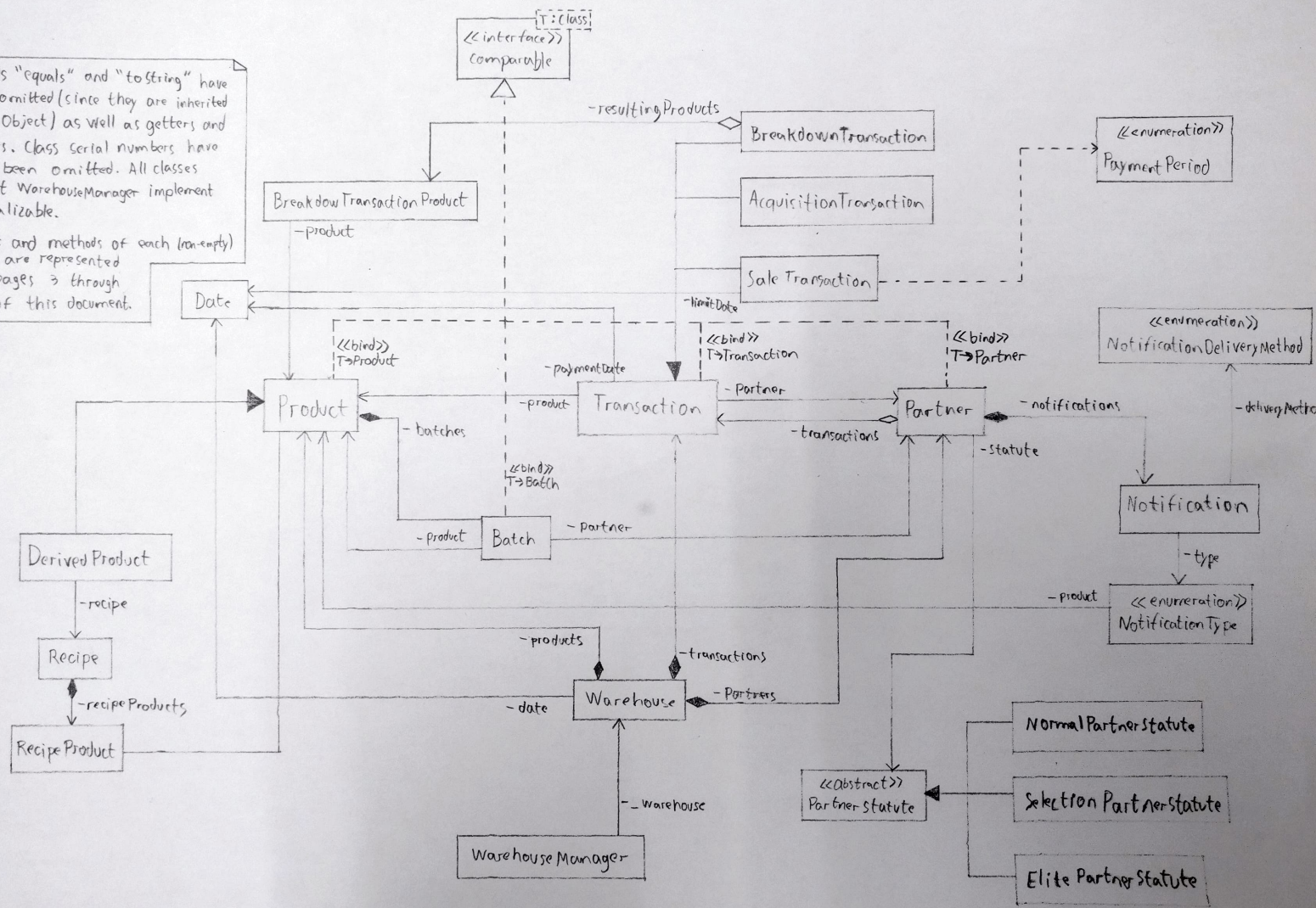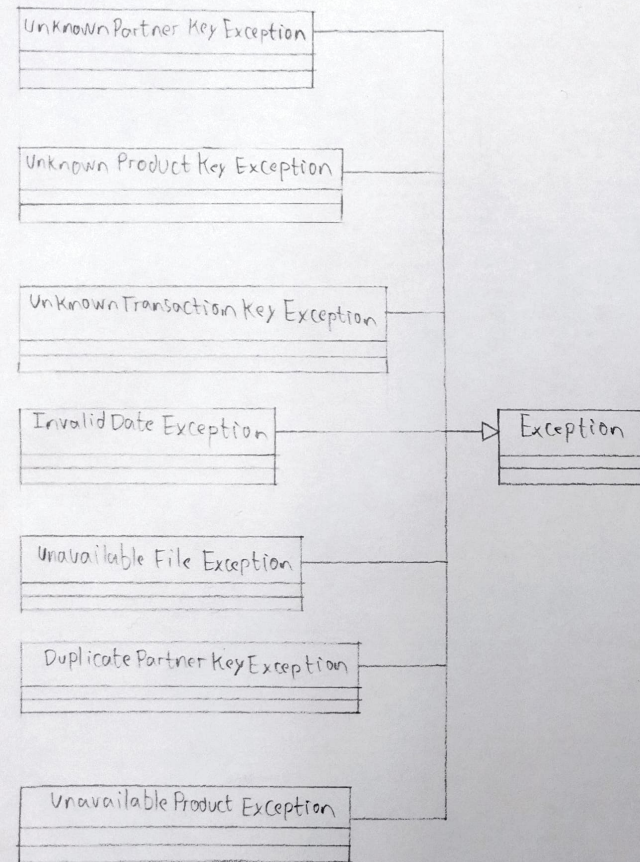Declaro por minha honra que este diagrama foi realizado apenas pelos elementos que constituem o grupo de projeto. Diogo Correia   ist199291

