

Assignment 3 - Word Game

Due Date: Start of Class on November 15th

For this assignment we will be implementing a 2 player word game. The game operates as follows:

On their turn, each player will be given a hand of 10 letters. The players will then try to make the longest word possible using those 10 letters. If the word is valid, they get points equal to the length of the word. The game ends once one player reaches 20 points, and both players have had an equal number of turns.

To implement this program, you are tasked with creating the following methods:

String getRandomString()

This method will generate a random 10 letter String. The String will start off as an empty String ("") and will be generated one character at a time using the following steps:

- Use the Random class (as described in the notes) to generate a random number from 0-25.
- Use this number to create a random letter from 'A' to 'Z'. (Recall addition with chars)
- Add this new character to the existing String.

Once your String is 10 letters long, it should be returned.

int countLetterInString(String, char)

This method will take as input a String and a character and will return an integer representing the number of times that character occurred in the String. You should use a loop to go through the String one character at a time, using the charAt() method, and keep track of how many times the letter occurs.

EX: For the String "AARDVARK" and the char 'A', this method should return 3.

boolean confirmAnswer(String, String)

(Hint: Read carefully!)

This method will takes as input a String representing the letters available to the player (their hand), and a String representing the word they are trying to make from those letters. This method will return true if it is possible to make the word using their hand, false if it is not possible.

To confirm that it is possible to make the word using a player's hand, you must confirm that for each letter, the player's hand must have more of that letter than the word being made.

For every letter of the alphabet, you should use the **countLetterInString** method to compare the number of times each that letter occurs in both the player's hand and in the word, in order to confirm that each letter occurs more times in the player's hand than in the word. If this is not the case, the word is invalid, and the method should return false.

EX:

For the hand "ABSDCYHSLA", and word "CABS", this method should return true, because all of the letters used in the word appears a sufficient number of times in the hand.

For the hand "ABSDCYHSLA", and word "CABBY" this method should return false, because B occurs 2 times in the word, and only 1 time in the hand.

void playGame()

The playGame method is where the game will be run. At the start of this method, both players should begin with 0 points. You should then use a loop to allow players to continue taking turns making words until at least one player has obtained 20 points.

For each player's turn, you should do the following:

- Start by generating a random String using getRandomString, and print it to the player.
- Prompt the player to input a word using the provided letters.
- Use the confirmAnswer method to determine whether or not it is possible to make that word with the letters provided in the random String.
 - If it is possible, that player earns points equal to the length of the word they made.
 - If it is not possible, the player gets no points.
- In either case, play then passes on to the next player who performs steps.

Note: We are not confirming that the player's words are actually real words, you can leave it up to the players themselves to dispute the validity of a word if necessary.

Once at least 1 player has achieved 20 points, the game ends. Whichever player had the most points should receive a congratulatory message. If both players had an equal number of points, you should indicate that the game ended in a tie.

void main(String[])

The main method should only call the playGame method, and do nothing else.

Sample Execution:

Player 1, it is your turn with the letters: "ZZRAMZATZA"
Player 1's word: RAZZMATAZZ
RAZZMATAZZ is valid! Player 1 has 10 points.
Player 2, it is your turn with the letters: "SAOIJDOSAT"
Player 2's word: AT
AT is valid! Player 2 has 2 points.
Player 1, it is your turn with the letters: "ZGGZGIAGNI"
Player 1's word: ZIGZAGGING
ZIGZAGGING is valid! Player 1 has 20 points.
Player 2, it is your turn with the letters "HIOJDOSAJO"
Player 2's word: AT
AT is INVALID! Player 2 still has 2 points.
Player 1 wins with 20 points!

Style Guidelines:

- Good variable and method names
- Proper capitalization of variables, methods, constants, and classes
- Using constants (with the word final) in place of magic numbers
- Proper indentation
- Commenting your code (every snippet of a few lines should have its own comment)

What to submit:

Hard copy:

- A printout of the Java source code.
- A flowchart for the `confirmAnswer` method.
- A printout of a sample execution.
- Test inputs and outputs for the `confirmAnswer` and `countLetterInString` methods.

Electronic copy:

- Just the `WordGame.java` file. Please double check you have submitted this file and not the `.class` file.