

```
interface Visitor {
    void visit (Item item);
    void visit (Pack pack);
}
```

```
interface Product {
    void accept (Visitor visitor);
    BigDecimal getPrice ();
}
```

```
class Item implements Product {
    private BigDecimal price;
    public Item (BigDecimal price) {
        this.price = price;
    }
```

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

```
@Override
    public BigDecimal getPrice () { return price; }
```

```

}

class Pack implements Product {
    private List <Product> products = new ArrayList <> ();
    public void addProduct (Product product) { products.add (product); }
```

```
@Override
    public void accept (Visitor visitor) { visitor.visit (this); }
    public List <Product> getProducts () { return products; }
```

```
@Override
    public BigDecimal getPrice () {
        return products.stream()
            .map (Product::getPrice)
            .reduce (BigDecimal.ZERO, BigDecimal::add);
    }
```

```

}
```

```
class PriceAdjustVisitor implements Visitor {
```

```
    private BigDecimal minPrice;
```

```
    public PriceAdjustVisitor (BigDecimal minPrice) { this.minPrice = minPrice; }
```

```
    @Override
```

```
    public void visit (Item item) {
```

```
        if (item.getPrice().compareTo(minPrice) < 0 {
```

```
            item.setPrice(minPrice);
```

```
        }
```

```
    }
```

```
    @Override
```

```
    public void visit (Pack pack) {
```

```
        for (Product product: pack.getProducts()) {
```

```
            product.accept(this);
```

```
        }
```

```
    }
```

```
}
```

```
class ItemCounterVisitor implements Visitor {
```

```
    private BigDecimal minPrice;
```

```
    private int count = 0;
```

```
    public ItemCounterVisitor (BigDecimal minPrice) { this.minPrice = minPrice; }
```

```
    @Override
```

```
    public void visit (Item item) {
```

```
        if (item.getPrice().compareTo(minPrice) < 0) { count++; }
```

```
    }
```

```
    @Override
```

```
    public void visit (Pack pack) {
```

```
        for (Product product: pack.getProducts()) {
```

```
            product.accept(this);
```

```
        }
```

```
    }
```

```
    public int getCount() { return count; }
```

```
}
```