UD06: Documentación de Código

1. Introducción

Documentar el código de un programa es añadir suficiente información como para explicar lo que hace, punto por punto, de forma que no sólo los ordenadores sepan qué hacer, sino que además los humanos entiendan qué están haciendo y por qué.

Porque entre lo que tiene que hacer un programa y cómo lo hace hay una distancia impresionante: todas las horas que el programador ha dedicado a elaborar una solución y escribirla en el lenguaje correspondiente para que el ordenador la ejecute ciegamente.

Documentar un programa no es sólo un acto de buen hacer del programador por aquello de dejar la obra rematada. Es además una necesidad que sólo se aprecia en su debida magnitud cuando hay errores que reparar o hay que extender el programa con nuevas capacidades o adaptarlo a un nuevo escenario. Hay dos reglas que no se deben olvidar nunca:

1. todos los programas tienen errores y descubrirlos sólo es cuestión de tiempo y de que el programa tenga éxito y se utilice frecuentemente
2. todos los programas sufren modificaciones a lo largo de su vida, al menos todos aquellos que tienen éxito

Un programa, si tiene éxito, probablemente será modificado en el futuro por quien lo codificó o por otro programador. Pensando en esta revisión de código es por lo que es importante que el programa se entienda: para poder repararlo y modificarlo.

2. ¿Qué hay que documentar?

Hay que añadir explicaciones a todo lo que no es evidente.

No hay que repetir lo que se hace, sino explicar por qué se hace

Y eso se traduce en:

- ¿de qué se encarga una clase? ¿un paquete?
- ¿qué hace un método?
- ¿cuál es el uso esperado de un método?
- ¿para qué se usa una variable o un atributo?
- ¿cuál es el uso esperado de un atributo?
- ¿qué algoritmo estamos usando? ¿de dónde lo hemos sacado?
- ¿qué limitaciones tiene el algoritmo? ¿qué limitaciones tiene la implementación?
- ¿qué se debería mejorar ... si hubiera tiempo?

3. Tipos de comentarios

En Java disponemos de tres notaciones para introducir comentarios:

Comentarios de una línea:

- Comienzan con los caracteres "//" y terminan con la línea
- Se utiliza para documentar código que no necesitamos que aparezca en la documentación externa (la que genera javadoc). Este tipo de comentarios se usa incluso cuando el comentario ocupa varias líneas, cada una de las cuales comienza con "//"

Comentarios de varias líneas:

- Comienzan con los caracteres "/*" y terminan con los caracteres "*/".
- A menudo e utiliza para eliminar código. Es habitual que el código obsoleto no queremos que desaparezca y lo mantenemos "por si acaso". Para que no se ejecute, se comenta.
  (En inglés se suele denominar "comment out")

Comentarios javadoc:

- Comienzan con los caracteres "/** ", se pueden prolongar a lo largo de varias líneas (que probablemente comiencen con el carácter "*") y terminan con los caracteres "*/".
- Se utilizan para generar documentación externa.

4. ¿Cuándo hay que poner un comentario?

1. Siempre, al principio de cada clase
2. Siempre, al principio de cada método
3. Siempre, ante cada variable de clase (atributos estáticos, constantes).
4. Al principio de un fragmento de código que no resulte evidente.
5. Cuando el programa haga algo “raro”

> Nota : Cuando un programa se modifica, los comentarios deben modificarse al tiempo, no sea
> que los comentarios acaben refiriéndose a un algoritmo que ya no utilizamos.

5. JavaDoc

El paquete de desarrollo Java incluye una herramienta, javadoc, para generar un conjunto de páginas web a partir de los ficheros de código. Esta herramienta toma en consideración algunos comentarios para generar una documentación bien presentada de clases y componentes de clases (variables y métodos).

Aunque javadoc no ayuda a la comprensión de los detalles de código, si ayuda a la comprensión de la arquitectura de la solución, lo que no es poco. Se dice que javadoc se centra en la interfaz (API - Application Programming Interface) de las clases y paquetes Java.

Javadoc realza algunos comentarios, de los que exige una sintaxis especial. Deben comenzar por "/**" y terminar por "*/", incluyendo una descripción y algunas etiquetas especiales:

```
/**
* Parte descriptiva.
* Que puede consistir de varias frases o párrafos.
*
* @etiqueta texto específico de la etiqueta
*/
```

Estos comentarios especiales deben aparecer justo antes de la declaración de una clase, un atributo o un método en el mismo código fuente. En las siguientes secciones se detallan las etiquetas (tags) que javadoc interpreta y posteriormente convierte en documentación.

Como regla general, hay que destacar que la primera frase (el texto hasta el primer punto) recibirá un tratamiento destacado, por lo que debe aportar una explicación concisa y contundente del elemento documentado. Las demás frases entrarán en detalles.

Etiqueta	Dónde se usa	Objetivo
`@autor` nombre	Clases, interfaces	Indicar el autor del código. Se pone una etiqueta por cada autor
`@version` identificadorDeVersión	Clases, interfaces	Información acerca de la versión
`@since`	Clases, métodos	Desde qué versión está. Ej: desde JDK 1.1
`@deprecated`	Clases, métodos	Para indicar que algo no debe utilizarse ya, ha quedado obsoleto, aunque se mantenga por compatibilidad. Se suele acompañar de qué es lo que hay que utilizar el su lugar.
`@see ClassName`	Clases, interfaces, métodos y atributos.	Pondrá la dirección para conectarse con esta clase en la documentación
	Clases,	

Etiqueta	Dónde se usa	Objetivo
`@see` ClassName#NombreMétodo	interfaces, métodos y atributos.	Pondrá la dirección para conectarse con este método en la documentación.
`@return` descripción	Métodos	Para describir los valores devueltos por cada método y su tipo.
`@exception` nombre descripción	Métodos	Excepciones que el método puede elevar. Se pone una etiqueta para cada excepción posible. Se suelen ordenar alfabéticamente.
`@param` nombre descripción	Métodos	Para describir los parámetros, su utilización y su tipo. Se pone una etiqueta para cada parámetro

6. Fuentes de información

- Wikipedia
- Code&Coke (Fernando Valdeón)
- Apuntes IES El Grao (Mª Isabel Barquilla?)
- Apuntes IOC (Marcel García)
- Apuntes José Luis Comesaña
- Apuntes IES Luis Vélez de Guevara 17-18 (José Antonio Muñoz Jiménez)
- Oracle proporciona material de cómo documentar las interfaces de los programas: (How to write doc comments for the javadoc tool)