CSE 1322L - Lab 9

Introduction

In this lab, you will write a simple diff checker. A diff checker is a program that determines if two files have the same content. If the files have different content, the diff checker specifies what changes need to be done to either file in order for the files to be identical.

Requirements

The features described below must be in your program:

- **static String diff(File, File)**: This method reads both files passed as arguments at the same time, line by line. Depending on the contents of the two files, one of the strings below will be returned
  - If either file fails to open, return "X (The system cannot find the file specified)", where X is the file's name
  - If the files are identical, return "Files are identical"
  - If a line is different between the two files, the returned string must contain the line number, and what the difference is in both files in the following format:

    ```
    Line X:
    < Y
    > Z
    ```

    Where X is the line number, Y is the contents at that line on the first argument, and Z is the contents at that line on the second argument.

    - This must be done for every line where both files have different content
    - If either file runs out of lines, the returned string must also contain "Files have different number of lines"
  - Close the files before retuning from diff()
- **main()**: Prompts the user for two file names. Creates two File objects with said names, passes them to diff(), and prints its output.

Deliverables

- Lab9.java
  - diff()
  - main()

Considerations

- While you are being given sample input files to compare to the sample output below, your solution must work for any pair of files.
- You don't need to worry about file paths. As long as your code passes the correct file name (the one provided by the user) to the File object's constructor, you will get full credit.
- The Scanner constructor which accepts a File as an argument throws an exception, which you will need to catch.
    - The error message that you are required to print if the file cannot be found can be retrieved by calling getMessage() on the Exception thrown
- Note that diff() does no printing and requests no information from the user. It simply accepts files and returns a string. It is main() that is in charge of requesting from the user the names of the files and outputting the results of diff().
- Once diff() is done reading the files, you should close their associated Scanners by calling close()
    - As you've learned in lecture, the best place to close the Scanners is in the FINALLY portion of your TRY-CATCH block. To do so, you must declare the Scanners before the TRY-CATCH block, initializing them as null. Once inside the TRY-CATCH block, reinitialize the Scanners with the appropriate constructor. Note that the FINALLY block will need to check if the Scanners are not null, or calling close() will crash with a NullPointerException.
    - You could also use a TRY-WITH-RESOURCES, if you know how.

Sample Output (user input in red)

```
[Diff Detector]
Enter the name of file 1: humpty1.txt
Enter the name of file 2: humpty2.txt

Line 1
< Humpty Dumpty sat on a wall,
> Humpty Dumpty was sad on the wall,
Line 3
< All the king's horses and all the king's men
> All the king's horses and men

Program complete
```

Sample Output (user input in red)

```
[Diff Detector]
Enter the name of file 1: humpty1.txt
Enter the name of file 2: humpty1.txt

Files are identical.

Program complete
```

Sample Output (user input in red)

```
[Diff Detector]
Enter the name of file 1: humptyIncomplete.txt
Enter the name of file 2: humpty1.txt

Line 1
< Humpty Dumpty was sad on the wall,
> Humpty Dumpty sat on a wall,
Files have different number of lines

Program complete
```

Sample Output (user input in red)

```
[Diff Detector]
Enter the name of file 1: humptyIncomplete.txt
Enter the name of file 2: humpty2.txt

Files have different number of lines

Program complete
```

Sample Output (user input in red)

```
[Diff Detector]
Enter the name of file 1: theTaleOfHumptyDumpty.txt
Enter the name of file 2: humpty1.txt

theTaleOfHumptyDumpty.txt (The system cannot find the file specified)

Program complete
```