Sale

Attributes
- item: Item
- buyer: Buyer
- date: Date
- sent: Date
- package: Package

Operations
+ Sale(item Item, buyer: Buyer, date: Date, sent: Date, package: Package)
+ getItem(): Item
+ getBuyer(): Buyer
+ getDate(): Date
+ getSent(): Date
+ getPackage(): Package

Date

Attributes
- day: int
- month: int
- year: int

Operations
+ Date(day: int, month: int, year: int)
+ getDay(): int
+ getMonth(): int
+ getYear(): int

Store

Attributes
- users: List<Users>
- packages: List<Packages>
- itemsStock: List<Items>
- benefits: Double
- costs: Double
- itemsSold: List<Items>
- sales: List<Sale>

Operations
+ main(args: String[]): static void

Box

Attributes
- depth: int

Operations
+ Box(width: int, height: int, depth: int)
+ getDepth(): int
+ setDepth(depth: int)
+ isSuitable(size: int[3]): boolean

Packages

Attributes
- width: int
- height: int

Operations
+ Packages(w: int, h: int)
+ getWidth(): int
+ getHeight(): int
+ setWidth(width: int)
+ setHeight(heigth: int)

Envelop

Attributes
- name: String

Operations
+ Envelop(width: int, height:int, name String)
+ getName(): String
+ setName(name: String)
+ isSuitable(size: int[2]): boolean

Sellers

Attributes
- account: String
- toSell: List<Items>
- sold: List<Items>

Operations
+ Sellers(account: String, name: string, user: string, password: string)
+ sell(item: Item)
+ addItem(item: Item)
- deposit(price: Double): boolean

Users

Attributes
- name: string
- id: string
- password: string

Operations
+ Users(name: string, id: string, password: string)
+ getName(): String
+ getID(): String
+ getPassword(): String
+ setName(name: String)
+ login(pass: String): Boolean

<<interface>> Comparable

Attributes

Operations
+ compareTo(o: Object): int

Unit

Attributes
- price: Double
- stock: int
- sold: int

Operations
+ Unit(price: int, stock: int, name: string, type: string, size: Double[3], cost: Double)
+ getPrice(): Double
+ calculateProfit(): Double
+ sell(quantity: int): double

Items

Attributes
- name: string
- type: string
- size: Double[3]
- cost: double
- pack: Package

Operations
+ Items(name: string, type: string, size: Double[3], cost: Double)
+ Item()
+ getName(): String
+ getType(): String
+ getSize(): Double[3]
+ getCost(): Double
+ getPackage(): Package
+ setName(name: String)
+ setType(Type: String)
+ setSize(size: Double[3])
+ setCost(cost: Double)
+ assignPackage(lp: List<Packages>)
abstract + getPrice(): Double
abstract + calculateProfit(): Double (abstract)

Buyers

Attributes

Operations
+ Buyers(account: String, name: string, user: string, password: string)
+ but(item: Item)
- pay(price: Double): boolean

Auction

Attributes
- bid: Double
- bidder: Buyer
- deadline: Date

Operations
+ Auction(startingBid: int, deadline:String, name: string, type: string, size: Double[3], cost: Double)
+ getPrice(): Double
+ calculateProfit(): Double
+ makeBid(bidder: Buyer, bid: double)
+ frozen(d: Date): boolean
+ getBuyer(): Buyer
+ getDeadline(): Date

<<interface>> Taxable

Attributes
static final + tax: Double

Operations
+ getPrice(): Double
+ getTaxPrice(): Double
+ getTax(): Double
+ sum(item: Taxable): Taxable

Weighed

Attributes
- stock: Double
- sold: Double
- price: Double

Operations
+ Weighed(price: Double, stock: Double, name: string, type: string, size: Double[3], cost: Double)
+ getPrice(): Double
+ calculateProfit(): Double
+ sell(weight: Double): Double

Admins

Attributes

Operations
+ Admins(name: string, user: string, password: string)
+ ban(user: User): boolean
+ manageAuction(item: Auction, date: String): boolean
+ printStock(items: List<Items>)

We set the Unit and Weighed classes to inherit from Taxable as we think they are required to use it, as this items normally are taxed with IVA.