File Processing Application

Task description:

Develop a command-line application in programming language of preference (Java or C++) that allows the user to manipulate text files with number content. The final solution should provide the following:

1. Handling of *.txt* files that contain lines with numbers. Each number is presented as a sequence of digits (0-9) and must not start with 0. The numbers in each line are separated by spaces or tabs. Each line must start with a number, but not space or tab.

   Example of the file content:
   112312321525 12312312 5234234234   231321321312 123123 2131353453435345345
   213123523520234234 2342340005320 3240230042340230803240 3240909
   9897823423975239

2. Command-line application that asks the user for a file (*txt* format).
3. After the user specified the path to the file the application should provide the following options:
   a. Validate the file contents – check if the file contents are valid, based on the restrictions in 1. and present to the user the result of the validation (the format is up to you, for instance: *"line <number>, character <number> is not allowed"* or *"number at line <number> starts with 0"*).
   b. Switch entire line from the file with entire another line. The user should specify two line indexes and the application should switch these lines.
   Example user input: <first_line_index> <second_line_index>
   c. Switch number at specific index in one line with a number with specific index from another line. The user should provide four indexes and the application should switch the number from the first line with the number from the second line.
   Example user input:
   <first_line_index> <first_line_number_index> <second_line_index><second_line_number_index>
   d. Reverse the numbers in a line. The first number should become last, etc.
   Example: the line *21 234 909 5* should be transformed to *5 909 234 21*.
   Example user input: <line_index>
   e. Reverse the file contents - reverse all the lines in the file (the first line should become last, etc.) and reverse each line content (as in d.)
   f. (Related to b., c., d. and e.) The result file (with the switches) should be saved in the same file that is initially specified by the user. Initial validation should also be performed as described in a. and if the file is with invalid content the switch operation should not be performed and the validation status should be printed.

Remarks: Proper validation of the user-provided arguments should be ensured. Consider proper exception handling and logging of events. Strive to design the application in object-oriented way.