  　　组合模式(Composite Pattern)有时候又叫做部分-整体模式，它使我们树型结构的问题中，模糊了简单元素和复杂元素的概念  
  ，客户程序可以向处理简单元素一样来处理复杂元素,从而使得客户程序与复杂元素的内部结构解耦。组合模式让你可以优化处理递  
  归或分级数据结构。有许多关于分级数据结构的例子，使得组合模式非常有用武之地。关于分级数据结构的一个普遍性的例子是你每  
  次使用电脑时所遇到的:文件系统。文件系统由目录和文件组成。每个目录都可以装内容。目录的内容可以是文件，也可以是目录。  
  按照这种方式，计算机的文件系统就是以递归结构来组织的。如果你想要描述这样的数据结构，那么你可以使用组合模式Composite。  
　　定义(GoF《设计模式》)：将对象组合成树形结构以表示“部分整体”的层次结构。组合模式使得用户对单个对象和使用具有一致性。  
　　涉及角色：  
　　1.Component 是组合中的对象声明接口，在适当的情况下，实现所有类共有接口的默认行为。声明一个接口用于访问和管理Component  
子部件。  
　　2.Leaf 在组合中表示叶子结点对象，叶子结点没有子结点。  
　　3.Composite 定义有枝节点行为，用来存储子部件，在Component接口中实现与子部件有关操作，如增加(add)和删除

(remove)等。  
　　  
　　适用性  
　　以下情况下适用Composite模式：  
　　1．你想表示对象的部分-整体层次结构  
　　2．你希望用户忽略组合对象与单个对象的不同，用户将统一地使用组合结构中的所有对象。  
　　总结  
组合模式解耦了客户程序与复杂元素内部结构，从而使客户程序可以向处理简单元素一样来处理复杂元素。  
如果你想要创建层次结构，并可以在其中以相同的方式对待所有元素，那么组合模式就是最理想的选择。本章使用了一个文件  
系统的例子来举例说明了组合模式的用途。在这个例子中，文件和目录都执行相同的接口，这是组合模式的关键。通过执行相

同的接口，你就可以用相同的方式对待文件和目录，从而实现将文件或者目录储存为目录的子级元素。

