Table of Contents

Overview

Technologies & Framework

Design

Flow Diagram

Sample Run

Overview

Stock Market application is developed to implement the sample test program given by JMPC (Grant Forest). Since the developer is technical person Java developer, he has assumed the finance related terminology and formulas which are not very clear in program statement.

Technologies & Framework

Below is the list of Technologies used in Application

* Java 1.6
* Spring 4.2.5
* Junit 4.12
* mockito 1.9.5
* log4j 1.2.17

Design Patterns implemented in Application

* Factory Design Pattern
* Strategy Design Pattern

Design

1. Its 3 layered design. Controller 🡪 Service 🡪 DAO
2. At the startup the Stock data given in problem statement is populated and saved in DAO
3. Since there are no UI forms, controller is responsible for capturing input and sends to service.
4. Service refers to Strategy Factory to instantiate one of Strategy based on input.
5. DAO maintains the data and provides the APIs to access it.
6. Other classes in application are
   1. App.java : Starting point of application and contains main() method
   2. AppConfig.java : Configuration class for Spring beans
   3. Test Classes : junit test classes
   4. Persistence classes : To hold the data
   5. Dto : To transfer data across layer
   6. Log4j.properties : Logging configuration

Flow Diagram

Controller Service DAO Memory

StockMarketDao

populateStock() getStock() recordTrade() getAllTradesForStock()

Add & maintain Stock & Trade data.

StockMarketContoller

captureInpute()

Responsible for capturing user input and send to Service. Perform validations and created input object containing user input

StockMarketService

performOperation()

Calls Strategy Factory Calls performOperation() on Strategy returned by factory

StockMap

TradeMap

OperationStrategyFactory

getStrategy()

This factory returns the appropriate Strategy for Input

Strategy

performOperation()

Strategy Design Pattern Interface

IndexCalculation Strategy

Contains algorithm to calculate Index

DividendCalculationStrategy

Contains algorithm to calculate Dividend

TradeRecorder Strategy

Contains algorithm to calculate PE ratio

PeRationCalculation Strategy

Contains algorithm to calculate PE ratio

VolumeWeighted StockPriceStrategy

Contains algorithm to calculate Vol. Weighted Stock price Strategy

Sample Run

**Below are the assumptions made while developing application** :-

* Stock map will be populated at time of application startup. New stock can’t be added by users.
* User need to enter the Stock name from the list of stocks specified in problem statement. Below is the list of stocks.

1. TEA
2. POP
3. ALE
4. GIN
5. JOE

**Below are the guidelines to run the application with sample data** :-

* Copy the ‘Stock Market’ application on the local drive.
* Run ‘mvn clean install’ command to compile the project.
* On successful compilation Stock Market🡪target folder will be created.
* Go inside the folder and run command ‘java -cp target/stockmarket-1.0.jar com.jpmc.app.App’ to run the application
* You will be prompted to enter the number from 1 to 6 and corresponding operation which will be performed
* Below is the sample Run screenshot for number 1 which corresponds to operation ‘Calculate the Dividend Yield’
* Logs will be generated at logging/Stockmarket.out on the same drive where the application is copied. Path can be changed in log4j.properties file.

