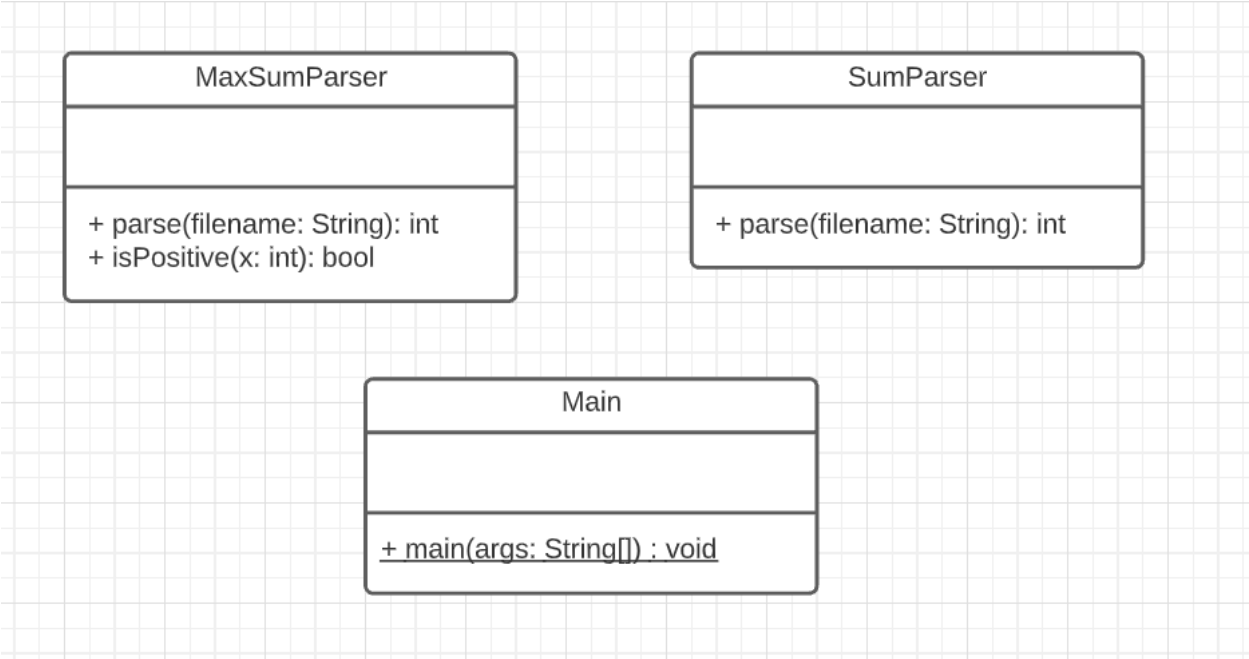
Question 1:

Use the following UML diagram to create a file parsing program.



Both MaxSumParser and SumParse will take in files that contain one integer value per line. The MaxSumParser should read in that file and return the maximum possible number that you could get by summing a subset of these numbers (in other words it only sums the positive numbers). The SumParser, on the other hand, should sum every number.

The main application should simply prompt the user to enter a filename and print out both the sum and maximum sum. If the file does not exist or cannot be read, the program should exist gracefully and inform the user that something went wrong.

Question 2:

You are trying to create an application that can perform several operations on a group of words. Those operations are getting the longest word (long), shortest word (short), the word that appears the first alphabetically by first letter only (az), and the word that appears last alphabetically by first letter only (za).

Your program should prompt the user for input. The user can then enter a command followed by all the words they wish for the command to be performed on. Use long, short, az, and za as the keywords for your commands.

Ex. >> long Hello Goodbye Nice to see you
>> Goodbye
Your word list here is [Hello, Goodbye, Nice, to, see, you] and the longest word is goodbye, so that will be printed.

All commands should have an execute() method that performs it's appropriate tasks. All commands should also have a parse() method that parses the list of words the user entered.

The general flow of your main should be as follows:

1. Find out which command the user entered
2. Create an instance of that command
3. Parse the words
4. Execute the command

If the user enters "quit" the program should terminate. However, since quit does not need to be parsed you should not treat it as a command.