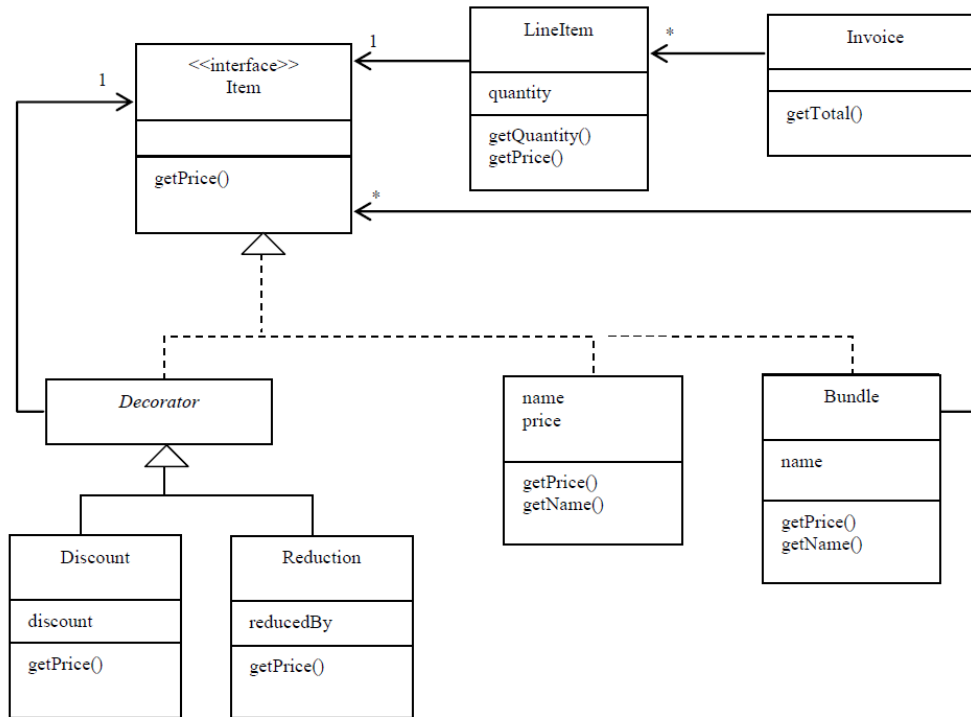


(10) A class diagram for managing invoices appears on the next page. Note the design involves both the Decorator pattern and the Composite pattern.

Consider an invoice with one line item. The line item has a quantity of 3 for a bundle named "Special". This bundle comprises two products: a CD named O2014 (with a price reduction of \$10 and then discounted by 10%), and another CD named O2013 (which is not discounted or reduced). The bundle itself is discounted by 10%.

a) Create an object diagram for this situation. Your object diagram must show links and the values assigned to the fields of each object.

b) Use a sequence diagram to show all messages sent when the invoice receives the getTotal() message and determines its total.



**Notes on methods:**

getPrice() in Discount is:

```
return item.getPrice() * (1 - (discount/100))
```

getPrice() in Reduction is:

```
return item.getPrice() - reducedBy
```

getPrice() in Product is:

```
return price
```

getTotal() in Invoice is:

```
total = 0
```

```
for each Lineltem:
```

```
total = total + lineltem.getPrice() * lineltem.getQuantity()
```

```
return total
```

getPrice() in Bundle is:

```
total = 0
```

```
for each Item:
```

```
total = total + item.getPrice()
```

```
return total
```

getPrice() in Lineltem is:

```
return item.getPrice()
```

getQuantity() in Lineltem is:

```
return quantity
```