StockMarketApp

Version 1.0

Generated by Doxygen 1.8.6

Tue Mar 8 2016 12:39:16

Code Written in C#

by

Dr Michael A. Idowu [1]
michade@hotmail.com
Dundee, United Kingdom

in response to the

Super Simple Stock Market challenge [2]

Contents

Chapter 1

Main Design

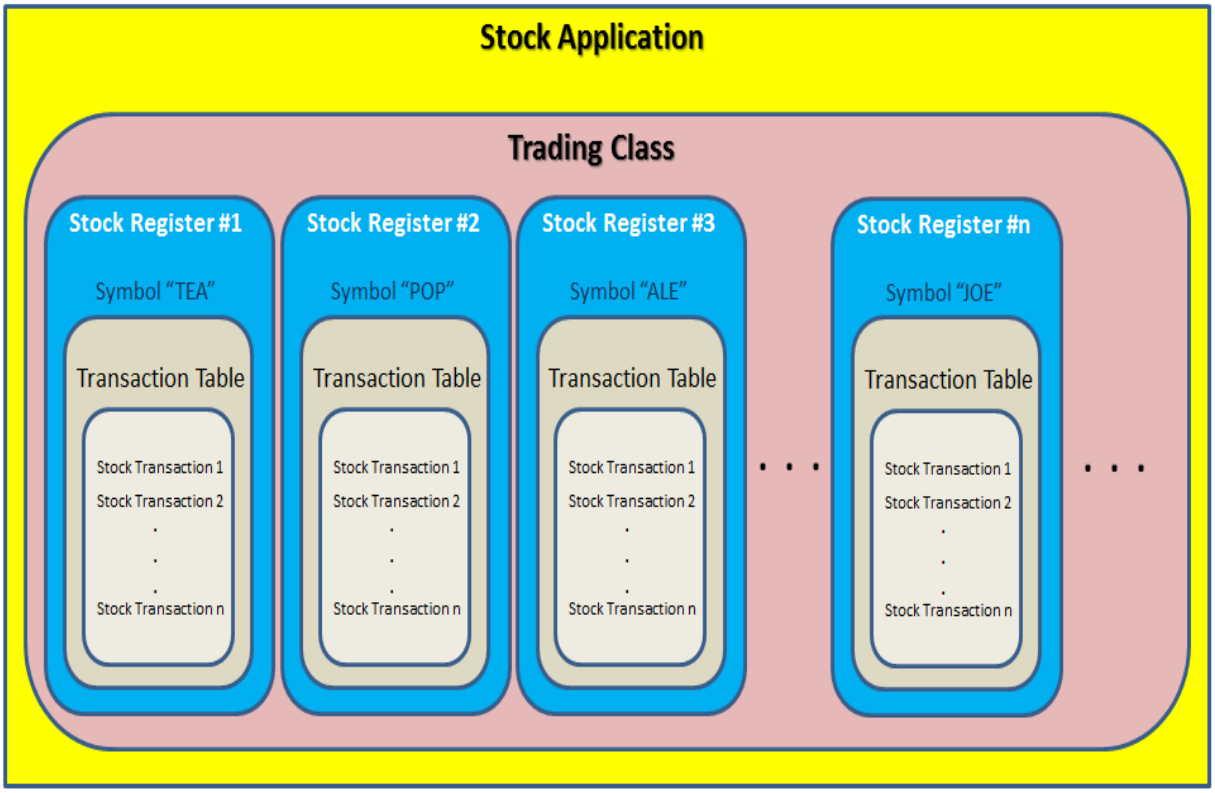

Figure 1.1: Super Simple Stock Market: Application Design for the Global Beverage Corporation Exchange

Chapter 2

Namespace Index

2.1 Packages

Here are the packages with brief descriptions (if available):

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 4

Namespace Documentation

4.1 Package StockApp

Classes

- class StockTransaction< TSymbolSource, TSymbolTarget, TTradeIndicator, TQty, TPrice, TTimeStamp >
- class StockRegister< TSymbol, TType, TLastDividend, TParValue, TFixedDividend >

  Define the register class;
  each stock is represented by this class.
  Note: StockRegister's relation to StockTransaction is 1-to-many
  xterised by the TransactionTable list variable
- class Trading
- class Application

Enumerations

- enum StockType { **Common**, **Preferred** }

  Define the two main stock types.
- enum TradeIndicator { **Buy**, **Sell** }

  Define the two main transaction indicators.
- enum StockQueryField {
  Symbol, **Type**, **LastDividend**, **FixedDividend**,
  ParValue }

  Define the main fields of the Global Beverage Corporation Exchange master table.

