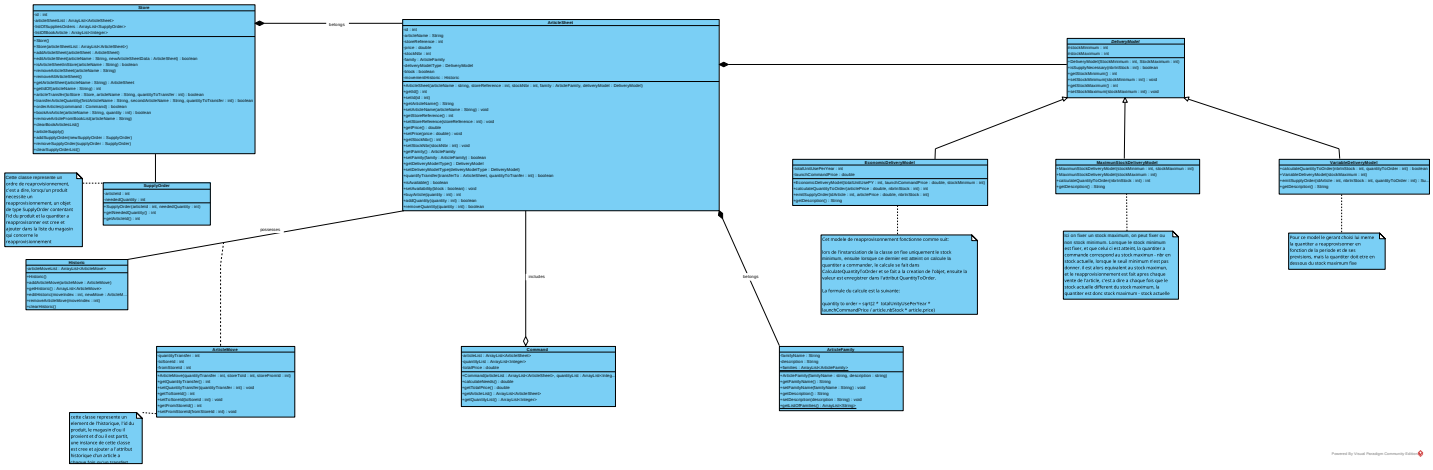
Store
id : int
articleSheetList : Array<ArticleSheet>
listOfSupplierOrders : Array<SupplyOrder>
listOfBlackArticle : Array<Integer>
+Store()
+Store(pArticleSheetList : Array<ArticleSheet>)
+addArticleSheet(pArticleSheet : ArticleSheet)
+addArticleSheet(pArticleName : String, newArticleSheetData : ArticleSheet) : boolean
+isArticleSheetExist(pArticleName : String) : boolean
+getArticleSheet(pArticleName : String)
+removeAllArticleSheet()
+getArticleSheet(pArticleName : String) : ArticleSheet
+getId(pArticleName : String) : int
+articleTransfer(pStore : Store, articleName : String, quantityToTransfer : int) : boolean
+transferArticleQuantity(firstArticleName : String, secondArticleName : String, quantityToTransfer : int) : boolean
+getArticleSheet(pCommand : Command) : boolean
+isArticleSheetAvailable(pArticleName : String, quantity : int) : boolean
+removeArticleFromBlackList(pArticleName : String)
+clearBlackArticleList()
+addSupplyOrder(newSupplyOrder : SupplyOrder)
+removeSupplyOrder(supplyOrder : SupplyOrder)
+clearSupplyOrderList()
belongs
ArticleSheet
id : int
articleName : String
storeReference : int
price : double
stockMax : int
family : ArticleFamily
deliveryModelType : DeliveryModel
block : boolean
+ArticleSheet(pArticleName : String, storeReference : int, stockMax : 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
+getStockMax() : int
+setStockMax(pStockMax : int) : void
+getFamily() : ArticleFamily
+setFamily(pFamily : ArticleFamily) : void
+getDeliveryModelType() : DeliveryModel
+setDeliveryModelType(pDeliveryModelType : DeliveryModel)
+isSufficientQuantity(pArticleSheet, pArticleSheet, quantityToTransfer : int) : boolean
+isBlocked() : boolean
+setAvailability(pBlock : boolean) : void
+longArticle(pquantity : int) : int
+addQuantity(pquantity : int) : boolean
+removeQuantity(pquantity : int) : boolean
includes
Command
articleList : Array<ArticleSheet>
quantityList : Array<Integer>
totalPrice : double
+Command(pArticleList : Array<ArticleSheet>, quantityList : Array<Integer>)
+calculateNeeds() : double
+getTotalPrice() : double
+getArticle(n:int) : Array<ArticleSheet>
+getQuantity(n:int) : Array<Integer>
DeliveryModel
stockMinimum : int
stockMaximum : int
+DeliveryModel(pStockMinimum : int, pStockMaximum : int)
+isSupplyNecessary(pArticleStock : int) : boolean
+getStockMinimum() : int
+setStockMinimum(pStockMinimum : int) : void
+getStockMaximum() : int
+setStockMaximum(pStockMaximum : int) : void
BusinessDeliveryModel
+calculateOrderPriceTotal() : double
+BusinessDeliveryModel(pStockMinimum : int, pStockMaximum : int)
+calculateQuantityToOrder(pArticlePrice : double, pArticleStock : int) : int
+isNeedSupplyOrder(pArticle : int, pArticlePrice : double, pArticleStock : int)
+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 * totalcostpUnitPerYear * EuROCommondPrice / articleStock * article price)
MaximumStockOrderModel
+MaximumStockOrderModel(pStockMinimum : int, pStockMaximum : int)
+calculateQuantityToOrder(pArticleStock : int) : int
+getDescription() : String
si 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 commander 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. Dans le cas ou on fixe un stock minimum, que le stock actuelle diffèrent du seuil maximum, la quantité est donc stock maximum - stock actuelle
VariableDeliveryModel
+calculateQuantityToOrder(pArticleStock : int) : int
+VariableDeliveryModel(pStockMinimum : int, pStockMaximum : int)
+isNeedSupplyOrder(pArticle : int, pArticleStock : int, quantityToOrder : int) : Boo
+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
SupplyOrder
amounted : int
sendedQuantity : int
+SupplyOrder(pamounted : int, sendedQuantity : int)
+getNeededQuantity() : int
+getAmounted() : int
Cette classe représente un ordre de réapprovisionnement, c'est a dire, lorsqu'un produit est en 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
possesses
Provisions
ArticleStockList : Array<ArticleSheet>
+Provisions()
+addArticleMove(pArticleMove : ArticleMove)
+getHistory() : Array<ArticleMove>
+addProvisions(pArticles, int, pDate) : Article
+addProvisions(pArticleprovisionArticle : int)
+clearHistory()
ArticleMove
quantityTransfer : int
sysTotalId : int
fromStoreId : int
+ArticleMove(pquantityTransfer : int, pSourceTotal : int, pStoreFromId : int)
+getQuantityTransfer() : int
+setQuantityTransfer(pquantityTransfer : int) : void
+getSysTotalId() : int
+setSysTotalId(pSysTotalId : int) : void
+getFromStoreId() : int
+setFromStoreId(pFromStoreId : int) : void
cette classe représente un élément de l'historique, l'id du produit, le magasine 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
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
+getArticleFamilies() : Array<ArticleFamily>
Powered By Visual Paradigm Community Edition