

Documentación en Java

09/11/2021

Alcántara Estrada Kevin Isaac

Documentar el código de un programa es añadir información suficiente para explicar qué es lo que éste hace, paso por paso, para que si una persona lee nuestro código pueda enterlo y comprender cómo es su funcionamiento.

En Java, se cuenta con Javadoc, que es una utilidad incluida en el JDK y nos ayuda a generar documentación automática en formato HTML a partir del código fuente Java.

Para que Javadoc funcione adecuadamente, los comentarios que considerará dentro del código deben seguir ciertos lineamientos, la sintáxis que deben seguir dichos comentarios es la siguiente:

```
/**
 * Texto con alguna descripción
 */
```

En Java suelen documentarse las clases, los métodos, paquetes y atributos; para ello, se utiliza como herramienta las etiquetas de Javadoc, que nos ayudan a clasificar la información que estamos añadiendo, entre otras cosas. Éstas inician con el símbolo de @ seguido de la palabra de la etiqueta

Algunas de éstas etiquetas son las siguientes:

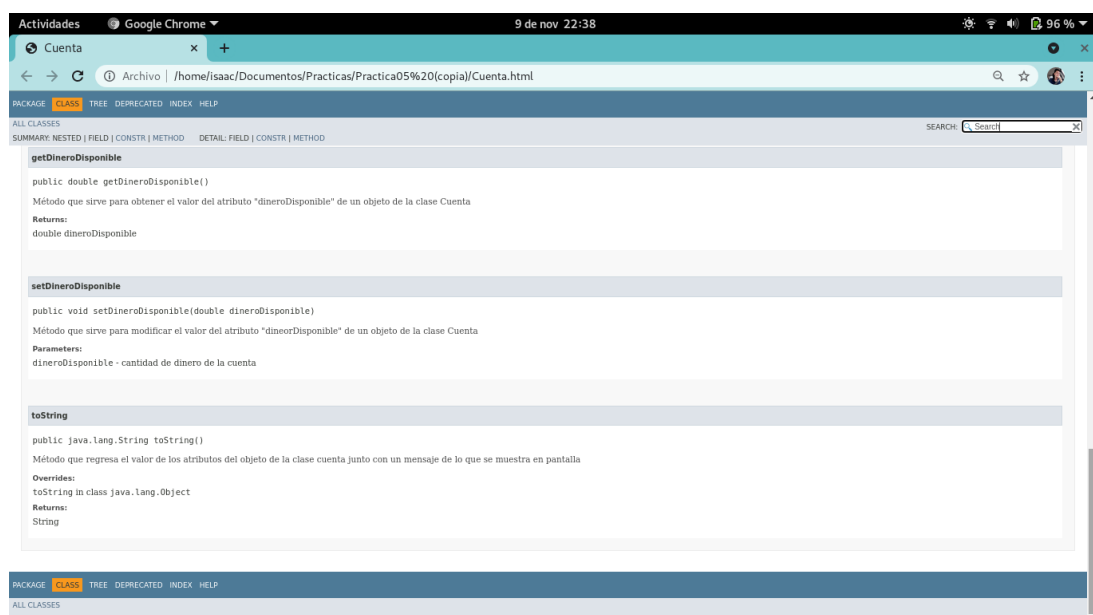
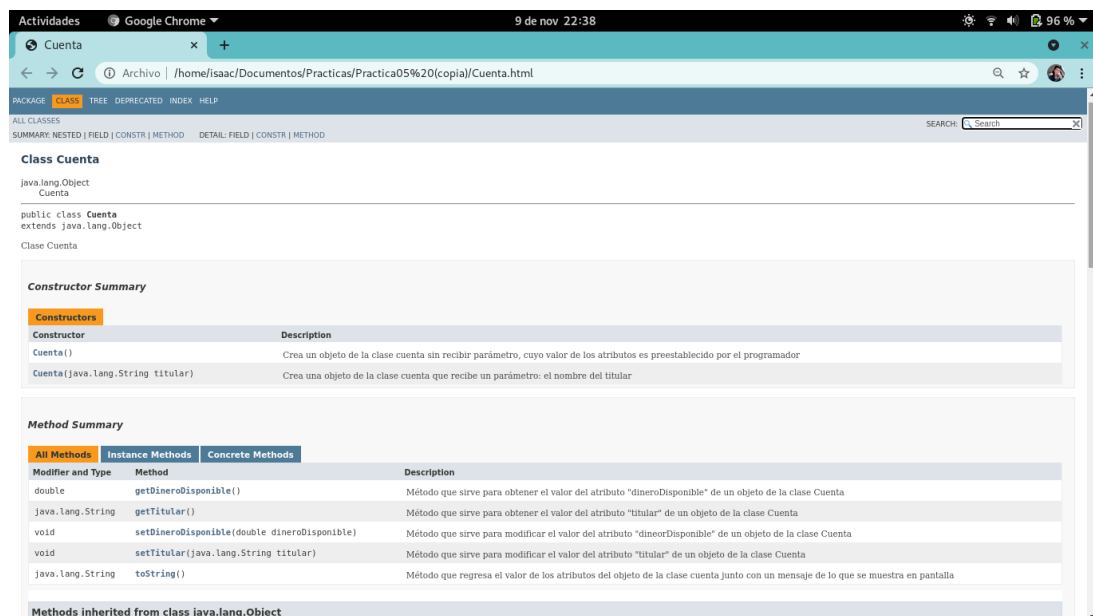
1. @author Sirve para escribir el nombre (o nombres) de quien desarrolló el método o la clase.
2. @deprecated Indica que el método o clase es obsoleto y no se recomienda su uso.
3. @param Definición de un parámetro de un método.
4. @return Informa sobre lo que devuelve el método.
5. @throws Excepción lanzada por el método
6. @version Indica la versión del método o clase
7. @see Sirve para Asociar o hacer referencia a otra clase, método o enlace.
8. @exception Realiza una función similar a la etiqueta @throws.
9. @since Se indica desde qué versión existe el método o la clase.

