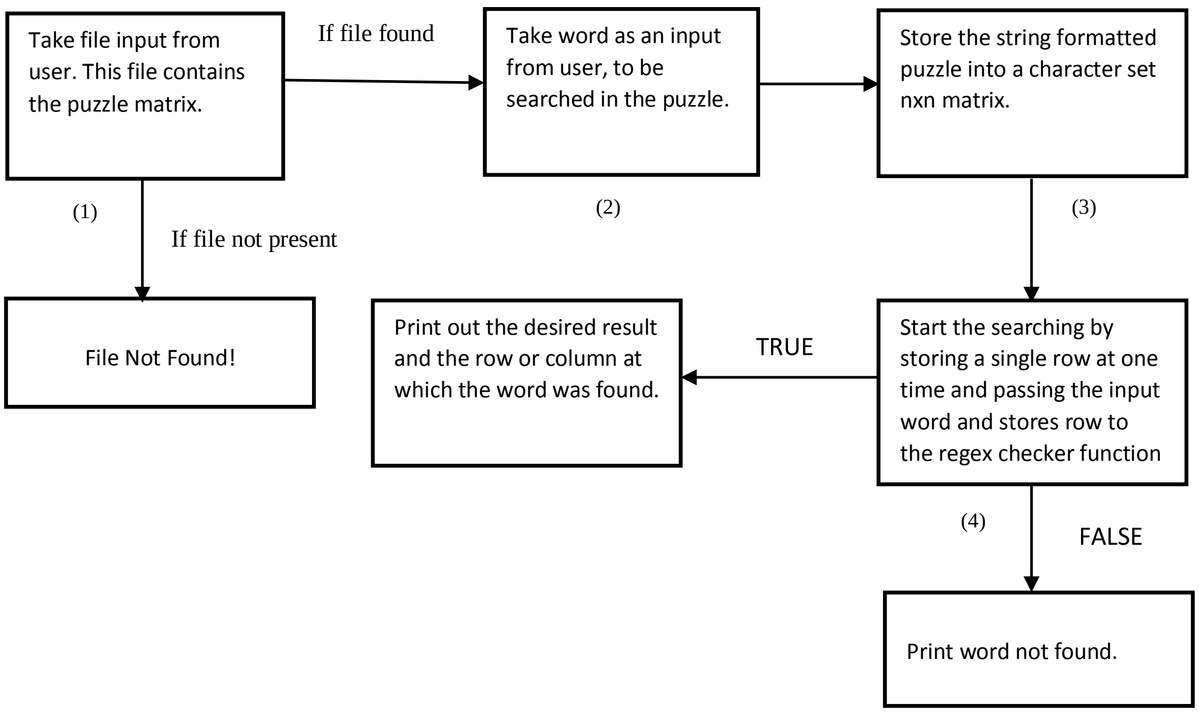
Report: Week 3

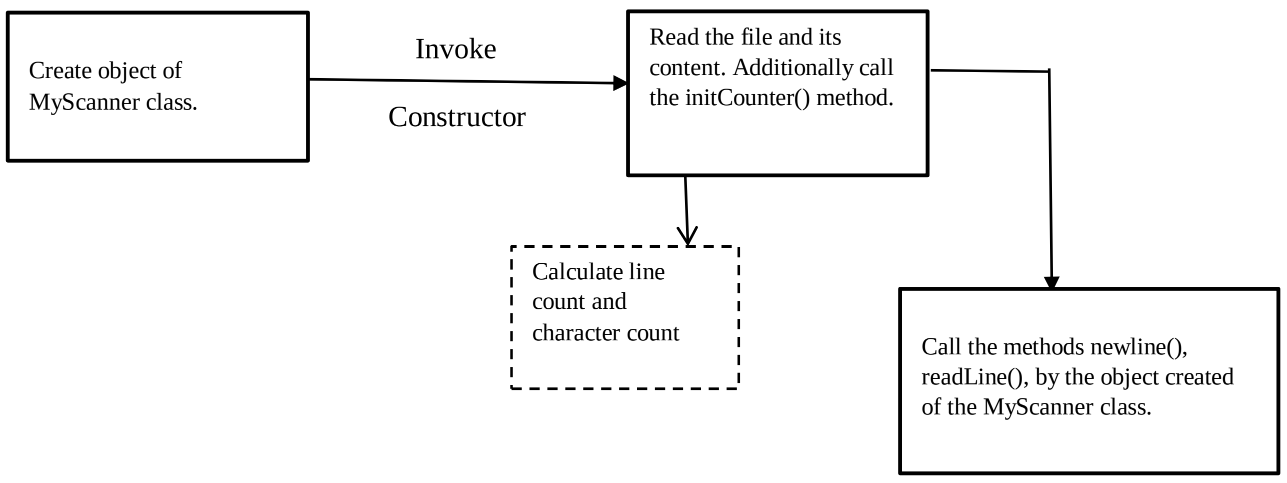
Assignment 1:WordSearch.java


Take file input from user. This file contains the puzzle matrix.
If file found
Take word as an input from user, to be searched in the puzzle.
Store the string formatted puzzle into a character set nxn matrix.
(1)
If file not present
(2)
(3)
File Not Found!
Print out the desired result and the row or column at which the word was found.
TRUE
Start the searching by storing a single row at one time and passing the input word and stores row to the regex checker function
(4)
FALSE
Print word not found.

The problem requires us to perform a word search puzzle by taking the puzzle and the word to be found in the puzzle from the user itself. We start the implementation by defining a Class with constructor called Work, the initial step is to ask the user for file input where the puzzle is present in a NXN matrix, we have also incorporated a File Not Found Exception. The file after being read creates a *'puzz'* character matrix which takes the size of it dynamically. The 'puzz' matrix is filled with individual letter from the puzzle, furthermore the user is prompted to provide a word they need to search until 'exit' word is typed. The main implementation of the program is initiated from here where a function for searching the word in all the eight direction is done. The first row of the matrix is read and is sent to a regex checker which then matches the word and checks if the buffer – where the matrix row is stored – contains the word or not, if not the next line is stored and so on. The same implementation is done for horizontal rows, vertical columns and diagonals. For reverse search the word is stored in a variable starting from the last letter stored as first and so forth, which also is passed to the regex checker for pattern matching word. The regex function is a Boolean function if matched, it returns True if not, False.

In this problem we mainly focused on learning how a Scanner class is used and how an input is taken using that and patterning matching using regex.

Assignment 2: MyScanner.java


Create object of MyScanner class.
Invoke
Constructor
Read the file and its content. Additionally call the initCounter() method.
Calculate line count and character count
Call the methods newline(), readLine(), by the object created of the MyScanner class.

The problem defined is to have a Scanner class implementation of our own. We have defined a Scanner class called 'MyScanner' class, furthermore creating an object of this class. We then invoke the constructor for variuous methods to be implemented. The methods to be implemented in the wrapper class are: hasNext() – gives the input token, by reading the file contents, close() – close the scanner, nextLine() – prints out the next line of the file from wherever the file pointer is present, getLineCount() & getCharacterCount() – prints the line and character count in the file inputted. For the implementation of these methods we read the file and its content, additionally call the initcounter which counts the character and the line count and keeps it ready for whenever the getLineCount() or getCharacterCount() are called. The hasNext() returns a boolean value if there is any input present else returns False, the nextLine() methods reads from the file if it's not empty and returns the line.

The implementation of our own Scanner class helped us to understand the inner working of how a certain class in Java works, furthermore what is encapsulation – what content should be visible to the user, what is necessary and required for it to function.