

| | | | | | |
|--|--|--|--|---|--|
| <pre> + priceSystem implements Serializable fields - salesPrice :int ~ currentPrice :int ~ initialPrice :int - priceHistory:HashMap<Integer, ArrayList<Date>> - currentDate :Date - onSale :boolean final <u>logger</u>:Logger final <u>consoleHandler</u>:Handler constructors ~ priceSystem(initialPrice: int) methods ~ checkSalePeriod(startDate: Date, endDate: Date):boolean ~ setSaleByPrice(salesPrice: Object, startDate: String, endDate: String):void ~ showPrice():int </pre> | <pre> + ProductManager fields - inventory:Inventory ~ totalRevenue :int ~ totalCost :int - ordersys:orderSystem - orderHistory:ArrayList<String> constructors ~ ProductManager(inventory:Inventory) methods - transferList(shoppingList:HashMap<Integer, Integer>):HashMap<Product, Integer> ~ modifyStatus(shoppingList:HashMap<Integer, Integer>):void ~ modifyProduct(product:Product, quantity: Integer):void ~ getSaleProducts():ArrayList<Product> - order(product:Product, difference: int):void ~ checkInStock(UPC: int[]):boolean ~ updateAccount(product:Product, quantity: Integer):void ~ reshelveProduct(upc:int, newLocation: int[]):void ~ getInventory():Inventory </pre> | <pre> + Product implements Serializable fields - name:String - upc:int ~ location :int[] ~ quantity:int ~ threshold:int ~ cost:int - section:String ~ priceSystem:priceSystem ~ salesHistory:ArrayList<Integer> final <u>logger</u>:Logger final <u>consoleHandler</u>:Handler constructors ~ Product(name:String, upc:int, quantity:int, section:String, price:int) methods ~ setPriceSystem(price:int):void ~ setName(name:String):void ~ setUpc(upc:int):void ~ setThreshold(threshold:int):void ~ setQuantity(quantity:int):void ~ showPrice():int ~ showUPC():int ~ setLocation(location: int[]):void ~ getLocation():int[] ~ toString():String </pre> | <pre> + Receiver fields - name:String - productManager:ProductManager - inventory:Inventory - reshelver:Reshelver final <u>logger</u>:Logger final <u>consoleHandler</u>:Handler constructors ~ Receiver(name:String, productManager: ProductManager) methods ~ recieveOrder(upc:int, quantity:int):void </pre> | <pre> + GroceryDemo fields - <u>cashier</u>:Cashier - <u>manager</u>:Manager - <u>reshelver</u>:Reshelver - <u>receiver</u>:Receiver - <u>productManager</u>:ProductManager constructors - GroceryDemo(cashierName: String, managerName: String, reshelverName: String, receiverName: String, path: String) methods - runEventLine(event: String):void - classifyAndExcuteAction(user:String, action:String, actiondetail: String):void - cashierAction(action:String, actiondetail: String):void - managerAction(action:String, actiondetail: String):void - receiverAction(action:String, actiondetail: String):void - reshelverAction(action:String, actiondetail: String):void - questionAnswer(action:String, actiondetail: String):void + <u>main</u>(args:String[]):void </pre> | <pre> ~ orderSystem fields - order_index:int - orderHistory:ArrayList<String> - pendingOrder:ArrayList<String> - productHistory:HashMap<Product, String> ~ final <u>logger</u>:Logger ~ final <u>consoleHandler</u>:Handler - inventory:Inventory - receiver:Receiver constructors ~ orderSystem(productManager: ProductManager, receiver: Receiver) methods ~ handleOrder(product:Product, quantity:int):void ~ hasArrived(orderNum:int):void </pre> |
|--|--|--|--|---|--|

| | | | | |
|--|--|--|--|---|
| <pre> + Manager fields - name:String - productManager:ProductManager final <u>logger</u>:Logger final <u>consoleHandler</u>:Handler constructors ~ Manager(name:String, productManager: ProductManager) methods ~ setSaleProduct(upc:int, discount:Object, startDate: String, endDate: String):void ~ checkProfit():void </pre> | <pre> + Reshelver fields - name:String ~ productManager:ProductManager final <u>logger</u>:Logger final <u>consoleHandler</u>:Handler constructors ~ Reshelver(name:String, productManager: ProductManager) methods ~ changeLocationProduct(upc:int, location: int[]):void </pre> | <pre> + Cashier fields - name:String - productManager:ProductManager - shoppingList:HashMap<Integer, Integer> - inventory:Inventory final <u>logger</u>:Logger final <u>consoleHandler</u>:Handler constructors ~ Cashier(name:String, productManager: ProductManager) methods ~ scanItems(upcList:ArrayList<Integer>):void ~ customerPaid():void </pre> | <pre> + csvReader fields - productList:ArrayList<Product> final <u>logger</u>:Logger final <u>consoleHandler</u>:Handler constructors ~ csvReader() methods ~ readCsv(path: String):ArrayList<Product> </pre> | <pre> + Inventory fields ~ productList:ArrayList<Product> final <u>logger</u>:Logger final <u>consoleHandler</u>:Handler constructors ~ Inventory(initialInventoryPath: String) methods ~ findByUpc(upc:int):Product ~ findByName(name: String):Product ~ saveInventory(filePath: String):void + <u>toString</u>():String ~ addProducts(upc:int, name:String, quantity:int, section:String, price:int):void </pre> |
|--|--|--|--|---|