4.1.1 Detailed Description

Application class for capturing all trade and trading data: Trading Class - called from the Application.Main block

Entity-Relationship (ER) Model to be used: StockRegister<TSymbol,TType,TLastDividend,TParValue,TFixed-Dividend> Class - to - StockTransaction<TSymbolSource,TSymbolTarget,TTradeIndicator,TQty,TPrice,TTime-Stamp> Class

Description of the ER Model Relationship: Stock - to - Trade Transactions

Xteristic of the ER Model: 1-to-Many relationship

Chapter 5

Class Documentation

5.1 StockApp.Application Class Reference

5.1.1 Detailed Description

Definition at line 548 of file Program.cs.

The documentation for this class was generated from the following file:

- E:/StockApp/StockApp/Program.cs

5.2 StockApp.StockRegister< TSymbol, TType, TLastDividend, TParValue, TFixedDividend > Class Template Reference

Define the register class;

each stock is represented by this class.

Note: StockRegister's relation to StockTransaction is 1-to-many

xterised by the TransactionTable list variable

Public Member Functions

- StockRegister (TSymbol symb, TType typ, TLastDividend ldiv, TParValue parval, TFixedDividend fdiv)

  ... the constructor class
- void UseSampleTransactions ()

  sample transaction entries for illustrating trade recording
- void **ResetTransactionTable** ()
- List< StockTransaction< string, string, TradeIndicator, double, double, DateTime > > QueryTransactions ()
- void Echo ()

  for writing to the console
- double CalculateDividendYield (double price)

  ... method for calculating the dividend yield
- double CalculatePEratio (double price)

  ... method for calculating the P/E ratio as indicated

Public Attributes

- TSymbol **symbol**
- TType **type**
- TLastDividend **lastDividend**
- TParValue **parValue**
- TFixedDividend **fixedDividend**
- double **geometricMean**
- double **volumeWeightedPrice**
- List< StockTransaction< string, string, TradeIndicator, double, double, DateTime > > TransactionTable

  Create a list variable for linking up with the StockTransaction class.

5.2.1 Detailed Description

Define the register class;

each stock is represented by this class.

Note: StockRegister's relation to StockTransaction is 1-to-many

xterised by the TransactionTable list variable

Template Parameters

TSymbol	
TType	
TLastDividend	
TParValue	
TFixedDividend	

Definition at line 146 of file Program.cs.

5.2.2 Constructor & Destructor Documentation

5.2.2.1 StockApp.StockRegister< TSymbol, TType, TLastDividend, TParValue, TFixedDividend >.StockRegister (TSymbol *symb,* TType *typ,* TLastDividend *ldiv,* TParValue *parval,* TFixedDividend *fdiv*)

... the constructor class

for clearing (resetting) the TransactionTable list

Definition at line 165 of file Program.cs.

5.2.3 Member Function Documentation

5.2.3.1 void StockApp.StockRegister< TSymbol, TType, TLastDividend, TParValue, TFixedDividend >.Echo ( )

for writing to the console

Definition at line 228 of file Program.cs.

5.2.3.2 List<StockTransaction<string, string, TradeIndicator, double, double, DateTime> > StockApp.StockRegister< TSymbol, TType, TLastDividend, TParValue, TFixedDividend >.QueryTransactions ( )

Create a method to read from the TransactionTable based on the stock symbol, the returned object is a list of StockTransaction

Definition at line 205 of file Program.cs.

5.2.3.3 void StockApp.StockRegister< TSymbol, TType, TLastDividend, TParValue, TFixedDividend >.UseSampleTransactions ( )

sample transaction entries for illustrating trade recording

Create sample trade transactions to use ...

Definition at line 182 of file Program.cs.

