**Problem:**

For this task, the candidate is expected to demonstrate good software design and implementation within the context of low-latency (high throughput) systems. The solution provided is expected to be of reasonable quality and take consideration of performance. Wherever the problem is unclear, the candidate should make (and state) any assumptions he or she thinks appropriate within the application context. Finally, although there is no time limit, this task is expected to take no longer than an hour.

The task:

Design and write an implementation of the Calculator interface below, in Java, so on the applyMarketUpdate method being called the TwoWayPrice returned is the VWAP two-way price for the instrument of the MarketUpdate. Each instrument can receive a two way price update from one of 50 markets, and in calculating the VWAP for the instrument the calculator should consider the most recent price update for each market (received so far)

The VWAP two-way price for an instrument is defined as:

Bid = Sum(Market Bid Price \* Market Bid Amount)/ Sum(Market Bid Amount)

Offer = Sum(Market Offer Price \* Market Offer Amount)/ Sum(Market Offer Amount)

It should be assumed the process calling Calculator.applyMarketUpdate is single threaded.

public interface Calculator {

TwoWayPrice applyMarketUpdate(final MarketUpdate twoWayMarketPrice);

}

public interface MarketUpdate {

Market getMarket();

TwoWayPrice getTwoWayPrice();

}

public interface TwoWayPrice {

Instrument getInstrument();

State getState();

double getBidPrice();

double getOfferAmount();

double getOfferPrice();

double getBidAmount();

}

public enum Instrument {

INSTRUMENT0,

…

INSTRUMENT19

}

public enum Market {

MARKET0,

…

MARKET49

}

public enum Side {

BID,

OFFER

}

public enum State {

FIRM,

INDICATIVE

}

**Solution:**

<https://github.com/senthamizh1976/vwapcalculator>

Thanks for giving me an opportunity to take part in problem solving interview. I have reviewed the problem an provided solution. I have uploaded the solution to the above Github repository server.

1. Test the following test cases to see the results.

/mycompany/test/CalculatorImplTest.java