## <<interface>> StockTrackable

+ receive(qty: int): void
+ issue(qty: int): boolean

# <<interface>> PriceAdjustable

+ applyDiscount(percent: double): void
+ applySurcharge(percent: double): void

# <<abstract>> InventoryItem

sku: Stringname: Stringquantity: int

- unitPrice: double

+ InventoryItem(sku, name, unitPrice)

+ getSku(): String
+ getName(): String

+ getQuantity(): int
+ getUnitPrice(): double

# setQuantity(q: int): void
+ value(): double {final}

+ category(): String {abstract}

+ toString(): String

#### PerishableItem

- shelfLifeDays: int

+ PerishableItem(sku, name, unitPrice, shelfLifeDays)

+ getShelfLifeDays(): int

+ category(): String

+ receive(qty: int): void

+ issue(qty: int): boolean
+ applyDiscount(percent: double): void

+ applySurcharge(percent: double): void

+ toString(): String

#### **ElectronicItem**

- warrantyMonths: int

+ ElectronicItem(sku, name, unitPrice, warrantyMonths)

+ receive(qty: int): void

+ issue(qty: int): boolean
+ applyDiscount(percent: double): void

+ applySurcharge(percent: double): void

#### ClothingItem

- size: String

- material: String

+ ClothingItem(sku, name, unitPrice, size, material)

+ getSize(): String

+ getMaterial(): String
+ category(): String

+ receive(qty: int): void

+ issue(qty: int): boolean

+ applyDiscount(percent: double): void

### InventoryManager

- items: List<InventoryItem>

+ InventoryManager()

+ addItem(item: InventoryItem): void
+ findBySku(sku: String): InventoryItem

+ totalValue(): double

+ totalQuantityByCategory(category: String): int

+ issue(sku: String, qty: int): boolean
+ issue(item: InventoryItem, qty: int): boolean

### UML Notation Legend:

### **Visibility:**

+ public

- private

# protected

## Relationships:

———

▷ Inheritance (extends, hollow triangle)

- - -  $\triangleright$  Realization (implements, dashed line + hollow triangle)

-----> Association (uses, regular arrow)

### Modifiers:

italic = abstract method or class

{final} = final method

<<interface>> = interface stereotype

<abstract>> = abstract class stereotype

Multiplicity: \* = zero or many