用于组合对象结构中各个元素的操作，是一种通用的表达，同样的访问者接口和同样的方法，只要提供不同的访问者具体实现，就表示不同的功能。

同时在组合对象中，接受访问的方法，也是一个通用的表达，不管你是什么样的功能，统统接受就好了，然后回调回去执行真正的功能。这样一来，各元素的类就不用再修改了，只要提供不同的访问者实现，然后通过这个通用表达，就结合到组合对象中来了，就相当于给所有的对象提供了新的功能。

示例的整体结构，

理解访问者模式



做最好的在线学习社区

网 址: <http://sishuok.com>

咨询QQ: 2371651507

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送

理解访问者模式

n 谁负责遍历所有元素对象

在访问者模式中，访问者必须要能够访问到对象结构中的每个对象，因为访问者要为每个对象添加功能，为此特别在模式中定义出一个ObjectStructure来，然后由ObjectStructure来负责遍历访问一系列对象中的每个对象。

1: 在ObjectStructure迭代所有的元素时，又分成两种情况。

- (1) 一种是元素的对象结构是通过集合来组织的，那么直接在ObjectStructure中对集合进行迭代，对每一个元素调用accept就好了。如同前面示例所采用的方式。
- (2) 另一种情况是元素的对象结构是通过组合模式来组织的，通常可以构成对象树，这种情况一般就不需要在ObjectStructure中迭代了，而通常的做法是在组合对象的accept方法里面，递归遍历它的子元素，然后调用子元素的accept方法，如同前面示例中Composite的实现，在accept方法里面进行递归调用子对象的操作。

理解访问者模式

2: 不需要ObjectStructure的时候

在实际开发中，有一种典型的情况可以不需要ObjectStructure对象，那就是只有一个被访问对象的时候。只有一个被访问对象，当然就不需要使用ObjectStructure来组合和迭代了，只要调用这个对象就好了。

事实上还有一种情况也可以不使用ObjectStructure，比如上面访问的组合对象结构，从客户端的角度看，他访问的其实就是一个对象，因此可以把ObjectStructure去掉，然后直接从客户端调用元素的accept方法。

3: 有些时候，遍历元素的方法也可以放到访问者当中去，当然也是需要递归遍历它的子元素的。出现这种情况的主要原因是：想在访问者中实现特别复杂的遍历，访问者的实现依赖于对象结构的操作结果。

前面的示例已经实现了：使用访问者模式和组合模式组合来实现了输出名称的功能，如果现在要实现把组合的对象结构按照树的形式输出，就是按照在组合模式中示例的那样，输出如下的树形结构：

理解访问者模式

```
+服装+  
+男装+  
-衬衣+  
-夹克+  
+女装+  
-裙子+  
-套装+
```

要实现这个功能，在组合对象结构中去遍历子对象的方式就比较难于实现，因为要输出这个树形结构，需要控制每个对象在输出的时候，向后的退格数量，这个需要在对象结构的循环中来控制，这种功能可以选择在访问者当中去遍历对象结构。

做最好的在线学习社区

网 址：<http://sishuok.com>

咨询QQ：2371651507

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送

理解访问者模式

n 访问者模式的优缺点

- 1: 好的扩展性
- 2: 好的复用性
- 3: 分离无关行为
- 4: 对象结构变化很困难
- 5: 破坏封装

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

做最好的在线学习社区

网 址：<http://sishuok.com>

咨询QQ：2371651507

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送