P r o g r a m m i n g I
P R O J E C T II

In the Java programming language, write a program that allows you to play a familiar game "**WORD SEARCH**".

Game rules:
The game is played on two-dimensional field of size $m \times n$, where each element of the field is button with corresponding label that represents one letter from English alphabet.
Given the starting element, the user can build words (longer then 2 letters) by selecting adjacent letters (adjacent buttons). Starting element is randomly determined.
The user determines when building of the word is done. The word will be validated if it is contained in the attached list.
Diagonals are allowed when selecting adjacent elements. If the selected word is in the dictionary, the word is scored and the letters used are disabled. Points are defined as the sum of the score of each letter. Each letter is scored as 2^{index} (index is position of letter in word).
The game ends when the player so decides. At the end, final score should be displayed.

The basic game should require / support:

- creating a user interface that allows play (each time a different playing field), with each new start of the game we get a new layout (random) (40)
- creating a user interface that saves the playing field to a file (and load when the user wants to continue playing the game) (15)
- game control menu (replay, setting the size of playing field). (10)
- the possibility to display the longest word (or the length of the longest word) (10)

HINT:

- when choosing a letter, the game shows us possible sequels (in all valid directions) (15)
- by pressing an additional button, the game tells us whether it is still possible to continue the already selected string - by reviewing the attached dictionary of words (10)

ADDITIONAL POINTS:

- (even over 100%): find and list all possible words that can be composed from a random fame (words that are in the attached dictionary) (up to (15))
- Pay attention to the correct use of the basic elements of object-oriented programming - static methods are prohibited, individual program modules are defined as independent classes. (up to -5)
- Make sure the contracts are used correctly! (up to -5)