

```

1  public interface IVisitable {
2      public void accept(IVisitor visitor);
3  }
4
5  public class CustomerGroup implements IVisitable{
6      private ArrayList<Customer> customers;
7      public CustomerGroup(){
8          customers = new ArrayList<Customer>();
9      }
10     public void accept(IVisitor visitor)
11     {
12         visitor.visit(this);
13         for (Iterator<Customer> it=customers.iterator(); it.hasNext();)
14             ((Customer)it.next()).accept(visitor);
15     }
16     public void addCustomer(Customer customer)
17     {
18         customers.add(customer);
19     }}
20
21 public class Customer implements IVisitable{
22     private String name;
23     private ArrayList<Order> orders;
24
25     public void accept(IVisitor visitor)
26     {
27         visitor.visit(this);
28         for (Iterator<Order> it=orders.iterator(); it.hasNext();)
29             ((IVisible)it.next()).accept(visitor);
30     }
31     public String getName() {
32         return name;
33     }
34     public void setName(String name) {
35         this.name = name;
36     }
37     public void addOrder(Order order)
38     {
39         orders.add(order);
40     }
41     public Customer(String name)
42     {
43         orders = new ArrayList<Order>();
44         this.name = name;
45     }}
46
47 public class Order implements IVisitable {
48     private String name;
49     private ArrayList<Item> items;
50
51     public Order(String name)
52     {
53         this.name = name;
54     }
55
56     public Order(String name, String itemName)
57     {
58         items = new ArrayList<Item>();
59         this.name = name;
60         this.addItem(new Item(itemName));
61     }
62     public String getName() {
63         return name;
64     }
65     public void setName(String name) {
66         this.name = name;
67     }
68     public void accept(IVisitor visitor)
69     {
70         visitor.visit(this);
71         for (Iterator<Item> it=items.iterator(); it.hasNext();)
72             ((Item)it.next()).accept(visitor);

```

```

73     }
74     public void addItem(Item item)
75     {
76         items.add(item);
77     }
78 }
79 public class Item implements IVisitable{
80     private String name;
81
82     public void accept(IVisitor visitor)
83     {
84         visitor.visit(this);
85     }
86     public Item(String name)
87     {
88         this.name = name;
89     }
90     public String getName() {
91         return name;
92     }
93     public void setName(String name) {
94         this.name = name;
95     }
96 }
97 public interface IVisitor {
98     public void visit(Customer customer);
99     public void visit(Order order);
100    public void visit(Item item);
101    public void visit(CustomerGroup group);
102 }
103
104 public class GeneralReport implements IVisitor{
105     private int customersNo;
106     private int ordersNo;
107     private int itemsNo;
108
109     public void visit(CustomerGroup group)
110     {
111         System.out.println(group.toString());
112     }
113     public void visit(Customer customer)
114     {
115         System.out.println(customer.getName());
116         customersNo ++;
117     }
118     public void visit(Order order)
119     {
120         System.out.println(order.getName());
121         ordersNo++;
122     }
123     public void visit(Item item)
124     {
125         System.out.println(item.getName());
126         itemsNo++;
127     }
128     public void displayResults()
129     {
130         System.out.println("Nr of customers:" + customersNo);
131     }
132 }
133 public class Main {
134     public static void main(String[] args) {
135         Customer c = new Customer("customer1");
136         c.addOrder(new Order(".order1", "..item1"));
137         c.addOrder(new Order(".order2", "..item1"));
138         c.addOrder(new Order(".order3", "..item1"));
139
140         Customer c2 = new Customer("customer2");
141         Order o = new Order(".order_a");
142         o.addItem(new Item("..item_a1"));
143         o.addItem(new Item("..item_a2"));
144         o.addItem(new Item("..item_a3"));

```

```
145         c2.addOrder(o);
146         c2.addOrder(new Order(".order_b", "..item_b1"));
147
148         CustomerGroup customers = new CustomerGroup();
149         customers.addCustomer(c);
150         customers.addCustomer(c2);
151
152         GeneralReport visitor = new GeneralReport();
153         customers.accept(visitor);
154         visitor.displayResults();
155     }
156 }
157
```