Problem Statement

Traders want a system to show the real-time value of her portfolio which consist of three types of products:

1. Common stocks.
2. European Call options on common stocks.
3. European Put options on common stocks.

Requirements

Your task is to design and implement a system in Java that:

1. Get the positions from a mock CSV position file (consisting of tickers and number of shares/contracts of tickers in the portfolio)
2. Get the security definitions from an embedded database.
    - Design a schema with small embedded database (H2 or SQLite) to store the security definitions (three supported types: Stock, Call, Put)
    - Each security in this database will have a ticker (identifier) and will have some static (e.g. strike, maturity)
3. Implement a **mock** market data provider that publishes stock prices.
    - The stock prices move according to either
        - Random pricing
        - or Preferable a discrete time geometric Brownian motion (see appendix) randomly between 0.5 – 2 seconds.
4. Calculate the real time option price with the underlying price
5. Publishes following details in real-time:
    - Each position's market value.
    - Total portfolio's NAV.
6. Implement a portfolio result subscriber.
    - Listener the above result
    - print it into console (pretty print)

Important Notes

- Not expected to support real stocks and options
- Not expected to consume real market data
- Not expected to design real-time communication channels via external messaging system / broker.
    - You can assume that the market data publisher described above run as a separate thread in the same program.
- Limit your third party libraries or components to Spring, Guava, Protobuf, Junit, Cucumber, H2 / SQLite.
    - Please do not distribute the binaries of these dependencies –just mention the name and version and we will source them ourselves.
    - If you need additional 3rd party libraries then feel free to check with us.
- Provide a README
- Project should be built with Gradle and JDK 1.8
- Zip the source code, protected with password

Sample

This is just for reference. If you have a better and **clear** input / output solution, you don't need to follow these samples.

Sample CSV position file

```
symbol,positionSize
AAPL,1000
AAPL-OCT-2020-110-C,-20000
AAPL-OCT-2020-110-P,20000
TELSA,-500
TELSA-NOV-2020-400-C,10000
TELSA-DEC-2020-400-P,-10000
```

Sample Real-time Output

```
## 1 Market Data Update
APPL change to 110.00
TELSA change to 450.00

## Portfolio
symbol                      price               qty              value
AAPL                       110.00             1,000.00         110,000.00
AAPL-OCT-2020-110-C          5.55           -20,000.00        -111,000.00
AAPL-OCT-2020-110-P          0.55            20,000.00          11,000.00
TELSA                      450.00              -500.00        -225,000.00
TELSA-NOV-2020-400-C        27.25            10,000.00         272,500.00
TELSA-DEC-2020-400-P         6.35           -10,000.00         -63,500.00

#Total portfolio                                               -6,000.00

## 2 Market Data Update
APPL change to 109.00

## Portfolio
symbol                      price               qty              value
AAPL                       109.00             1,000.00         109,000.00
AAPL-OCT-2020-110-C          5.50           -20,000.00        -110,000.00
AAPL-OCT-2020-110-P          0.56            20,000.00          11,200.00
TELSA                      450.00              -500.00        -225,000.00
TELSA-NOV-2020-400-C        27.25            10,000.00         272,500.00
TELSA-DEC-2020-400-P         6.35           -10,000.00         -63,500.00

#Total portfolio                                               -5,800.00


## 3 Market Data Update
TELSA change to 451.00


## Portfolio
symbol                      price               qty              value
AAPL                       109.00             1,000.00         109,000.00
AAPL-OCT-2020-110-C          5.50           -20,000.00        -110,000.00
AAPL-OCT-2020-110-P          0.56            20,000.00          11,200.00
TELSA                      451.00              -500.00        -225,500.00
TELSA-NOV-2020-400-C        27.30            10,000.00         273,000.00
TELSA-DEC-2020-400-P         6.30           -10,000.00         -63,000.00

#Total portfolio                                               -5,300.00
```

Appendix

You are expected to understand all the terms used in the material that follows.

Discrete Time Geometric Brownian motion for stock prices

In this model, we assume that if we know the price of a stock is S then after Δt number of seconds it will be $S+\Delta S$ where,

$$\frac{\Delta S}{S} = \mu\left(\frac{\Delta t}{7257600}\right) + \sigma\epsilon\sqrt{\frac{\Delta t}{7257600}}$$

Here, μ is the expected return on the stock (assume it to be one of the static fields for a stock and assign every stock a unique value between 0 and 1), σ is the annualized standard deviation of the stock (assume it to be one of the static fields for a stock and assign every stock a unique value between 0 and 1). ε is a random variable that is drawn from a standardized normal distribution every time this formula is invoked.

The price of a stock can never be less than 0. You can start your trading day with a start of the day price of your choice for every stock you support in your system.

Market Value

The market value of a long position in a common stock is defined as the number of shares held times the stock's price. If the position is short, then this value is multiplied by -1. The market value of a long position in an option (all types --put or call, American or European) is defined as the number of contracts held times the option's theoretical price. If the position is short, then this value is multiplied by -1.

The portfolio's NAV is the arithmetic sum of all positions' market values.

European Option Pricing Formula

Each option has a fixed time to maturity (let's say t years) and a fixed strike price (K). You can assume that the risk free interest rate in the market is constant at r. For this exercise, assume that it is 2% per annum. Given a stock with current price S and stock price's annualized standard deviation (volatility) σ, a European call option's price will be c and a European put option's price will be p:

$$c = SN(d_1) - Ke^{-rt}N(d_2)$$
$$p = Ke^{-rt}N(-d_2) - SN(-d_1)$$

Where

$$d_1 = \frac{\ln\left(\frac{S}{K}\right) + \left(r + \frac{\sigma^2}{2}\right)t}{\sigma\sqrt{t}}$$

$$d_2 = d_1 - \sigma\sqrt{t}$$

The function $N(x)$ refers to the cumulative probability that a variable that is distributed according to a standardized normal distribution has a value less than x. $ln(x)$ refers to the natural logarithm function and e is the Euler's number (base of the natural logarithm function).