# **Programming Assignment Unit 1**

Computer Science, University of the People

CS 1103-01 Programming 2 - AY2024-T4

Instructor, Rupa Sarda

April 18, 2024

For this assignment we were charged with building an application that takes a string and checks it in multiple methods to check length and count of words and characters.

A few important features that are used are the try…catch blocks to manage exception handling and closing of resources. Also we use the Hash and Set collection objects to help us collect information and use the results in a simple and efficient way.

Code:

import java.io.\*;

import java.util.\*;

public class Unit1 {

    private static final String TEXT\_FILE\_EXTENSION = ".txt";

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        String userInput;

        // User Input with Input Validation

        while (true) {

            System.out.println("Please enter some text you wish to analyze:");

            System.out.println("You can also enter a filename (e.g. 'sample.txt')");

            userInput = scanner.nextLine();

            if (userInput.endsWith(TEXT\_FILE\_EXTENSION) && checkFileExists(userInput)) {

                userInput = getFileContent(userInput);

                if (userInput != null) {

                    break;

                }

            } else if (!userInput.trim().isEmpty()) {

                break;

            } else {

                System.out.println("Input cannot be empty. Please try again.");

            }

        }

        // Make sure all characters are lowercase for consistency

        String userInputLower = userInput.toLowerCase();

        // Count how many characters are in the input not including spaces and

        // whitespace characters

        int countOfNonWhitespaceChars = countCharacters(userInputLower);

        System.out.println("Total number of characters (not including whitespace): " + countOfNonWhitespaceChars);

        // Count how many words are in the input (we assume words are separated by

        // whitespace)

        List<String> allWords = Arrays.asList(userInputLower.split("\\s+"));

        System.out.println("Total number of words: " + allWords.size());

        // Find the most common character in the input

        char topChar = findMostCommonCharacter(userInputLower);

        System.out.println("Most common character: " + topChar);

        // Find the frequency of a specific character from user input

        char userInputChar = getValidCharacter(scanner);

        int charFreq = Collections.frequency(convertToListOfCharacters(userInputLower), userInputChar);

        System.out.println("Frequency of '" + userInputChar + "': " + charFreq);

        // Find the frequency of a specific word from user input

        String userInputWord = getValidWord(scanner);

        int wordFrequency = Collections.frequency(allWords, userInputWord);

        System.out.println("Frequency of \"" + userInputWord + "\": " + wordFrequency);

        // Find the number of unique words in the input

        Set<String> uniqueWords = new HashSet<>(allWords);

        System.out.println("Number of unique words: " + uniqueWords.size());

        scanner.close();

    }

    private static boolean checkFileExists(String filename) {

        InputStream stream = Unit1.class.getResourceAsStream(filename);

        if (stream == null) {

            System.out.println("File does not exist. Please try again.");

            return false;

        }

        try {

            stream.close();

        } catch (IOException e) {

            System.out.println("Error closing file stream.");

        }

        return true;

    }

    private static String getFileContent(String filename) {

        StringBuilder content = new StringBuilder();

        try (InputStream stream = Unit1.class.getResourceAsStream(filename);

                Scanner fileScanner = new Scanner(stream)) {

            fileScanner.useDelimiter("\\A");

            while (fileScanner.hasNext()) {

                content.append(fileScanner.next());

            }

            return content.toString();

        } catch (IOException e) {

            System.out.println("Error reading file: " + filename);

            return null;

        }

    }

    private static int countCharacters(String text) {

        return text.replace(" ", "").length();

    }

    private static char findMostCommonCharacter(String text) {

        Map<Character, Integer> charFreqMap = new HashMap<>();

        for (char c : text.replace(" ", "").toCharArray()) {

            charFreqMap.put(c, charFreqMap.getOrDefault(c, 0) + 1);

        }

        return Collections.max(charFreqMap.entrySet(), Map.Entry.comparingByValue()).getKey();

    }

    private static char getValidCharacter(Scanner scanner) {

        String input;

        do {

            System.out.println("Please enter a single character:");

            input = scanner.nextLine().toLowerCase();

        } while (input.length() != 1 || !Character.isLetter(input.charAt(0)));

        return input.charAt(0);

    }

    private static String getValidWord(Scanner scanner) {

        String word;

        do {

            System.out.println("Please enter a word:");

            word = scanner.nextLine().trim().toLowerCase();

        } while (word.isEmpty());

        return word;

    }

    private static List<Character> convertToListOfCharacters(String text) {

        ArrayList<Character> characters = new ArrayList<>();

        for (char c : text.toCharArray()) {

            if (c != ' ') {

                characters.add(c);

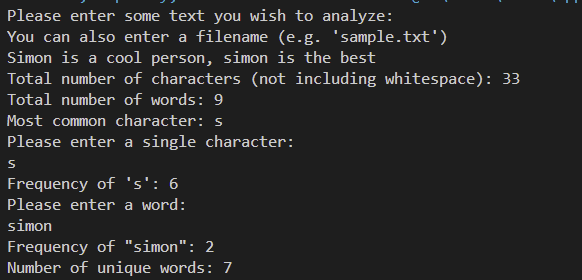
            }

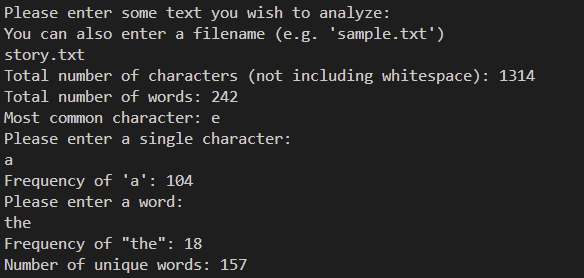
        }

        return characters;

    }

}

Screenshots:



## References

* Learning Guides  
  <https://my.uopeople.edu/course/view.php?id=7671>
* Eck, D. J. (2022). Introduction to programming using java version 9, JavaFX edition. Licensed under CC 4.0.  
  <https://math.hws.edu/javanotes/>
* Morelli, R. & Wade, R. (n.d.). Exceptions - When things go wrong. LibreTexts. Licensed under CC 4.0.  
  <https://eng.libretexts.org/Bookshelves/Computer_Science/Programming_Languages/Java_Java_Java_-_Object-Oriented_Programming_(Morelli_and_Walde)/00%3A_Front_Matter>

Source:

