

Problema 8: Ficheros de texto

Datos personales	
Apellidos:	Prieels
Nombre:	Cedric

1 *Escribir en un fichero de texto, con formato*

Descripción

Se desea obtener una tabla de conversión de temperatura en grados Fahrenheit a Celsius. Escribir para ello un programa (un método main) que escriba un fichero llamado "temperaturas.txt" con las temperaturas entre -20°F y 120°F, ambas inclusive, de 10 en 10, y su conversión a °C. El formato del fichero será como en este ejemplo:

```
-20 F  -28.9 C
-10 F  -23.3 C
  0 F   -17.8 C
 10 F   -12.2 C
 20 F    -6.7 C
...
```

Como puede observarse, cada temperatura va en una línea, y los datos están en columnas de la misma anchura, con los grados Celsius con un decimal.

La conversión se realizará con esta fórmula: $C = (F - 32) / 1.8$

Respuesta que se pide

```
/**
 * Programa que nos da como resultado un nuevo fichero txt llamado temperaturas.txt con las
 * conversiones de las temperaturas entre -20°F y 120°F en grado Celsius (con un paso de 10°F).
 */
```

```
import java.io.*;
```

```
public static void main() {
```

```
    try (PrintWriter out = new PrintWriter(new FileWriter("temperaturas.txt"))) {
```

```
        int F = -20;
```

```
        double C = (F-32)/1.8;
```

```
        while (F <= 120) {
```

```
        C = (F-32)/1.8;
        out.printf("%d %1s %5.1f %1s", F, "F", C, "C");
        F += 10;
    }
} catch (IOException e) {

    System.out.println("Un problema ha ocurrido");

}
}
```

2 Leer de un fichero de texto

Descripción

Escribir una clase que contenga un método estático llamado `leeDatosDeFich` que recibe como parámetro el nombre de un fichero y retorna un array con los 10 primeros números reales contenidos en dicho fichero.

El fichero tiene mezcladas palabras con números. Un ejemplo de fichero a leer podría ser:

hola 7.0 3.45 otra 2.1 dos palabras 56.7 -34.5 -4.2 pepe 20.8 -3.5 10.3 lolo -18.70 40.2

El método `leeDatosDeFich` deberá ignorar las palabras y meter en el array los primeros 10 números que encuentre. Así, para el caso anterior el método debería retornar el array:

[7.0, 3.45, 2.1, 56.7, -34.5, -4.2, 20.8, -3.5, 10.3, -18.70]

En el caso de que el fichero contenga menos de 10 números, el método deberá lanzar la excepción `NoSuchElementException` (lanzada automáticamente por los métodos de la clase `Scanner` cuando se trata de leer un dato una vez alcanzado el fin del fichero). Esta excepción es de las no comprobadas, pero es conveniente indicar que se puede lanzar con una cláusula `@throws` en el comentario de documentación.

Respuesta que se pide

```
/**
 * Método que permite leer los 10 primeros números reales en un fichero de texto cuyo nombre se
 * pasa por argumento, que puede tener cualquier tipo de información.
 */

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

Programación, Curso 2015-2016

```
public static double[] leeDatosDeFich(String nombreFichero) {  
  
    double miArray[] = new double[10];  
    int i = 0;  
  
    try (Scanner in = new Scanner(new FileReader(nombreFichero))) {  
  
        while(in.hasNext() && i < 10) {  
  
            if (in.hasNextDouble()) {  
  
                miArray[i] = in.nextDouble();  
                i++;  
  
            }  
        }  
    } catch(FileNotFoundException e) {  
  
        System.out.println("No se encuentra el fichero " + nombreFichero);  
  
    }  
    return miArray;  
}
```

3 ***Lectura de un fichero de texto***

Descripción:

Escribir un método que recibe como parámetro un String que contiene el nombre de un fichero donde se encuentra un programa Java en código fuente. El método retorna un booleano indicando si hay llaves desparejadas o no.

El fichero de texto se leerá línea a línea (métodos `hasNextLine()` y `nextLine()` de la clase `Scanner`). Para cada línea leída se recorrerán todos sus caracteres para analizar solo aquellos que son llaves: '{' o '}'.

Para hacer este método se creará un contador de llaves abiertas, que inicialmente es cero. Cuando

Programación, Curso 2015-2016

durante el recorrido de los caracteres se encuentra una llave izquierda '{' el contador de llaves abiertas se incrementa en uno. Cuando se encuentra una llave derecha '}' el contador de llaves abiertas se decrementa. Las llaves están desparejadas si encontramos una cualquiera de estas dos situaciones:

- al final del método el contador de llaves abiertas no es cero
- en cualquier momento el contador de llaves abiertas es negativo

Respuesta que se pide

```
/**
 * lecturaCodigo es un método que recibe como parámetro un String que da el nombre del fichero txt
 * de entrada, y retorna un booleano indicando si hay llaves desparejadas o no en el fichero de
 * entrada.
 */

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

public static boolean lecturaCodigo(String nombreFichero) {

    int contadorLlaves = 0;
    String lineaAnalizada = "";

    try (Scanner in = new Scanner(new FileReader(nombreFichero))) {

        while (in.hasNextLine()) {

            lineaAnalizada = in.nextLine();

            if (lineaAnalizada.contains("{")) {
                contadorLlaves ++;
            } else if (lineaAnalizada.contains("}")) {
                contadorLlaves -- ;
            }

            if (contadorLlaves < 0) return false;

        }

        if (contadorLlaves != 0) {

            return false;

        }

    }

}
```

Programación, Curso 2015-2016

```
    } catch(FileNotFoundException e) {  
  
        System.out.println("No se encuentra el fichero " + nombreFichero);  
  
    }  
    return true;  
}
```