Design Consideration:

1. I tried to design the code simple and easy to understand.

2. I have provided 100% unit test coverage for the code, I put strong emphasis on unit testing.

3. Each Classes are loosely coupled to other application component through principal of programming to interface.

4. Single purposed classes, Each class purpose is well defined and focused.

**Sequence Diagram:**

------> WriteCommand

Client ------------> MessageProcessor ------------>CommandFactory ------> AppendCommand

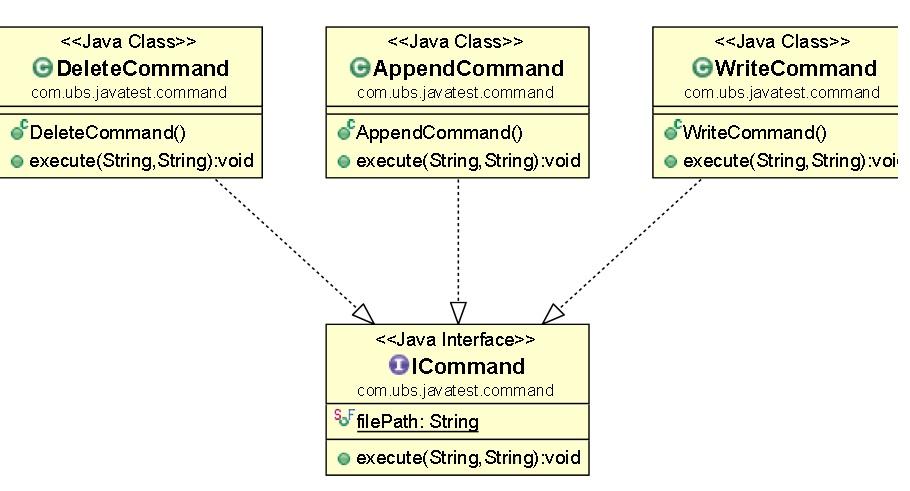
------> DeleteCommand

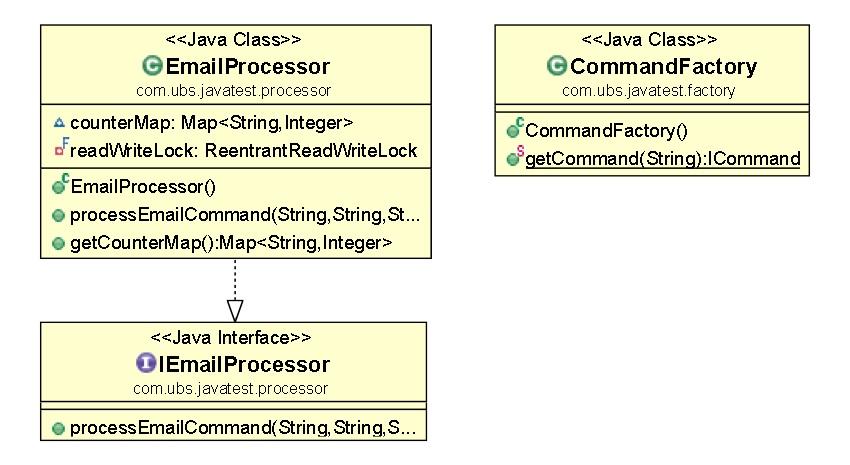
Write Command: Write to new file or overwrite the existing content

Append Command: append to same file if file already exists

DeleteCommand: delete the file

**Class Diagram:**





**\* In high volume application design consideration would be different:**

1. Incoming email should be process by Filter Thread ,which basically identifies the message type based on subject of message and push the messages to corresponding process queue.

2. Process queue should be implemented using BlockingQueue for each type of Email (Write,Append and Delete)

3. Each Queue need to have Thread Pool attached to it to process incoming messages in their respective process queue.