ITT Head Hunter

EXERCISE 2.

Elaborado por Victor López

Problema.

Using first_names.txt (text file containing over five-thousand first names), begin by sorting it into alphabetical order. Then workout the alphabetical value for each name, multiply this value by its alphabetical position to obtain a name score.

> Example: assuming sorted and COLIN to be the 938^{th} name it is worth: 3+15+12+9+14. So, the score would be 938*53 = 49,714.

What is the total of all the name scores in the file?

Paso a paso.

Para resolver este problema, seguí los siguientes pasos:

1. Leí el archivo "first_names.txt" utilizando la clase FileReader y BufferedReader de Java.
2. Para almacenar los nombres leídos, utilicé un array de strings.
3. Utilicé el método split() de la clase String para separar los nombres que estaban separados por comas en el archivo.
4. Para agregar los nombres al array, utilicé el método Arrays.copyOf() y System.arraycopy() de Java.
5. Ordené los nombres en orden alfabético utilizando el método sort() de la clase Arrays.
6. Recorrí el array de nombres utilizando un bucle for y calculé la puntuación de cada nombre utilizando un método separado.
7. En el método de cálculo de puntuación, recorrí cada letra del nombre utilizando un bucle for y sumé los valores alfabéticos de cada letra utilizando su valor ASCII.
8. Multipliqué la puntuación obtenida por el índice del nombre en el array ordenado.
9. Sumé todas las puntuaciones de los nombres y las imprimí en la consola. *Total score of all the names in the file:* ***71433828***

Código.

```
import java.io.BufferedReader;

import java.io.FileReader;

import java.io.IOException;

import java.util.Arrays;


public class exercise2 {


   public static void main(String[] args) {


        String fileName = "first_names.txt"; // name of the file containing
the first names


        // read the names from the file and store them in an array

        String[] names = readNamesFromFile(fileName);


        // sort the names in alphabetical order

        Arrays.sort(names);


        int totalScore = 0; // variable to store the total score


        // calculate the score for each name and add it to the total score

        for (int i = 0; i < names.length; i++) {

            int nameScore = calculateExercise1(names[i], i+1);

            totalScore += nameScore;
```

```
        }


         System.out.println("Total score of all the names in the file: " +
totalScore);
    }


    /**
     * Reads the first names from the given file and returns them in an
array
     * @param fileName name of the file containing the first names
     * @return array of first names read from the file
     */
    private static String[] readNamesFromFile(String fileName) {
        String[] names = new String[0];
                    try (BufferedReader br = new BufferedReader(new
FileReader(fileName))) {
            String line;
            while ((line = br.readLine()) != null) {
                // split the line into names separated by commas
                String[] splitNames = line.split(",");
                // add each name to the array
                                names = Arrays.copyOf(names, names.length +
splitNames.length);
                    System.arraycopy(splitNames, 0, names, names.length -
splitNames.length, splitNames.length);
            }
        } catch (IOException e) {
            e.printStackTrace();
```

```
        }
        return names;
    }

    /**
     * Calculates the score for the given name
     * @param name name for which to calculate the score
     * @param position position of the name in the sorted array of names
     * @return score for the given name
     */
    private static int calculateExercise1(String name, int position) {
        int score = 0;
        for (int i = 0; i < name.length(); i++) {
            char c = name.charAt(i);
             score += (int) c - 64; // calculate the alphabetical value of
each character
        }
         return score * position; // multiply the score by the position of
the name
    }
}
```