

## Order.java

```
1 package model.catalog;
2
3 import java.text.SimpleDateFormat;
14
15 @XmlRootElement(name="order")
16 public class Order {
17     private static final double HST_COST = 0.13;
18     private static final double SHIPPING_COST = 5.0;
19     private static final double SHIPPING_BUFFER_COST = 100.0;
20     private static final SimpleDateFormat sdf = new
SimpleDateFormat("yyyy-MM-dd");
21
22     private List<Item> items;
23     private Account customer;
24     private double total;
25     private double shipping;
26     private double HST;
27     private double grandTotal;
28
29     private int id;
30     private String submitted;
31
32     public Order() {
33         this(new ArrayList<Item>());
34     }
35
36     public Order(List<Item> items) {
37         this(0, items);
38     }
39
40     public Order(int id, List<Item> items) {
41         this.id = id;
42         this.items = items;
43     }
44
45     @XmlElementWrapper(name="items")
46     @XmlElement(name="item")
47     public List<Item> getItems() {
48         return items;
49     }
50
51     public void setItems(List<Item> items) {
52         this.items = items;
53     }
```

## Order.java

```
54
55 @XmlElement(name="customer")
56 public Account getCustomer() {
57     return customer;
58 }
59
60 public void setCustomer(Account customer) {
61     this.customer = customer;
62 }
63
64 @XmlElement(name="total")
65 public double getTotal() {
66     return Utils.round(total, 2);
67 }
68
69 public void setTotal(double total) {
70     this.total = Utils.round(total, 2);
71 }
72
73 @XmlElement(name="shipping")
74 public double getShipping() {
75     return Utils.round(shipping, 2);
76 }
77
78 public void setShipping(double shipping) {
79     this.shipping = Utils.round(shipping, 2);
80 }
81
82 @XmlElement(name="HST")
83 public double getHST() {
84     return Utils.round(HST, 2);
85 }
86
87 public void setHST(double hST) {
88     HST = Utils.round(hST, 2);
89 }
90
91 @XmlElement(name="grandTotal")
92 public double getGrandTotal() {
93     return Utils.round(grandTotal, 2);
94 }
95
96 public void setGrandTotal(double grandTotal) {
97     this.grandTotal = Utils.round(grandTotal, 2);
```

## Order.java

```
98     }
99
100    @XmlAttribute(name="id")
101    public int getId() {
102        return id;
103    }
104
105    public void setId(int id) {
106        this.id = id;
107    }
108
109    @XmlAttribute(name="submitted")
110    public String getSubmitted() {
111        return submitted;
112    }
113
114    public void setSubmitted(String submitted) {
115        this.submitted = submitted;
116    }
117
118    public void setSubmitted(Date submitted) {
119        this.submitted = sdf.format(submitted);
120    }
121
122    public synchronized void computeGrandTotal() {
123        this.setGrandTotal(this.getTotal() + this.getHST() +
124        this.getShipping());
125    }
126
127    public synchronized void computeTotalHST() {
128        this.setHST((this.getTotal() + this.getShipping()) *
129        HST_COST);
130    }
131
132    public synchronized void computeShipping() {
133        if (this.getTotal() > 0 && this.getTotal() <=
134        SHIPPING_BUFFER_COST) {
135            this.setShipping(SHIPPING_COST);
136        } else {
137            this.setShipping(0.0);
138        }
139    }
140
141    public synchronized void computeTotalCost() {
```

## Order.java

```
139         double total = 0.0;
140         for (Item it: this.getItems()) {
141             total += it.computeTotalPrice();
142         }
143         this.setTotal(total);
144     }
145
146     public synchronized void computeAllCosts() {
147         this.computeTotalCost();
148         this.computeTotalHST();
149         this.computeShipping();
150         this.computeGrandTotal();
151     }
152
153     public synchronized void clearCart() {
154         this.items.clear();
155     }
156
157     public synchronized void clearOrder() {
158         this.clearCart();
159         this.computeAllCosts();
160     }
161 }
162
```