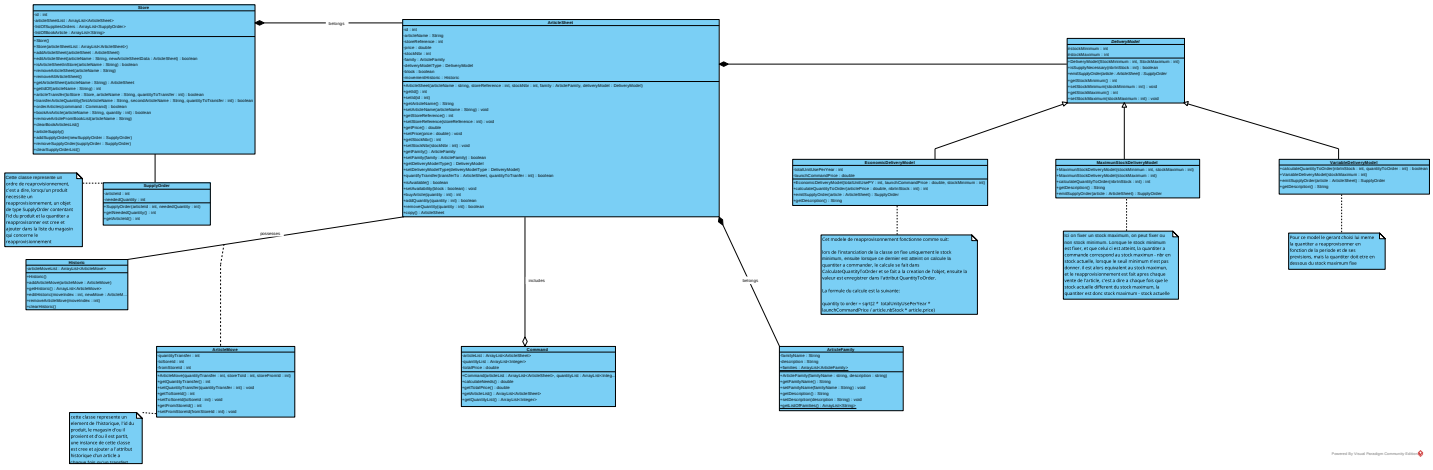
belongs
Store
id : int
articleSheetList : Array<ArticleSheet>
listOfSupplierOrders : Array<SupplyOrder>
listOfBlockArticle : Array<String>
+Store()
+Store(pArticleSheetList : Array<ArticleSheet>)
+addArticleSheet(pArticleSheet : ArticleSheet)
+isArticleSheetExist(pArticleName : String, newArticleSheetData : ArticleSheet) : boolean
+isArticleSheetExistCode(pArticleName : String) : boolean
+getArticleSheetByCode(pArticleName : String)
+removeArticleSheet()
+getArticleSheet(pArticleName : String) : ArticleSheet
+getId(pArticleName : String) : int
+articleTransferOnStore(pStore : Store, pArticleName : String, quantityToTransfer : int) : boolean
+transferArticleQuantityToArticleOnStore(pFirstArticleName : String, secondArticleName : String, quantityToTransfer : int) : boolean
+isArticleAlreadyUsedInCommand(pArticleName : String) : boolean
+addQuantityToArticle(pArticleName : String, quantity : int) : boolean
+removeArticleFromBlockedArticle(pArticleName : String)
+clearBlockArticleList()
+getStorMap()
+addSupplyOrder(newSupplyOrder : SupplyOrder)
+removeSupplyOrder(pSupplyOrder : SupplyOrder)
+clearSupplyOrderList()
Cette classe représente un ordre de réapprovisionnement, c'est a dire, lorsqu'un produit doit être réapprovisionnement, un objet de type SupplyOrder contenant l'id du produit et la quantité a réapprovisionner est crée et ajouter dans la liste du magasin qui concerne le réapprovisionnement
SupplyOrder
-ancestId : int
-neededQuantity : int
+SupplyOrder(pAncestId : int, neededQuantity : int)
+getNeededQuantity() : int
+getAncestId() : int
ArticleSheet
id : int
articleName : String
storeReference : int
price : double
inStockMin : int
family : ArticleFamily
deliveryModelType : DeliveryModel
block : boolean
blockReason : Reason
+ArticleSheet(pArticleName : String, storeReference : int, stockMin : int, family : ArticleFamily, deliveryModel : DeliveryModel)
+getId() : int
+setId(pId : int)
+getArticleName() : String
+setArticleName(pArticleName : String) : void
+getStoreReference() : int
+setStoreReference(pStoreReference : int) : void
+getPrice() : double
+setPrice(pPrice : double) : void
+getStockMin() : int
+setStockMin(pStockMin : int) : void
+getFamily() : ArticleFamily
+setFamily(pFamily : ArticleFamily) : boolean
+getDeliveryModelType() : DeliveryModel
+setDeliveryModelType(pDeliveryModelType : DeliveryModel)
+setArticleQuantityToArticle(pArticleSheet, quantityToTransfer : int) : boolean
+isBlocked() : boolean
+setBlockedBy(block : boolean) : void
+longArticle(pQuantity : int) : int
+addQuantity(pQuantity : int) : boolean
+removeQuantity(pQuantity : int) : boolean
+copy() : ArticleSheet
includes
DeliveryModel
-stockMinimum : int
-stockMaximum : int
+DeliveryModel(pStockMinimum : int, pStockMaximum : int)
+isSupplyNecessary(pArticleStock : int) : boolean
+setSupplyOrder(pArticle : ArticleSheet) : SupplyOrder
+getStockMinimum() : int
+setStockMinimum(pStockMinimum : int) : void
+getStockMaximum() : int
+setStockMaximum(pStockMaximum : int) : void
EconomicDeliveryModel
-totalUnitCostPerYear : int
-fixedCommandPrice : double
+EconomicDeliveryModel(totalUnitCostPY : int, fixedCommandPrice : double, stockMinimum : int)
