

IES Virrey Morcillo. (Nombre Ciclo Formativo)

Generación de documentación en un proyecto java en eclipse

Módulo: Entornos de desarrollo

Nombre del Alumno: Santiago Alarcón

Fecha: 23.03.21

Contenido

Criterios de Evaluación 1

Material 1

Objetivos 1

1. Introducción 1
2. Tareas 2
3. Conclusión 2
4. Links 2

Criterios de Evaluación

- CE.x.x) ...

Material

Haremos uso de los siguientes recursos para realizar las tareas:

- Ordenador
- Máquina virtual con .ova de windows 10
- Conexión de Internet
- Eclipse IDE
- Procesador de Textos Pages

Objetivos

- Conocer ...
- Repasar ...

1. Introducción

La documentación Javadoc es una colección de páginas HTML de todas las clases, métodos, parámetros y retornos junto con la información y especificaciones que quiera incluir el desarrollador de la API que en el caso de las clases de JDK incluye abundantes e interesantes detalles de implementación a tener en cuenta al usar las clases. El Javadoc es también es una herramienta de línea de comandos que permite generar la colección de páginas HTML a partir del código fuente Java.

2. Tareas

Tarea 1) Añade autor y versión a las clases dentro del archivo java
Solución:

```
1
2 import java.util.ArrayList;
3 import java.util.Iterator;
4
5 /**
6  * Esta clase define un array a partir de ArrayList sobre el que opera de diferentes formas
7  * @author Santiago
8  * @version 23/03/2021.1
9  */
10
11 public class ArrayListString {
12
13
14     public static void main(String[] args) {
15
16         // Declaración el ArrayList
17         System.out.println("... Creamos un ArrayList de Strings ...");
18         ArrayList<String> nombreArrayList = new ArrayList<String>();
19
20         // Añadimos 10 Elementos en el ArrayList
21         System.out.println("\n... Añadimos 10 elementos al ArrayList ...");
```

Tarea 2) Añade descripciones a las clases, métodos, constructores y sus parámetros añadiendo etiquetas HTML dentro del archivo java

Solución:

```
import java.util.ArrayList;
import java.util.Iterator;

/**
 * Esta clase define un array a partir de ArrayList sobre el que opera de diferentes formas
 * @author Santiago
 * @version 23/03/2021.1
 */

public class ArrayListString {

/**
 * El main se va a utilizar para facilitar el trabajo sobre el código
 * @param args
 */
    public static void main(String[] args) {
        //Imprimir lo que se está haciendo
        System.out.println("... Creamos un ArrayList de Strings ...");
        //Declaración del ArrayList
        ArrayList<String> lista = new ArrayList<String>();

        // Añadimos 10 Elementos en el ArrayList
        lista = anyadirElementos(lista);

        //Añadimos elemento en la posición 2
        lista = anyadirElementoPosicion(lista,2,"Elemento 3");

        // Imprimimos los Elementos del ArrayList
        mostrarlista(lista);

        //Contar los elementos del Array
        contarelementos(lista);

        // Eliminamos el primer elemento del ArrayList, es decir el que ocupa la posición
        '0'
            lista = borrarelemento(lista, 0);

        // Eliminamos los elementos de ArrayList que sean iguales a "Elemento 3"
        lista = borrarelementotres (lista);

        // Imprimimos el ArrayList despues de eliminar los elementos iguales a "Elemento
        3"
            mostrarlista(lista);

        // Mostramos el numero de elementos que tiene el ArrayList tras las eliminaciones:
        contarelementos(lista);

        }//fin del main

        /**
         * Añadimos 10 Elementos en el ArrayList
         * @param nombreArrayList variable del array creado que vamos a rellenar
         * @return devolver el array con sus valores
         */
        public static ArrayList<String> anyadirElementos (ArrayList<String>
nombreArrayList){
            System.out.println("\n... Añadimos 10 elementos al Arratlist ...");
            for (int i=1; i<=10; i++){
                nombreArrayList.add("Elemento "+i);
            }
            return nombreArrayList;
        }//fin de la función AnyadirElementos

        /**
         * Añadimos un nuevo elemento al ArrayList en la posición 2

```

```

    * @param nombreArrayList El array con el que trabajamos
    * @param posicion Posicion en la que vamos a añadir el elemento
    * @param texto El nombre a mostrar para este cambio en el array
    * @return devuelve el array modificado
    */
    public static ArrayList<String> anyadirElementoPosicion (ArrayList<String>
nombreArrayList, int posicion, String texto){
    System.out.println("\n... Añadimos 1 elemento (" + texto + ") al ArrayList
en la posición 2 ...");
    nombreArrayList.add(2, "Elemento 3");
    return nombreArrayList;
} // fin función anyadirElementoPosicion

/**
 * Imprimimos los Elementos del ArrayList
 * @param nombreArrayList El array que se va a imprimir
 */
public static void mostrarlista (ArrayList<String> nombreArrayList){
    System.out.println("\nImprimimos los elementos del ArrayList: ");
    Iterator<String> reiterator = nombreArrayList.iterator();
    while(reiterator.hasNext()){
        String elemento = reiterator.next();
        System.out.print(elemento+" / ");
    }
} //fin de la función mostrarlista

/**
 * Obtenemos el numero de elementos del ArrayList
 * @param nombreArrayList El array cuyos elementos se cuentan
 */
public static void contarelementos (ArrayList<String> nombreArrayList){
    int numElementos = nombreArrayList.size();
    System.out.println("\n\nEl ArrayList tiene "+numElementos+" elementos");
} // fin de contarelementos

/**
 * Eliminamos el primer elemento del ArrayList, es decir el que ocupa la
posición '0'
    * @param nombreArrayList El array sobre el que trabajamos
    * @param posicion La posición del elemento que se va a borrar
    * @return Devolver el array modificado
    */
    public static ArrayList<String> borrarelemento (ArrayList<String>
nombreArrayList, int posicion){
    System.out.println("\n... Eliminamos el primer elemento del ArrayList
("+nombreArrayList.get(0)+")...");
    nombreArrayList.remove(0);
    return nombreArrayList;
} // fin de borrarelemento

/**
 * Eliminamos los elementos de ArrayList que sean iguales a "Elemento 3"
 * @param nombreArrayList El array con el que estamos trabajando
 * @return devolver el array sin el elemento repetido
 */
    public static ArrayList<String> borrarelementotres (ArrayList<String>
nombreArrayList){
    System.out.println("\n... Eliminamos los elementos de ArrayList que sean
iguales a \"Elemento 3\" ...");
    Iterator<String> reiterator = nombreArrayList.iterator();
    while(reiterator.hasNext()){
        String elemento = reiterator.next();
        if(elemento.equals("Elemento 3"))
            reiterator.remove(); // Eliminamos el Elemento que hemos
obtenido del Iterator
    }
    return nombreArrayList;
} // fin borrarelementotres
}

```

Tarea 3) Genera otra vez la documentación con javadoc y realiza un documento pdf con las páginas que ha generado javadoc.

Solución: Documento pdf adjunto en la entrega

```
Generating /Users/santi/eclipse-workspace/53667350R_Santiago/doc/package-summary.html...
Generating /Users/santi/eclipse-workspace/53667350R_Santiago/doc/package-tree.html...
Generating /Users/santi/eclipse-workspace/53667350R_Santiago/doc/constant-values.html...
Generating /Users/santi/eclipse-workspace/53667350R_Santiago/doc/class-use/ArrayListString.t
Generating /Users/santi/eclipse-workspace/53667350R_Santiago/doc/package-use.html...
Generating /Users/santi/eclipse-workspace/53667350R_Santiago/doc/overview-tree.html...
Generating /Users/santi/eclipse-workspace/53667350R_Santiago/doc/deprecated-list.html...
Building index for all the packages and classes...
Generating /Users/santi/eclipse-workspace/53667350R_Santiago/doc/index-files/index-1.html...
Generating /Users/santi/eclipse-workspace/53667350R_Santiago/doc/index-files/index-2.html...
Generating /Users/santi/eclipse-workspace/53667350R_Santiago/doc/index-files/index-3.html...
Generating /Users/santi/eclipse-workspace/53667350R_Santiago/doc/index-files/index-4.html...
Building index for all classes...
Generating /Users/santi/eclipse-workspace/53667350R_Santiago/doc/allclasses-index.html...
Generating /Users/santi/eclipse-workspace/53667350R_Santiago/doc/allpackages-index.html...
Generating /Users/santi/eclipse-workspace/53667350R_Santiago/doc/index.html...
Generating /Users/santi/eclipse-workspace/53667350R_Santiago/doc/help-doc.html...
2 warnings
```

3. Conclusión

Java doc es una automatización del proceso de generar documentación asociada a código con el añadido de su formato html de cara a la accesibilidad.

4. Links

<https://picodotdev.github.io/blog-bitix/2017/09/la-herramienta-de-documentacion-javadoc-de-java/>

<http://www.dit.upm.es/~pepe/doc/adsw/base/doc/doc.htm>