

Otros comentarios de Java

Solamente has utilizado un tipo de comentario de Java y aun cuando tratamos de elegir nombres claros que nos recordaran qué es lo que representan (nombres nemónicos), no es suficiente con el identificador para saber qué está representando.

Java tiene otros dos tipos de comentarios. Los primeros sirven sólo internamente para recordar al programador o a alguien que quiera revisar el código, pedazos de código que pudiesen no ser muy claros. Empiezan con `/*` y terminan con `*/`. Se pueden extender tantas líneas como se desee, pero no pueden estar anidados (uno dentro de otro) pues el comentario termina en cuanto se encuentra los primeros `*/` caracteres que cierran comentarios. Por ejemplo:

```
1 public String toString ();
2 /* Este metodo lo vamos a implementar de
3    manera chistosa */
```

En la línea 2 empieza un comentario, que Java no trata de entender, y termina en la línea 3. Si quisieras comentar todo este segmento de código, colocando `/*` antes de la primera línea y `*/` después de la última, el compilador te marcaría un error, pues trataría de interpretar como si fuere Java la cuarta línea y te diría que una declaración de tipo no puede empezar con `*/`.

```
1 /* public String toString ();
2 /* Este metodo lo vamos a implementar de
3    * manera chistosa */
4 */
```

Este tipo de comentario es muy útil cuando se desea, por ejemplo, comentar todo un método o una parte importante, aunque sí tiene el problema de que si aparece otro comentario de estos en el texto que deseamos comentar, vamos a tener problemas. Por eso, casi cualquier IDE para Java a la que le pedimos que nos comente toda una región, utiliza comentarios de escape de línea para que no vaya a haber problemas del tipo que acabamos de mencionar.

Otro posible problema con los comentarios que acabamos de introducir es que se te olvide cerrar un comentario que abriste. En este caso el compilador de Java simplemente va a ignorar todo lo que se encuentre entre tu inicio de comentario y el primer fin de comentario que se encuentre. Si no encuentra ninguno dará un mensaje de que se le acabó el archivo sin encontrar el fin de la clase. Algunos paran sin decir nada. En general, si un archivo no trivial de Java compila a la primera es conveniente que revises tus comentarios, a ver si todos están cerrados.

Javadoc

El segundo tipo de comentario que deseamos mencionar es el que se utiliza para que los proyectos de Java estén “auto documentados”, esto es, que la documentación se pueda generar a partir del código mismo. Javadoc es una aplicación del lenguaje Java que permite documentar nuestros proyectos al mismo tiempo que los estamos codificando, de manera sencilla y directamente en el código. A partir de los comentarios que escribas con el formato especificado por Javadoc puedes ejecutar en tu ambiente de programación un programa que convierte esos comentarios en una página (en `html`) que podrás ver directamente en tu navegador y que describa las interfaces o clases que codifiques y documentes.

Los comentarios para Javadoc empiezan con `/**` y terminan con la primer pareja `*/` que encuentren en el código. Se acostumbra ponerlos a cada uno de ellos en un renglón solos para que los distingas mejor pero no importa realmente. No importa lo que estés documentando, tienen siempre al principio una breve descripción de lo que estás comentando, que debe terminar con un punto (.) y al que puedes seguir con una descripción más extensa que también debe terminar con punto. El comentario lo colocas inmediatamente antes de lo que se quiere documentar. Por ejemplo, si está documentando una clase o interfaz, lo colocas antes del encabezado. Si se trata de un método público también colocas el comentario de Javadoc antes del encabezado del método y lo mismo si se trata de atributos de acceso público.

Dependiendo de lo que se está documentando tienes algunas anotaciones que se agregan y que vas a usar en tu proyecto. Empiezan con una arroba, le sigue una etiqueta y a continuación lo que describe la anotación. Puedes intercalar en esta documentación código de `html`, aunque versiones más recientes de Javadoc ya no lo requieren.

Anotaciones posibles para Javadoc

Si se trata de una clase o interfaz, **además de la descripción de la misma**, tenemos las siguientes anotaciones que puedes usar. Cada una de ellas empieza con una arroba (`@`), le sigue inmediatamente una etiqueta y a continuación la documentación de la misma. Dependiendo de la etiqueta de que se trate, Javadoc espera una determinada documentación.

Clase o interfaz En una clase o interfaz, además de la descripción de la misma, nos interesa:

@author Le sigue el nombre del autor.

@version La versión de la clase o interfaz y posiblemente una fecha.

Métodos públicos Además de la descripción del método:

@param Le sigue el nombre del parámetro y cómo se usa. Se coloca uno por cada parámetro que tenga el método. No aparece si no hay parámetros.

@return Se describe el valor que regresa el método. No aparece si el método no regresa nada (tiene como valor de regreso **void**).

En el caso de los métodos, si la documentación de Javadoc no coincide con lo especificado en Java, la documentación de `html` no se generará. Por ejemplo, si el nombre de los parámetros no coincide.

Atributos públicos Se escribe uno por renglón y antes del identificador un breve descripción de para qué se usa.

Si documentas tu interfaz con estos comentarios (que debimos ir haciendo conforme codificábamos, pero ocupaba demasiado lugar y nos hubiésemos tardado más) queda como se ve en el archivo `ServiciosDisco.java`, que puedes cargar desde esta misma página.