The documentation for this class was generated from the following file:

- E:/StockApp/StockApp/Program.cs

5.3 StockApp.StockTransaction< TSymbolSource, TSymbolTarget, TTradeIndicator, TQty, TPrice, TTimeStamp > Class Template Reference

Public Member Functions

- StockTransaction (TSymbolSource source, TSymbolTarget target, TTradeIndicator indc, TQty qno, TPrice pr, TTimeStamp t)

  Define the constructor class.

Public Attributes

- TSymbolSource **symSource**
- TSymbolTarget **symTarget**
- TTradeIndicator **indicator**
- TQty **qty**
- TPrice **price**
- TTimeStamp **time**

5.3.1 Detailed Description

Define a generic class for capturing and recording all trade transactions

Definition at line 49 of file Program.cs.

The documentation for this class was generated from the following file:

- E:/StockApp/StockApp/Program.cs

5.4 StockApp.Trading Class Reference

Public Member Functions

- Trading (StockRegister< string, StockType, double, double, double >[] stTable)

  Constructor definition 1.
- Trading ()

  Constructor definition 2.

- List< StockRegister< string, StockType, double, double, double > > **FetchStockFromExchange** (string symbol)
- List< StockRegister< string, StockType, double, double, double > > QuerySymbol (string qSymb)

  useful for fetching a stock record from the master register (in case multiple records exist)
- double CalcGeometricMean (string qSymb)

  ...calculating the geometric mean
- double CalcVolWeightedStockPrice (string qSymb, int NoOfMins)

  ... calculating the volume weighted stock price
- void GMAndWPForAllStocks (int NoOfMins)

  ... calculating the geometric mean and volume weighted stock price for all stocks
- int FindIndex (string qSymb)

  ... find the record index from the master table
- void QueryStockType (StockType qType)

  ... reporting all records associated with a specific StockType
- void QueryLastDividend (double qlastDividend)

  ... reporting all records associated with a specific LastDividend value
- void QueryFixedDividend (double qfixedDividend)

  ... reporting all records associated with a specific FixedDividend value
- void QueryParValue (double qparValue)

  ... reporting all records associated with a specific ParValue
- void CalcDividendAndPEratio (string symbol, string pr)

  ... handling how to calculate dividend yield and P/E ratio
- void CalcDividendAndPEratio (string symbol, double dprice)

  ... handling how to calculate dividend yield and P/E ratio
- void InputPrice (string symbol, double dprice)

  ... inputing stock symbol and price to use
- void Transactions (string symbSource, string symbTarget, TradeIndicator indc, double qty, double price)

  ... handling trade transactions - query

Public Attributes

- StockRegister< string, StockType, double, double, double >[] **stockTable**

5.4.1 Detailed Description

Creating the main class for controlling and regulating the stock trade/trading application logic

Encapsulates an array of the StockRegister; each StockRegister uniquely represents a stock

Definition at line 297 of file Program.cs.

5.4.2 Constructor & Destructor Documentation

5.4.2.1 StockApp.Trading.Trading (StockRegister< string, StockType, double, double, double >[] *stTable*)

Constructor definition 1.

Create stock master table

Definition at line 302 of file Program.cs.

5.4.3 Member Function Documentation

5.4.3.1 double StockApp.Trading.CalcGeometricMean (string *qSymb*)

...calculating the geometric mean

Console.Write(double.TryParse(tt.qty.ToString(), out d).ToString());

Definition at line 341 of file Program.cs.

5.4.3.2 void StockApp.Trading.GMAndWPForAllStocks (int *NoOfMins*)

... calculating the geometric mean and volume weighted stock price for all stocks

calculating geometric mean and volume weighted price for all stocks

print

negative numbers depict "undefined"

Definition at line 389 of file Program.cs.

5.4.3.3 List<StockRegister<string, StockType, double, double, double> > StockApp.Trading.QuerySymbol (string *qSymb*)

useful for fetching a stock record from the master register (in case multiple records exist)

Parameters

qSymb	

Returns

Definition at line 324 of file Program.cs.

The documentation for this class was generated from the following file:

- E:/StockApp/StockApp/Program.cs

Index