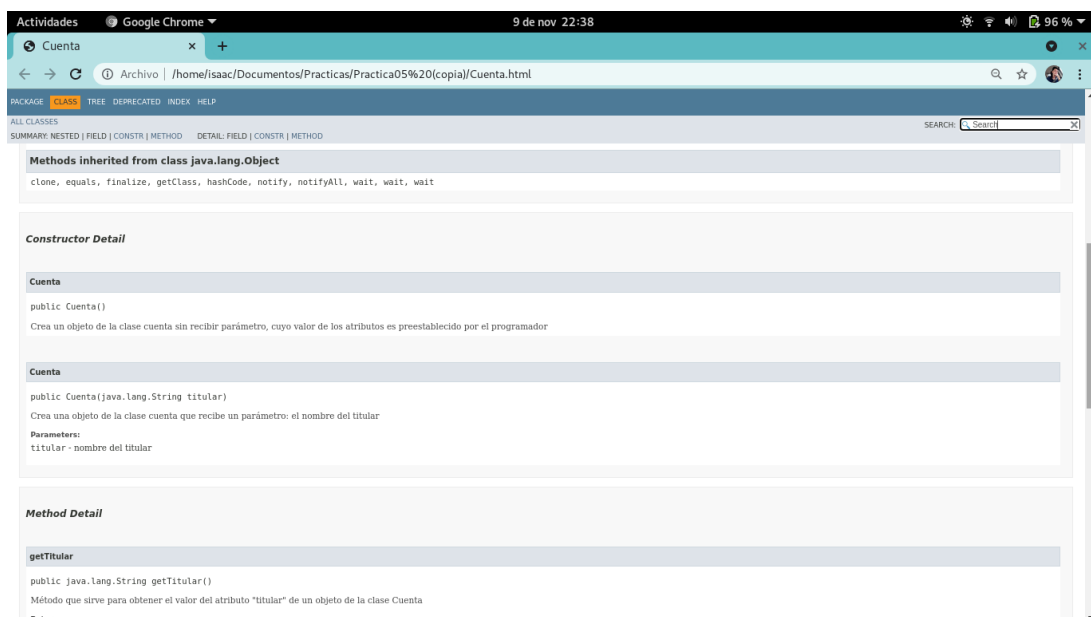
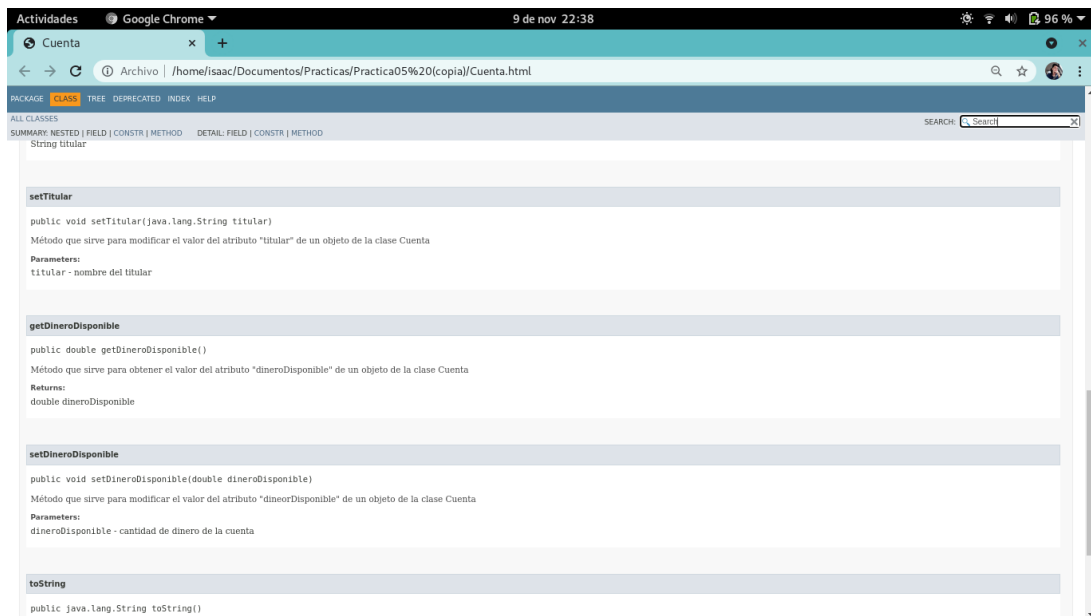
10. `{@code}` Incluye en el comentario un trozo de código que se formatea de manera especial.
11. `{@docRoot}` Incluye una ruta relativa al directorio raíz donde se genera la documentación.
12. `{@link}` Incluye un enlace a otra sección de la documentación, método o clase.
13. `{@linkplain}` Similar a `@link` pero el enlace es un texto plano.

