Cosc 1p03 Assignment 1

(Due date Jan 28th 16:00 est, Late date Jan 31st 16:00 est)

Object

To become familiar with Ragged Array Structures (array of arrays) and using 2-D array random indexing.

Back Ground

A word search puzzle consists of a board and a set of words. The board is usually laid out as a square of characters. In the assignment this square will be 25x25. The word set which is given requires the player to find the word within the board. Words may be represented forward, backward, up, down and on any of the 4 diagonals.

The Assignment

You are given a word puzzle which you must solve. The data file **wordSearch.dat** consists of 21 words followed by a 25x25 word board. You are to read the words into a ragged array. This array will consist of 21 rows, where each row represents 1 word. The board should be read into a second 25x25 char matrix. For each word in the word list, identify it's location in the board. For output, print the board with just the found words.

The solution to the puzzle is given below. This also represents the desired output from your program.

```
Found   21
I
 N              PECNEREFER
  T  R       C O
   E  E     S I
    RL T   O N
     FI UC T    T       C
L     AN PE    C       O
I      CKRM   E       L
S       EE OJ       L
T         DBC      ENTER
S D       O       CP
  R               T  A
  A            KI     C  S
  O   Y        OS      KS
  B    R      N  I     AA
  Y     O        ED  L  G
  E      M        VDC     E
  K   Y   E      EEIR
     H A   M      CPTA
      E R   N      NYAH
       A R   I      ETM
```

```
P A   A     IBI
 S     M     CUR
              SSP
  PROTOTYPE
```

How you should proceed with the solution and things to think about.

1. Create the data structures for the ragged array to hold the word list. Each word in the list will represent a right sized char array. The board is simply a 25x25 char matrix.
2. Read the data file loading both the word list and the board. When reading the word list consider reading a word as a String, and then using the toCharArray method to convert the string to a right sized character array.
3. Write a small test routine which prints both of these data structures to a displayer or system output. It is nice to know that your program works correctly up to this point. The output should match the input.
4. Write a search routine which, for each char in the matrix, will search all 8 directions for each word in the list.
5. Consider using a 2nd 25x25 matrix only for output. Once a word is found, write it into this output matrix. Printing the output matrix will give the above output. The output matrix should be initialized to blank characters.

Keep the problem manageable by implementing each search direction one at a time. E.g. get it working for finding words spelled forward first, then implement each additional direction 1 at a time. This allows for easy debugging. You will also notice, that once you get forward working, the rest are largely straight forward.

Dealing with a matrix such as this will require keeping a close tab on upper and lower bounds (Exceptions such as ArrayOutOfBounds).

A filter method

```
private char getChar(int i, int j)
```

can be used to index the puzzle matrix, implementing bounds checking. If a valid character can be returned then that character is returned, otherwise some non-alpha character can be returned which will result in a failed match. This simplifies the comparison algorithm.

Submission

Include in your submission, the IntelliJ project directory, zipped. Be sure your name and student number are included in your source file. Also, provide good commenting as per javaDoc standards (see class example code).

Assignment submission is via Sakai.