pkg

<<interface>>
IStockInventory

+ addCap(category : String, cap : int) : void
+ getCap(category : String) : int
+ addItem(item : String, category : String, quantity : String, ppu : String) : void

---

StockInventoryImpl

– uniqueInventoryInstance : StockInventoryImpl
– cards : List<String>
– items : Map<String,String[]>
– categoryCap : Map<String,String>

+ getInventoryInstance() : StockInventoryImpl
– StockInventoryImpl() : void
+ addItem(item : String, category : String, quantity : String, ppu : String) : void
+ getCap(category : String) : int
+ addCap(category : String, cap : int) : void

<<interface>>
IProcessOrder

+ processOrder(order : Map<String,String>, card : String) : int

---

Billing

– uniqueInventoryInstance : StockInventoryImpl
– cardNumber : String
– orderDetails : Map<String,String>

+ main() : void
+ setInventory(uniqueInventoryInstance : StockInventoryImpl) : void
+ setCategoryCap(uniqueInventoryInstance : StockInventoryImpl) : void
+ setCards(uniqueInventoryInstance : StockInventoryImpl) : void
+ readInput(uniqueInventoryInstance : StockInventoryImpl) : int
+ setOrder(orderFilename : String, uniqueInventoryInstance : StockInventoryImpl) : int

---

ProcessOrderImpl

– capError : boolean
– stockError : boolean
– itemsExceedingCap : List<String>
– itemsExceedingStock : List<String>

+ processOrder(order : Map<String,String>, String : int, card : String) : int
+ checkCategoryCapExceeds (uniqueInventoryInstance : StockInventoryImpl, order : Map<String,String>) : void
+ checkStockExceeds(uniqueInventoryInstance : StockInventoryImpl, order : Map<String,String>) : void

---

<<interface>>
IOutputWriterFactory

+ writeText(fileText : String, items : List<String>) : void

CSVFactoryImpl

+ writeText(fileText : String, items : List<String>) : void

TextFactoryImpl

+ writeText(fileText : String, items : List<String>) : void