

Ejercicios ficheros

Ejercicio 1: Contador de palabras Escribe un programa que lea un fichero de texto y cuente el número de palabras presentes en él. Puedes considerar que las palabras están separadas por espacios. Al final, muestra el resultado por pantalla.

archivo.txt

```
Este es un archivo de ejemplo.  
En este archivo tenemos varias palabras.  
Vamos a contar cuántas palabras hay en total.
```

Salida esperada:

```
20
```

Posibles soluciones:

```
import java.io.*;  
import java.util.Scanner;  
  
public class Ejercicio1 {  
    public static void main(String[] args) {  
        lecturaScanner();  
        lecturaFileReader();  
        lecturaBufferedReader();  
    }  
  
    public static void lecturaScanner() {  
        File f = new File("ficheros/archivo.txt");  
  
        int palabras = 0;  
        try (Scanner reader = new Scanner(f)) {  
            while (reader.hasNext()) {  
                reader.next();  
                palabras++;  
            }  
        } catch (FileNotFoundException e) {  
            System.out.println("No existe el archivo");  
        }  
  
        System.out.println(palabras);  
    }  
  
    public static void lecturaFileReader() {  
        File f = new File("ficheros/archivo.txt");  
  
        int palabras = 1;    // assume last word  
        try (FileReader reader = new FileReader(f)) {  
            int currCharValue = reader.read();  
  
            while (currCharValue != -1) {  
                char currChar = (char) currCharValue;  
  
                if (currChar == ' ' || currChar == '\n')  
                    palabras++;  
            }  
        }  
    }  
}
```

```

        palabras++;

        currCharValue = reader.read();
    }

    } catch (IOException e) {
        throw new RuntimeException(e);
    }

    System.out.println(palabras);
}

public static void lecturaBufferedReader() {
    File f = new File("ficheros/archivo.txt");

    int palabras = 1;
    try (BufferedReader reader = new BufferedReader(new FileReader(f))) {
        int currCharValue = reader.read();

        while (currCharValue != -1) {
            char currChar = (char) currCharValue;

            if (currChar == ' ' || currChar == '\n')
                palabras++;

            currCharValue = reader.read();
        }

    } catch (IOException e) {
        throw new RuntimeException(e);
    }

    System.out.println(palabras);
}
}

```

Ejercicio 2: Lector de archivo CSV Crea un programa que lea un archivo CSV (valores separados por comas) y muestre su contenido por pantalla. Puedes asumir que cada línea del archivo contiene una serie de valores separados por comas. Puedes utilizar la clase `Scanner` para leer el archivo y `String.split(",")` para separar los valores de cada línea.

personas.csv

```

Nombre,Edad,Ciudad
Juan,25,Madrid
María,30,Barcelona
Pedro,35,Valencia

```

Salida esperada:

```

Nombre: Juan, Edad: 25, Ciudad: Madrid
Nombre: María, Edad: 30, Ciudad: Barcelona
Nombre: Pedro, Edad: 35, Ciudad: Valencia

```

Posibles soluciones:

```

import java.io.*;
import java.util.Scanner;

public class Ejercicio2 {
    public static void main(String[] args) {
        lecturaScanner();
        lecturaFileReader();
        lecturaBufferedReader();

        try {
            Scanner sc = new Scanner(new File("ficheros/personas.csv"));

            while (sc.hasNextLine())
                System.out.println(sc.nextLine());

        } catch (FileNotFoundException e) {
            throw new RuntimeException(e);
        }

    }

    public static void lecturaScanner() {
        File f = new File("ficheros/personas.csv");

        try (Scanner reader = new Scanner(f)) {
            reader.nextLine();

            while (reader.hasNextLine()) {
                String line = reader.nextLine();
                String[] valores = line.split(",");

                System.out.printf("Nombre: %s, Edad: %s, Ciudad: %s\n", valores[0], valores[1],
valores[2]);
            }

        } catch (FileNotFoundException e) {
            System.out.println("El archivo no existe");
        }

    }

    public static void lecturaFileReader() {
        File f = new File("ficheros/personas.csv");

        try (FileReader reader = new FileReader(f)) {
            while ((char) reader.read() != '\n') {
            }

            int currCharValue = reader.read();
            while (currCharValue != -1) {
                StringBuilder line = new StringBuilder();

                while ((char) currCharValue != '\n' && currCharValue != -1) {
                    line.append((char) currCharValue);
                    currCharValue = reader.read();
                }
            }
        }
    }
}

```

```

        // show values
        String[] valores = line.toString().split(",");
        System.out.printf("Nombre: %s, Edad: %s, Ciudad: %s\n", valores[0], valores[1],
valores[2]);

        currCharValue = reader.read();
    }

    } catch (IOException e) {
        throw new RuntimeException(e);
    }
}

public static void lecturaBufferedReader() {
    File f = new File("ficheros/personas.csv");

    try (BufferedReader reader = new BufferedReader(new FileReader(f))) {
        reader.readLine();

        String line = reader.readLine();
        while (line != null) {
            String[] valores = line.split(",");
            System.out.printf("Nombre: %s, Edad: %s, Ciudad: %s\n", valores[0], valores[1],
valores[2]);

            line = reader.readLine();
        }

    } catch (IOException e) {
        throw new RuntimeException(e);
    }
}
}

```

Ejercicio 3: Buscador de palabras Escribe un programa que lea un archivo de texto y busque una palabra específica dentro de él. Debes permitir que el usuario ingrese la palabra a buscar por teclado. El programa debe mostrar cuántas veces se encuentra la palabra en el archivo.

palabras.txt

```

Este es un archivo de texto de ejemplo.
Vamos a buscar cuántas veces aparece la palabra ejemplo en este texto.
La palabra ejemplo puede aparecer varias veces en diferentes posiciones.