Decisão por minha honra que este diagrama foi realizado apenas pelos elementos que constituem o grupo de projeto. Rafael Sanz e Oliveira   ist199311

T : Class
<<interface>>
Comparable

Methods "equals" and "toString" have been omitted (since they are inherited from Object) as well as getters and setters. Class serial numbers have also been omitted. All classes except WarehouseManager implement Serializable.

Fields and methods of each (non-empty) class are represented on pages 3 through 6 of this document.

- resulting Products

BreakdownTransaction

AcquisitionTransaction

Sale Transaction

<<enumeration>>
Payment Period

Breakdow Transaction Product

- product

Date

- limit Date

<<enumeration>>
Notification Delivery Method

<<bind>>
T→Product

<<bind>>
T→Transaction

<<bind>>
T→Partner

- payment Date

Product

Transaction

Partner

- notifications

- delivery Method

- product

- partner

- batches

- transactions

- statute

Notification

<<bind>>
T→Batch

Batch

- partner

- product

- type

- product

<<enumeration>>
Notification Type

Derived Product

- recipe

Recipe

- products

- transactions

- recipe Products

Recipe Product

- date

Warehouse

- Partners

Normal Partner Statute

Selection Partner Statute

<>
Partner Statute

Warehouse Manager

- warehouse

Elite Partner Statute

```
┌─────────────────────────────┐
│ Unknown Partner Key Exception │───────┐
├─────────────────────────────┤       │
│                             │       │
└─────────────────────────────┘       │
                                       │
┌─────────────────────────────┐       │
│ Unknown Product Key Exception │───────┤
├─────────────────────────────┤       │
│                             │       │
└─────────────────────────────┘       │
                                       │
┌───────────────────────────────────┐ │
│ Unknown Transaction key Exception │─┤
├───────────────────────────────────┤ │
│                                   │ │
└───────────────────────────────────┘ │
                                       │
┌─────────────────────────┐           │      ┌─────────────┐
│ Invalid Date Exception  │───────────┼─────▷│  Exception  │
├─────────────────────────┤           │      ├─────────────┤
│                         │           │      │             │
└─────────────────────────┘           │      └─────────────┘
                                       │
┌─────────────────────────┐           │
│ Unavailable File Exception │─────────┤
├─────────────────────────┤           │
│                         │           │
└─────────────────────────┘           │
                                       │
┌──────────────────────────────┐      │
│ Duplicate Partner Key Exception │────┤
├──────────────────────────────┤      │
│                              │      │
└──────────────────────────────┘      │
                                       │
┌──────────────────────────────┐      │
│ Unavailable Product Exception │──────┘
├──────────────────────────────┤
│                              │
└──────────────────────────────┘
```

