**Throttled Market Data Stream**

Market data stream is a sequence of small market data updates for different instruments. Typical rate is up to 1000 updates per second per instrument. Each update contains only updated fields and reflects the recent state of the instrument on the market.

There might be several market data providers, which typically provides different subsets of market data fields. For simplicity, market data update can be treated as a dictionary <byte FieldId, long FieldValue>. Please review the skeleton in attached solution.

It does make sense for remote clients (GUI applications) to consume not more than 4-5 updates per second per instrument, so throttling mechanism should be implemented to avoid unnecessary updates and maintain acceptable for GUI application update rate.

For example, several market data updates in a row might be received:

AAPL\_1 [1: 10, 4: 200, 12: 187]

AAPL\_2 [1: 12, 4: 210]

AAPL\_1 [ 12: 189]

AAPL\_1 [ 2: 24]

AAPL\_2 [5: 120]

These updates should be aggregated as follows:

AAPL\_1 [1: 10, 4: 200, 12: 189, 2: 24]

AAPL\_2 [1: 12, 4: 210, 5: 120]

Your task:

1. Implement throttling of the market data stream for GUI client to receive only aggregated updates with approximate rate 4-5 messages per second
2. Implement missed methods and make sure ThrottledMarketDataStream class is thread-safe
3. Add unit tests to cover the implemented logic

Please pay attention to performance and try to avoid unnecessary memory allocations.

Feel free to fix issues and change data structures in sample solution if necessary.