<<interface>> Portfolio

+ getPortfolioName():String
+ alculatePrice(Date):void
+ getStockShare(int StockshareId):StockShare
+addStockShare(SharePrice price, double num):void
+ getTotalCost():double
+getAllStocks:List<Double>

<<interface>> PortfolioAdvanced

+ getStartDate():date
+ setStartDate(Date):void
+ getEndDate():Date
+setEndDate(Date endDate):void
+ intPeridicity():int
+setPeridicty(int p):void
+ getBuyingCommision():double
+ setBuyingCommission(double d):void
+ setSellingCommsion(double d): void
+ getSellingCommsion():double
+ invest(Date date):void
+ investPeridocially(Date date, int perodic): void
+ getInvestmentAmount():double
+setInvestmentAmount(double d):void
+getRatios():List<Double>
+setratios(List<>Double s):void
+addStockSymbol(String Stocksymbol):void

<<interface>> StockShare

+ calculatePrice(Date date):double
+getshareCost():double
+getId():int

<<interface>> StockShareCommission

+ getBuyingCommsion():double
+setBuyingcommision(double):void
+ getSellingCommsion():double
+tSellingCommsion(double d):void

PortfolioImpl

+ portfolioName:Sting = ""
+ List<StocksShare> stocks:StocksShare

PortfolioAdvancedImpl

+ portfolioName:Sting = ""
+ List<StocksShareCommsion> stocks:StocksShare
+ startDate:Date
+endDate:Date

StockShareFree

+ stockSymbol:Stirng = ""
+ numberofshare:double
+ counting:int
+id:int

StockShareCommsionImpl

+ buyingCommsion:double
+sellingcommssion:double