Pragma Securities Coding Assignment

I divided the problem into 4 parts:
Book – To store the given market data for Hotspot. I have chosen array to store the price and quantity, and separated them into 2, ask and bid book for simplicity.
In usual cases the market data is constantly changing, but here since I'm working with a fixed set, I went for a basic data structure.
Client – Has a function which send order to the engine. I also have the main function in the client class, and creating 2 client object for buy and sell.
I have 3 basic inputs to send an order, the new message tag 3, side (sell/buy) and quantity. Since I'm concentrating only on these 3 inputs, i have directly used variables to handle them instead of Fix messages.
Engine – Based on the inputs, the huge quantity is sliced down to 1,3 or 5 million and we send multiple fill orders to exchange to execute the full quantity, since we are allowed to take these quotes in increments of 1,3,5 million dollars.
Eg: EUR/USD, Ask, FX_HOTSPOT_FIRM,1.15184,6000000 For this example, first order is sent with quantity 5 million and second order for the remaining 1 million.
Engine: Assuming that the engine always send a fill successfully, ignored all other reject, partially filled options.
Since the price in the book is not changing, and only "A buy order for 20 million dollars as the executed quantity" without a price limit was mentioned I have ignore price as an input, however since they are ordered from ascending to descending, we get the best price in this case.

Execution Instruction:

Run: java -jar PragmaAssignment.jar

Attachments: PragmaAssignment.jar – executable jar, TradingEngine.zip has all the .java files

OUTPUT:

```
Execution log:
Order Slice: 1
Qty filled: 1000000.0
Qty remaining: 1.9E7
Order Slice: 2
Qty filled: 1000000.0
Qty remaining: 1.8E7
Order Slice: 3
Qty filled: 1000000.0
Qty remaining: 1.7E7
Order Slice: 4
Qty filled: 1000000.0
Qty remaining: 1.6E7
Order Slice: 5
Qty filled: 3000000.0
Qty remaining: 1.3E7
```

```
Order Slice: 6
Qty filled: 1000000.0
Qty remaining: 1.2E7
Order Slice: 7
Qty filled: 1000000.0
Qty remaining: 1.1E7
Order Slice: 8
Qty filled: 1000000.0
Qty remaining: 1.0E7
Order Slice: 9
Qty filled: 1000000.0
Qty remaining: 9000000.0
Order Slice: 10
Qty filled: 5000000.0
Qty remaining: 4000000.0
Order Slice: 11
Qty filled: 1000000.0
Qty remaining: 3000000.0
Order Slice: 12
Qty filled: 1000000.0
Qty remaining: 2000000.0
Order Slice: 13
Qty filled: 1000000.0
Qty remaining: 1000000.0
Order Slice: 14
Qty filled: 1000000.0
Qty remaining: 0.0
Order Executed Successfully
Execution log:
Order Slice: 1
Qty filled: 1000000.0
Qty remaining: 1.1E7
Order Slice: 2
Qty filled: 1000000.0
Qty remaining: 1.0E7
Order Slice: 3
Qty filled: 1000000.0
Qty remaining: 9000000.0
Order Slice: 4
Qty filled: 1000000.0
Qty remaining: 8000000.0
Order Slice: 5
Qty filled: 1000000.0
Qty remaining: 7000000.0
Order Slice: 6
Qty filled: 5000000.0
Qty remaining: 2000000.0
Order Slice: 7
Qty filled: 1000000.0
Qty remaining: 1000000.0
Order Slice: 8
Qty filled: 1000000.0
Qty remaining: 0.0
Order Executed Successfully
```