



AUT01_01. Actividad 1 Composición de Strings

1. Indicaciones para subir la tarea.

La tarea debe subirse con el nombre AUT01_01_Apellido1Apellido2Nombre.zip el proyecto debe llamarse AUT01_01_Apellido1Nombre....subirlo a la web antes de la fecha indicada en el evagd. La entrega tarde supone la mitad de la puntuación.

2. Creación de una página web simple.

El objetivo del ejercicio es componer una página web muy simple, utilizando Java y las funciones de concatenación de cadenas de caracteres.

2.1 La página web

Es simplemente un archivo de texto, que deberá contener un texto como el mostrado a continuación:

```
1 <html>
2 <head>
3     <title>Título de la página</title>
4 </head>
5 <body>
6     <h1>Hola: Nombre de la persona</h1>
7 </body>
8 </html>
```

Las cadenas que indican respectivamente “Título de la página” y “Nombre de la persona”, deben ser leídas por teclado e insertadas dentro del texto de la página.

2.2 La aplicación

La aplicación simplemente debe leer el título de la página y el nombre de la persona por consola y generar un archivo de texto como el indicado en el punto anterior en el directorio temporal.



```

<terminated> CartaApp [Java Application] C:\Archivos de programa\Jav
Introduce título: Título de la página

Introduce nombre: Nombre de la persona
    
```

La aplicación deberá componer una cadena de texto con el código HTML mostrado en el punto anterior. La cadena irá componiéndose fragmento a fragmento dado que tenemos que intercalar el título de la página y el nombre de la persona que serán leídos por teclado. Se recomienda el uso de la clase **StringBuffer**.

Para leer un dato por teclado:

```

System.out.print("Introduce título: ");
String titulo = sc.nextLine();
    
```

2.3 Clase de utilidad para generación del archivo de texto

Se proporciona un método, en la clase FileCreator, que fácilmente crea un archivo de texto, a partir de un String con el contenido de la página -si por ejemplo hemos utilizado un StringBuilder obtendríamos el texto con una expresión como: **sb.toString()** -

```

package es.cifpcm.io.file;

import java.io.File;
import java.io.FileWriter;
import java.io.IOException;

/**
 * @author Inma Martínez
 */
public class FileCreator {

    /**
     * Crea un archivo de texto, con el nombre y ruta indicados en
     * path y el texto pasado en el argumento text
     *
     * @param path Ruta y nombre del archivo de texto a crear
     * (ej: C:/TEMP/hola.html)
     * @param text String que contiene el texto de la página
     */
    public static void createFile(String path, String text) {

        File file = new File(path);
    
```



```
        try {
            FileWriter fw = new FileWriter(file);
            fw.write(text);
            fw.close();
        } catch (IOException e) {
            System.err.println(e.getMessage());
            e.printStackTrace();
        }
    }
}
```

Crear un archivo fuente FileCreator.java en la carpeta es/cifpcm/io/file

Copiar en ese archivo el código FUENTE anterior

Compilar el archivo fuente para obtener FileCreator.class

Para llamar a esta función desde nuestro programa, simplemente haremos:

```
FileCreator.createFile(ficheroTexto, miTexto);
```

El directorio será el temporal del sistema dado por la variable del sistema y el fichero Apellido1Nombre.html. Sacar ruta completa por consola para que el usuario del programa sepa donde encontrar su nueva web (y yo para corregirla).

3. Puntuaciones.

Uso de StringBuilder +2

Uso de CamelCase apropiadamente +2

Nombre de ficheros, variables y paquetes adecuadamente +3

Generación del fichero correctamente +3