## 访问者（Visitor）

### Intent

为一个对象结构（比如组合结构）增加新能力。

### Class Diagram

* Visitor：访问者，为每一个 ConcreteElement 声明一个 visit 操作
* ConcreteVisitor：具体访问者，存储遍历过程中的累计结果
* ObjectStructure：对象结构，可以是组合结构，或者是一个集合。

### Implementation

public interface Element {
 void accept(Visitor visitor);
}

class CustomerGroup {

 private List<Customer> customers = new ArrayList<>();

 void accept(Visitor visitor) {
 for (Customer customer : customers) {
 customer.accept(visitor);
 }
 }

 void addCustomer(Customer customer) {
 customers.add(customer);
 }
}

public class Customer implements Element {

 private String name;
 private List<Order> orders = new ArrayList<>();

 Customer(String name) {
 this.name = name;
 }

 String getName() {
 return name;
 }

 void addOrder(Order order) {
 orders.add(order);
 }

 public void accept(Visitor visitor) {
 visitor.visit(this);
 for (Order order : orders) {
 order.accept(visitor);
 }
 }
}

public class Order implements Element {

 private String name;
 private List<Item> items = new ArrayList();

 Order(String name) {
 this.name = name;
 }

 Order(String name, String itemName) {
 this.name = name;
 this.addItem(new Item(itemName));
 }

 String getName() {
 return name;
 }

 void addItem(Item item) {
 items.add(item);
 }

 public void accept(Visitor visitor) {
 visitor.visit(this);

 for (Item item : items) {
 item.accept(visitor);
 }
 }
}

public class Item implements Element {

 private String name;

 Item(String name) {
 this.name = name;
 }

 String getName() {
 return name;
 }

 public void accept(Visitor visitor) {
 visitor.visit(this);
 }
}

public interface Visitor {
 void visit(Customer customer);

 void visit(Order order);

 void visit(Item item);
}

public class GeneralReport implements Visitor {

 private int customersNo;
 private int ordersNo;
 private int itemsNo;

 public void visit(Customer customer) {
 System.out.println(customer.getName());
 customersNo++;
 }

 public void visit(Order order) {
 System.out.println(order.getName());
 ordersNo++;
 }

 public void visit(Item item) {
 System.out.println(item.getName());
 itemsNo++;
 }

 public void displayResults() {
 System.out.println("Number of customers: " + customersNo);
 System.out.println("Number of orders: " + ordersNo);
 System.out.println("Number of items: " + itemsNo);
 }
}

public class Client {
 public static void main(String[] args) {
 Customer customer1 = new Customer("customer1");
 customer1.addOrder(new Order("order1", "item1"));
 customer1.addOrder(new Order("order2", "item1"));
 customer1.addOrder(new Order("order3", "item1"));

 Order order = new Order("order\_a");
 order.addItem(new Item("item\_a1"));
 order.addItem(new Item("item\_a2"));
 order.addItem(new Item("item\_a3"));
 Customer customer2 = new Customer("customer2");
 customer2.addOrder(order);

 CustomerGroup customers = new CustomerGroup();
 customers.addCustomer(customer1);
 customers.addCustomer(customer2);

 GeneralReport visitor = new GeneralReport();
 customers.accept(visitor);
 visitor.displayResults();
 }
}

customer1
order1
item1
order2
item1
order3
item1
customer2
order\_a
item\_a1
item\_a2
item\_a3
Number of customers: 2
Number of orders: 4
Number of items: 6

### JDK

* javax.lang.model.element.Element and javax.lang.model.element.ElementVisitor
* javax.lang.model.type.TypeMirror and javax.lang.model.type.TypeVisitor