Stack Indentation Checker

In this problem, you will create a program that can read in a series of files which are passed in as command line arguments. Your program will read through these files and generate listings with line numbers. While generating the listings your program will check for the correctness of the indentation used in the program using a "stack" algorithm. When improper indentation is found, your program will stop processing this file and move onto the next file.

Here are the rules for proper indentation:

1. completely blank lines are ignored.
2. the only thing we are looking at is the column number of the first non-blank character. We don't care about the actual text on the line or what the first character is ... it can be: if, {, for, etc. and it just doesn't matter.
3. When a line of text is indented, we don't care how many spaces it is indented ... i.e. 1, 3, 4, 8, etc. is just fine
4. When a line of text is indented less than the previous line, the column number must match the column number of a previous line which is still "in play". The following text will attempt to clarify the rules:

```
Example 1 of  indentation indentation rules
12        //good indentation
  34567   //good indentation

  34567   //good indentation
     6789           //good indentation
       8901234567   //good indentation
            3456    //good indentation
       89           //good indentation
    56789   /// Improperly indented ... no match for column 5
  345

Example 2 of improper indentation ( more subtle example)
12
  34567
     6789
       890
  3456  // At this point the "6789" and "890" lines are "popped" off
the indentation stack
       890   // Everything is properly indented up to this point
     67 // Improperly indented ... the previous "6789" line is not "in
play" because of the "3456" line.
```

The files that I want you to test your code with are found in the following links:

properlyIndented.txt - This file should pass

properlyIndented2.txt- This file should pass

notIndentedProperly.txt - This file should fail

notIndentedProperly2.txt- This file should fail

notIndentedProperly3.txt- This file should fail

So when you run your program, you will be using the command line arguments of:

properlyIndented.txt properlyIndented2.txt notIndentedProperly.txt notIndentedProperly2.txt notIndentedProperly3.txt

Here is the output of my published answer for this problem with the command line arguments of:

properlyIndented2.txt notIndentedProperly.txt notIndentedProperly2.txt

IndentationOutput.txt - output from running the published answer with the above command line arguments

---

If you find code templates useful, I am providing the following template for a solution to this problem. If you can produce the same results, without the template then write the code using your technique. I would like you to use a Stack in your solution.

```
class BadIndentationException extends RuntimeException
{
    BadIndentationException(String error)
    {
        super(error);
    }
}
```

```
public class IndentChecker {
    Stack<Integer> indentStack = new Stack<Integer>();

    private int findFirstNonBlank(String line)
    {
        // return index of first non-blank character
        // return -1 if the line doesn't contain a non-
blank character
    }

    private void processLine(String line, int lineNumber)
    {
        int index = findFirstNonBlank(line);

       // Skip blank lines ... i.e. return immediately

       // If the stack is empty, then push this index
onto the stack and return

       // If this index > than the top of the stack, then
push this index onto the stack and return

       // Pop off all Indentation indexes > index

       // At his point the top of the stack should match
the current index.  If it
       // doesn't throw a BadIndentationException.  In
the error message, include the source Line Number

    }

    public void checkIndentation(String fileName)
    {
        // Clear the stack

        Scanner input = null;
        try {
            input = new Scanner (new File(fileName));
            // read through the file line by line
            // for each line, call processLine to check
indentation
        }
```

```
        catch (BadIndentationException error)
        {
            System.out.println(error);
        }
        catch (FileNotFoundException e)
        {
            System.out.println("Can't open file:
"+fileName);
        }
        finally
        {
            if (input != null)
                input.close();
        }
    }


    public static void main(String[] args) {

        IndentChecker indentChecker = new
IndentChecker();

        for (int i=0; i < args.length; i++)
        {
            System.out.println("Processing file: " +
args[i]);
            indentChecker.checkIndentation(args[i]);
        }

    }

}
```