Para generar la documentación, desde la terminal, escribimos el comando “javadoc” y el nombre del archivo separados por un espacio (javadoc Archivo.java)

Algunos ejemplos de la documentación que se generaron los siguientes:

EJEMPLO 1)





EJEMPLO 2)

Actividades Google Chrome 9 de nov 22:47

Primitivo x +

← → C Archivo | /home/isaac/Documentos/Practicas/Practica05%20(copia)/Primitivo.html

PACKAGE CLASS TREE DEPRECATED INDEX HELP

ALL CLASSES SEARCH Search

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

Class Primitivo

java.lang.Object
Primitivo

public class **Primitivo**
extends java.lang.Object

Clase Primitivo

Constructor Summary

Constructors	Description
Primitivo()	

Method Summary

All Methods	Static Methods	Concrete Methods	Description
static void	main(java.lang.String[] args)		Método para imprimir la cantidad de bytes que ocupa cada tipo de dato primitivo en java

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

Actividades Google Chrome 9 de nov 22:47

Primitivo x +

← → C Archivo | /home/isaac/Documentos/Practicas/Practica05%20(copia)/Primitivo.html

PACKAGE CLASS TREE DEPRECATED INDEX HELP

ALL CLASSES SEARCH Search

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

static void main(java.lang.String[] args) Método para imprimir la cantidad de bytes que ocupa cada tipo de dato primitivo en java

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

Primitivo

public Primitivo()

Method Detail

main

public static void main(java.lang.String[] args)

Método para imprimir la cantidad de bytes que ocupa cada tipo de dato primitivo en java

Parameters:
args -

PACKAGE CLASS TREE DEPRECATED INDEX HELP

ALL CLASSES

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD