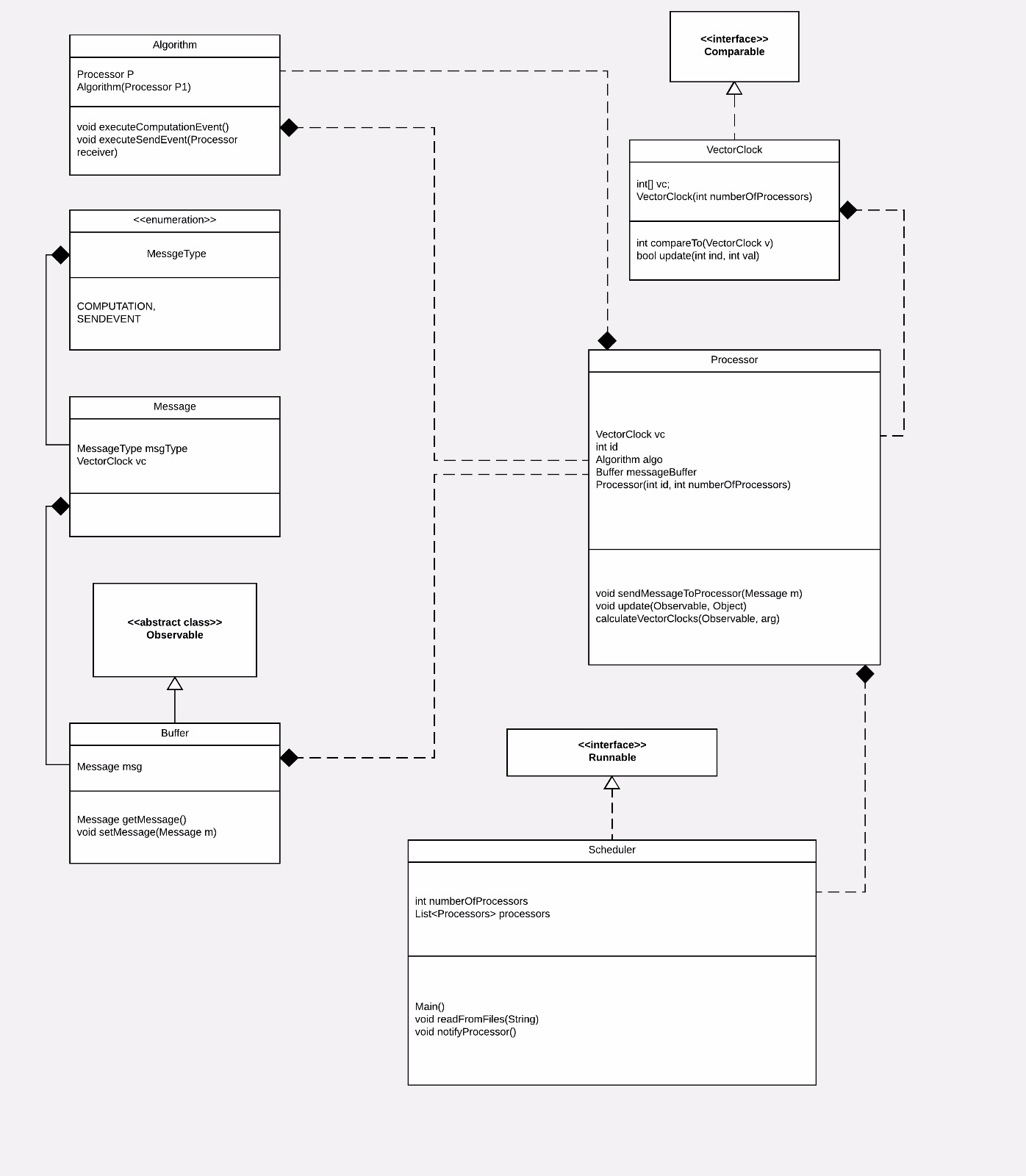
**LOGIC INTUITION**

1) The algorithm start from the scheduler which takes input as number of procesors from the user and filename associated to each Processor.  
2) It then forks multiple threads which monitors and reads the file associated to each Processor for events and then calls the Processor which handles the event based on the event type.  
3) Rest of the flow and interactions should be explanatory from the sequence diagram and class diagram.

**CLASS DIAGRAM**



**How to work around with Code**

Create an execution Plan for a Processor in a file and save it.  
Currently there are only two type of events supported.  
1) COMPUTATION EVENT: For a computation event just write C (in CAPITAL) for that Processor  
2) SEND EVENT: For a send event write S (in CAPITAL). For example in order to simulate a send event to Processor 2, write S 2.

Once you have created execution plans for each processor, run the main.java file from the src folder.  
On execution it will ask you to input the number of processors and the name of the files containing their execution plans.

Since all the processors are simulated to run individually in their own threads, the execution timeline for each Processor may vary.   
The store output will show the vector clocks as they were updated for each processor and it provides intuition of the order in which the execution plan at each processor may occur.

**SAMPLE OUTPUT**

