**Task 2 Using Client-Server**

**Design:**

There are 4 major classes which have been created to simulate the random testing tool.

|  |  |
| --- | --- |
| Client | Simulates the client |
| Generator | Generates a multitude of random strings to be inputted into the program |
| ClientServer | Main |
| Server | Simulates the server |

|  |  |
| --- | --- |
| **tests.TestClient** | Junit test cases |

**Instructions:**

Please compile all Java classes.

javac \*.java

Run jUnit tests on the individual methods in **TestClient**

**Test Cases:**

|  |  |
| --- | --- |
| **Test name:** | testOutputIsProvidedWithValidInput() |
| **Test objective:** | To check if an output is provided with a valid input |
| **Test input:** | One random string of 50 characters in length (including spaces) |
| **Expected output:** | At least one line of random text is written to the output file. |
| **Result:** | Pass |

|  |  |
| --- | --- |
| **Test name:** | testNoWordInput() |
| **Test objective:** | To check if an output is provided with a blank input |
| **Test input:** | One random string of 0 characters in length (empty) |
| **Expected output:** | Nothing is written to the output file |
| **Result:** | Pass |

|  |  |
| --- | --- |
| **Test name:** | testOneWordInput() |
| **Test objective:** | To check if an output is provided with a 50 char random input |
| **Test input:** | One random string of 50 characters in length (including spaces) |
| **Expected output:** | At least one line of random text is written to the output file. |
| **Result:** | Pass |

|  |  |
| --- | --- |
| **Test name:** | testPunctuationMarksAreAllowed() |
| **Test objective:** | To check if an output is provided with a 50 char random input with punctuation marks |
| **Test input:** | One random string of 50 characters in length (including commas) |
| **Expected output:** | At least one line of random text is written to the output file. |
| **Result:** | Pass |

|  |  |
| --- | --- |
| **Test name:** | testRepeatingWords() |
| **Test objective:** | To check if an output is correctly sorted with a 50 char random input |
| **Test input:** | - Random string of |
| **Expected output:** | Both outputs are written to the output file in lexographical order |
| **Result:** | Fail |
| **Fix:** | The output did not accept a string where:  Input = input + “\n” + input;  Solution  Input = input + “ ” + input; |

|  |  |
| --- | --- |
| **Test name:** | testSymbolsUsedInBookTitles() |
| **Test objective:** | To check if an output is provided with a 50 char random input with a book title and some random chars |
| **Test input:** | Star Wars[]!$%^&\*() |
| **Expected output:** | The line of random text is written to the output file. |
| **Result:** | Pass |

|  |  |
| --- | --- |
| **Test name:** | testSingleCharacter() |
| **Test objective:** | To check if an output is provided with a 1 char random input |
| **Test input:** | One char random input |
| **Expected output:** | The line of random text is written to the output file. |
| **Result:** | Pass |

|  |  |
| --- | --- |
| **Test name:** | testDuplicateTitles() |
| **Test objective:** | To check if an output is provided with a 20 char random input with a book title and some random chars |
| **Test input:** | Input = random string with 20 chars  Input  Random string with 20 chars  Input |
| **Expected output:** | The duplicates are written to the output file in the correct order |
| **Result:** | Pass |

***Task 2 was completed solely by Chris Chapman c3256955***