+calculateQuantityToOrder(pArticlePrice : double, pStockStock : int) : int
+setSupplyOrder(pArticle : ArticleSheet) : SupplyOrder
+getDescription() : String
Cet modèle de réapprovisionnement fonctionne comme suit:
lors de l'instanciation de la classe on fixe uniquement le stock minimum, ensuite lorsque ce dernier est atteint on calcule la quantité a commander, le calcule se fait dans CalculateQuantityToOrder et se fait a la création de l'objet, ensuite la valeur est enregistrer dans l'attribut QuantityToOrder.
La formule du calcule est la suivante:
quantity to order = (qty)2 * totalUnitsUsePerYear * fixedCommandPrice / articleUnitCost * article price)
MaximumDeliveryModel
-stockMaximum : int
-stockMinimum : int
+MaximumDeliveryModel(pStockMaximum : int, pStockMinimum : int)
+calculateQuantityToOrder(pStockStock : int) : int
+setSupplyOrder(pArticle : ArticleSheet) : SupplyOrder
Ici on fixer un stock maximum, on peut fixer ou non stock minimum. Lorsque le stock minimum est fixer, et que celui ci est atteint, la quantité a commande correspond a (stock maximum - nbr en stock actuelle, lorsque le seuil minimum n'est pas atteint il est alors equivalent au stock maximum, et le réapprovisionnement est fait après chaque vente. Par exemple si le stock max est de 100 et le stock actuelle différent du stock maximum, la quantité est donc stock maximum - stock actuelle
VariableDeliveryModel
-calculateQuantityToOrder(pArticleQuantityToOrder : int) : boolean
+calculateDeliveryModelStockMinimum() : int
+setSupplyOrder(pArticle : ArticleSheet) : SupplyOrder
+getDescription() : String
Pour ce model le gérant choisi lui même la quantité a réapprovisionner en fonction de la période et de ses provisions, mais la quantité doit etre en dessous du stock maximum fixe
Reason
-articleStockList : Array<ArticleSheet>
+Reason()
+addArticleMoveOnArticleMove : ArticleMove)
+getHistory() : Array<ArticleMove>
+addArticleMove(pDate, int, pArticle : ArticleMove)
+addArticleMove(pArticleMove)
+clearHistory()
ArticleMove
-quantityTransfer : int
-isStorted : int
-fromStorId : int
+ArticleMove(pQuantityTransfer : int, pStorToId : int, pStorFromId : int)
+getQuantityTransfer() : int
+setQuantityTransfer(pQuantityTransfer : int) : void
+getStorToId() : int
+setToStorId(pStorToId : int) : void
+getFromStorId() : int
+setFromStorId(pFromStorId : int) : void
Cette classe représente un élément de l'historique, l'id du produit, le magasin d'où il provient et d'où il est parti, une instance de cette classe est crée et ajouter a l'attribut historique d'un article a
Command
-articleList : Array<ArticleSheet>
-quantityList : Array<Integer>
-totalPrice : double
+Command(pArticleList : Array<ArticleSheet>, quantityList : Array<String>)
+calculateNeeded() : double
+getTotalPrice() : double
+getArticle(n int) : Array<ArticleSheet>
+getQuantity(n int) : Array<Integer>
assigns
ArticleFamily
-familyName : String
-description : String
-families : Array<ArticleFamily>
+ArticleFamily(pFamilyName : String, description : String)
+getFamilyName() : String
+setFamilyName(pFamilyName : String) : void
+getDescription() : String
+setDescription(pDescription : String) : void
+getListOfFamilies() : Array<ArticleFamily>
Powered By Visual Paradigm Community Edition