```

Salida esperada:

```

La palabra "ejemplo" aparece 3 veces en el archivo.

```

Posibles soluciones:

```

import java.io.*;
import java.util.Scanner;

public class Ejercicio3 {
    public static int cuentaAparicionesBufferedReader(File f, String palabra) {
        palabra = palabra.toLowerCase();

```

```

try (BufferedReader reader = new BufferedReader(new FileReader(f))) {
    int apariciones = 0;

    String line;
    while ((line = reader.readLine()) != null) {
        // split the line into string array ignoring punctuation marks
        String[] lineSplitted = line.replaceAll("[^a-zA-Z\\s]", "")
            .split(" ");

        // compare each word
        for (String currWord : lineSplitted) {
            if (currWord.equalsIgnoreCase(palabra))
                apariciones++;
        }
    }

    return apariciones;

} catch (IOException e) {
    return -1;
}
}

public static int cuentaAparicionesFileReader(File f, String palabra) {
    palabra = palabra.toLowerCase();

    try (FileReader reader = new FileReader(f)) {
        int apariciones = 0;

        StringBuilder currWord = new StringBuilder();
        int currCharValue = reader.read();
        while (currCharValue != -1) {
            // get the current word
            while (currCharValue != ' ' && currCharValue != '\n' && currCharValue != -1) {
                char currChar = (char) currCharValue;

                // append current char to current word only if it is letter
                if (Character.isLetter(currChar))
                    currWord.append(Character.toLowerCase(currChar));

                currCharValue = reader.read();
            }

            // compare the word
            if (currWord.toString().equalsIgnoreCase(palabra))
                apariciones++;

            // update curr char and curr word
            currWord.setLength(0);
            currCharValue = reader.read();
        }

        return apariciones;
    }
}

```

```

        } catch (IOException e) {
            return -1;
        }
    }

    public static int cuentaAparicionesScanner(File f, String palabra) {
        try (Scanner reader = new Scanner(f)) {
            int apariciones = 0;

            while (reader.hasNext()) {
                String curr = reader.next()
                    .toLowerCase()
                    .replaceAll("[^a-zA-Z]", "");

                if (curr.equals(palabra))
                    apariciones++;
            }

            return apariciones;

        } catch (FileNotFoundException e) {
            return -1;
        }
    }

    public static void main(String[] args) {
        File f = new File("ficheros/palabras.txt");

        // Pedir la palabra
        System.out.print("Introduce la palabra a buscar: ");
        Scanner reader = new Scanner(System.in);
        String palabra = reader.next();
        reader.close();

        // Imprimir las apariciones
        System.out.println(cuentaAparicionesBufferedReader(f, palabra));
    }
}

```

Ejercicio 4: Analizador de frecuencia de palabras Crea un programa que lea un archivo de texto y genere un informe de frecuencia de palabras. El programa debe contar cuántas veces aparece cada palabra en el archivo y luego mostrar un informe de las palabras y su frecuencia.

palabras2.txt

Este es un archivo de texto de ejemplo.
 En este archivo tenemos varias palabras repetidas.
 Vamos a contar cuántas veces aparece cada palabra en el texto.

Salida esperada:

```

Informe de aparicion de palabras
-----
La palabra "tenemos" aparece 1 veces
La palabra "aparece" aparece 1 veces

```

```
La palabra "de" aparece 2 veces
La palabra "a" aparece 1 veces
La palabra "archivo" aparece 2 veces
La palabra "el" aparece 1 veces
La palabra "en" aparece 2 veces
La palabra "vamos" aparece 1 veces
La palabra "veces" aparece 1 veces
La palabra "es" aparece 1 veces
La palabra "repetidas" aparece 1 veces
La palabra "este" aparece 2 veces
La palabra "texto" aparece 2 veces
La palabra "palabra" aparece 1 veces
La palabra "cada" aparece 1 veces
La palabra "palabras" aparece 1 veces
La palabra "un" aparece 1 veces
La palabra "varias" aparece 1 veces
La palabra "contar" aparece 1 veces
La palabra "cuantas" aparece 1 veces
La palabra "ejemplo" aparece 1 veces
```

Posibles soluciones:

```
import java.io.File;
import java.io.FileNotFoundException;
import java.util.HashMap;
import java.util.Map;
import java.util.Scanner;

public class Ejercicio4 {
    public static void main(String[] args) {
        File f = new File("ficheros/palabras2.txt");

        HashMap<String, Integer> frecuencias = new HashMap<>();

        try (Scanner reader = new Scanner(f)) {
            while (reader.hasNext()) {
                String curr = reader.next()
                    .toLowerCase()
                    .replaceAll("[^a-zA-Z]", "");

                if (frecuencias.containsKey(curr))
                    frecuencias.put(curr, frecuencias.get(curr) + 1);
                else
                    frecuencias.put(curr, 1);
            }
        } catch (FileNotFoundException e) {
            System.err.println("El archivo no existe");
        }

        System.out.println("Informe de aparicion de palabras");
        System.out.println("-----");

        for (Map.Entry<String, Integer> entry : frecuencias.entrySet()) {
            System.out.printf("La palabra \"%s\" aparece %d veces\n", entry.getKey(),
                entry.getValue());
        }
    }
}
```

