CSC 210 - Software Development
Programming Assignment 1

Due: Monday, August 30th at 11:59pm
Submit: WordSearch.java to Gradescope

Word Search!

Word search is a word game or puzzle that consists of the letters of words placed in a rectangular grid. The goal is to find and mark all of the English words that may occur horizontally, vertically, or diagonally. A list of the hidden words may or may not be provided.

An early creator of these puzzles, Norman E. Gibat, published a 20 x 20 grid (with the names of 34 Oklahoma cities and towns) in the Selenby Digest on March 1, 1968. Since then the puzzle has found its way into popular games and apps on the internet.

Requirements

A program is needed that can read a rectangular grid of characters from a text file and find all of the words that occur horizontally and vertically, both forwards and backwards and up and down. A valid word is at least three letters long and can be found in an English dictionary. In this case, the dictionary is provided as a text file. The size of the grid is at least 4x4 and the number of rows and columns will be on the first two lines of the file. An example of a grid file is:

```
6
6
y c o d e j
h s e y p k
l p h b w a
l o b w x z
w o b a a i
p l y y c g
```

and the first few lines of the dictionary file:

```
Aarhus
Aaron
Ababa
aback
...
```

and the output of the program would be:

```
cod
code
ode
ply
jed
doc
yes
abo
bow
spool
pool
way
loop
loops
```

It should be possible to run the program from a command line by providing the dictionary filename as the first argument and the grid filename as the second argument. It is expected that this program will be used as a tool in a pipeline of other tools and will not be extended for other uses.

Design

Based on the requirements it looks like we can take a simple procedural approach and break down the task into a sequence of subtasks or procedures, along with whatever data structures are needed. To find the subtasks it can help to write down the steps in 'pseudocode' and repeat this process, *refining each step* into greater detail. When working on the details of a subtask, such as searching for a word in the grid from left to right, it helps to *sketch diagrams* of what your program will need to do so that you can get some insight into how to solve the tricky parts.

Implementation

The program will be written in Java. Java strongly supports object-oriented programming but also allows procedural programming. We can put all of our code in the class that contains the main() function, adding any functions and data structures that we need. Try to write your program incrementally by first creating the overall structure with empty functions (stubs) and testing often to see that your program is always working as expected. The entire program will be in a single file, called **WordSearch.java**, and that is what you will submit to Gradescope. You can run the program from within your IDE by supplying the command line arguments to the IDE, usually in a dialog box associated with the 'run' command.

Testing

The program should be tested with the example grid and dictionary files provided. It should also be tested with other grids to explore the program's performance on a variety of typical grids and also on some atypical or defective grids. When testing, you are checking to see that the program satisfies the requirements.

Remember that when moving through the stages of requirements, design, implementation and testing, that it is often necessary to go back to a previous stage and revisit decisions and make changes. It is always better to make design changes early in a project rather than later.