Warehouse Manager
- _fileName : String
+ save() : void
+ save As (fileName : String) : void
+ load (fileName : String) : void
+ import File (textFile : String) : void
+ lookup Payments By Partner (partnerId : String) : Transaction []
+ lookup Product Batches Under Given Price (priceLimit : double) : Batch []
+ advance Date (days : int) : void
+ display Date () : int
+ available Balance () : double
+ accounting Balance () : double
+ register Partner (id : String, name : String, address : String) : void
+ get All Partners () : Partner []
+ get Partner (partnerId : String) : Partner
+ get Partner Sales Or Breakdowns (partnerId : String) : Transaction []
+ get Partner Acquisitions (partnerId : String) : Transaction []
+ toggle Product Notifications (partnerId : String, productId : String) : void
+ get All Products () : Product []
+ get All Batches () : Batch []
+ get Batches By Partner (partnerId : String) : Batch []
+ get Batches By Product (productId : String) : Batch []
+ receive Payment (transactionId : String) : void
+ register Acquisition Transaction (partnerId : String, productId : String, price : double, quantity : int) : void
+ create Simple Product (productId : String) : void
+ create Derived Product (productId : String, aggravating Factor : double, recipe Products : String [], recipeQuantities : int []) : void
+ register Breakdown Transaction (partnerId : String, productId : String, quantity : int) : void
+ register Sale Transaction (partnerId : String, deadline : int, productId : String, quantity : int) : void
+ get Transaction (transactionId : String) : Transaction

Warehouse
− availableBalance : double
+ lookupPaymentsByPartner(partnerId : string) : Transaction[]
+ lookupProductBatchesUnderGivenPrice(priceLimit : double) : Batch[]
+ advanceDate(days : int) : void
+ displayDate() : int
+ calculateAccountingBalance() : double
+ registerAcquisitionTransaction(partnerId : String, productId : String, price : double, quantity : int) : void
+ createSimpleProduct(productId : String) : void
+ createDerivedProduct(productId : String, aggravatingFactor : double, recipeProducts : String[], recipeQuantities : int[]) : void
+ registerBreakdownTransaction(partnerId : String, productId : String, quantity : int) : void
+ registerSaleTransaction(partnerId : String, deadline : int, productId : String, quantity : int) : void
− getNextTransactionId() : int
+ getAllBatches() : Batch[]
+ getAllBatchesUnderGivenPrice(priceLimit : double) : Batch[]
+ registerPartner(id : String, name : String, address : String) : Partner
+ subscribeToAllProducts(partner : Partner) : void
+ getAllPartners() : Partner[]
+ getPartner(partnerId : string) : Partner
+ getPartnerSalesOrBreakdowns(partnerId : String) : Transaction[]
+ getPartnerAcquisitions(partnerId : String) : Transaction[]
+ toggleProductNotifications(partnerId : String, ProductId : String) : void
+ getAllProducts() : Product[]
+ getBatchesByPartnerKey(partnerId : string) : Batch[]
+ getBatchesByProductKey(productId : string) : Batch[]
+ receivePayment(transactionId : string) : void
+ getTransaction(transactionId : string) : Transaction

Transaction

- id: int
- value: double
- quantity: double

+ baseValue(): double

SaleTransaction

- adjustedValue: Double
- paid: boolean

+ isPaid(): boolean
+ pay(): double
+ calculateCurrentPaymentPeriod(): PaymentPeriod

BreakdownTransaction

- paidValue: Double

+ paymentValue(): double

AcquisitionTransaction

+ unitPrice(): double

Date

- days: int

+ advanceDate(days: int): void
+ currentDate(): int
+ dateDiff(other: Date): int

Product

- id: String

+ getQuantityInBatches(): int
+ getTotalQuantity(): int
+ getBatchesForAcquisition(quantity: int): Batches[]
+ getCheapestPrice(): int
+ subscribe(partner: Partner): void
+ unsubscribe(partner: Partner): void
+ toggleSubscription(partner: Partner): void
+ isSubscribed(partner: Partner): boolean

BreakdownTransactionProduct

- <<final>> quantity: int
- <<final>> price: double

Notification

- productPrice: double

«enumeration»
PaymentPeriod

P1
P2
P3
P4

RecipeProduct

- quantity: int

«enumeration»
NotificationType

NEW
BARGAIN

«enumeration»
NotificationDeliveryMethod

APP

Batch

- quantity: int
- price: double

Recipe

- aggravatingFactor: double

Derived Product

+ getTotalQuantity(): int
+ getBuildableQuantity(): int
+ getPriceForBreakdown(): double
+ buildFromRecipe(quantity: int): void