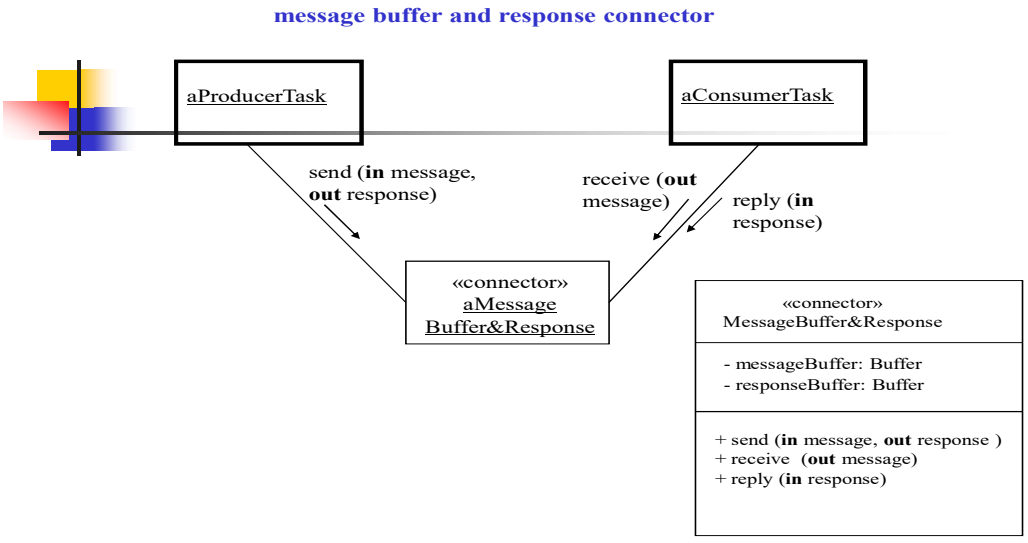
Spring 2021-CS2365 – Multithread Homework

You are requested to implement a "message buffer and response connector" class using Java programming language. The following figure depicts a message buffer and response connector class, followed by detailed operations specifications. The message buffer and response connector class has three operations, send(), receive(), and reply().

Your program should create two separate threads for testing your implementation, a producer thread, and a consumer thread. The producer thread sends a message having a structure (string, integer) – e.g., (add, 3) or (multiply, 7) - to a consumer thread via a message buffer and response connector. The consumer thread encapsulates a SimpleCalculation class that has two operations, add() and "multiply(). **You should implement the SimpleCalculation class as well**. When the consumer thread receives a message from the connector, it extracts the message and then calls one of the operations on the SimpleCalculation class, depending on the message. For example, the consumer thread calls the add() operation if it receives a message like (add, 3), while it calls the multiply() operation if it gets a message like (multiply, 3). Add() operation in the SimpleCalculation adds 10 to the integer in a message from the producer thread and returns the result. The multiply() operation multiplies the integer by ten and returns the result.

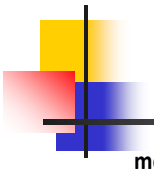
The producer thread should print out the messages before it sends the messages to the consumer thread. The messages are (add, 5), (multiply, 9), (multiply, 4), (add, 3), (add, 10), (add, 30), (multiply, 7) sent to the consumer thread. Also, the producer thread should print out the results returned from the consumer thread.

Please make a zip file containing all source codes (.java files) for this assignment and submit it to the blackboard.



26

Message buffer & response connector



```
monitor MessageBuffer&Response
 -- Encapsulates a message buffer that holds at most one message
 -- and a response buffer that holds at most one response.
 -- Monitor operations are executed mutually exclusively.
private messageBufferFull : Boolean = false;
private responseBufferFull : Boolean = false;
public send (in message, out response)
  place message in buffer;
  messageBufferFull := true;
  notify;
  while responseBufferFull = false do wait;
  remove response from response buffer;
  responseBufferFull := false;
end send;

public receive (out message)
  while messageBufferFull = false do wait;
  remove message from buffer;
  messageBufferFull := false;
end receive;

public reply (in response)
  Place response in response buffer;
  responseBufferFull := true;
        notify;
end reply;
end MessageBuffer&Response;
```