salesianos
CÁDIZ

Entornos de Desarrollo

UD6. Optimización y Documentación
(3. Documentación)

Índice

Introducción

- **Necesidad que se aprecia** cuando hay **errores** que **reparar** o hay que **extender** el **programa** con nuevas capacidades o adaptarlo a un nuevo escenario.
- **Reglas** que no se deben olvidar nunca:
  - **Todos** los **programas tienen errores** y descubrirlos sólo es cuestión de tiempo y de que el programa tenga éxito y se utilice frecuentemente
  - **Todos** los **programas sufren modificaciones** a lo largo de su vida, al menos todos aquellos que tienen éxito
- Todo programa que tenga éxito será modificado en el futuro, bien por el programador original, bien por otro programador que le sustituya.
- **¿Para qué sirve documentar el código?**
- Para **explicar** su **funcionamiento**, de forma que cualquier persona que lea la documentación pueda entender la finalidad del código.

Introducción

- **¿A quiénes interesa el código fuente?**
  - Autores del propio código
  - Otros desarrolladores del proyecto
  - Clientes de la API del proyecto
- **¿Por qué documentar el código fuente?**
  - Mantenimiento
  - Reutilización
- **Documentar código** → añadir suficiente información para explicar lo que hace el código, de forma que las personas entiendan qué están haciendo y por qué.

Introducción

- Más vale que **sobre documentación** a **que falte**. Pero con cautela, ya que cuando un programa se modifica, a la vez, deben modificarse la documentación para que no quede obsoleta y acabe refiriéndose a un código que ya no existe.

¿Qué hay que documentar?

- Añadir explicaciones a todo lo que no es evidente.
- Funcionalidad general del código
- ¿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?
- ¿cuál es el uso esperado de una variable?
- ¿qué algoritmo estamos usando? ¿de dónde lo hemos sacado?
- ¿qué limitaciones tiene el algoritmo?
- ¿qué se debería mejorar ... si hubiera tiempo?

Comentarios

- **javadoc**
- Se utiliza para generar documentación externa.

```
/**
 *
 * Descripción principal (texto / HTML)
 *
 *
 * Tags (texto / HTML)
 *
 *
 */
```

- **Una línea**
- Caracteres "//" y terminan con la línea. Se utiliza para documentar código que no necesitamos que aparezca en la documentación externa (que genere javadoc).
- **Tipo C**
- Caracteres "/*", se pueden prolongar a lo largo de varias líneas (que probablemente comiencen con el carácter "*") y terminan con los caracteres "*/". Mantenimiento de código obsoleto “por si acaso”.

Javadoc: documentación de APIs

- Generar documentación de una API **a mano** es tedioso y propenso a errores
- El paquete de desarrollo Java incluye una herramienta, **javadoc,** para generar un conjunto de páginas web a partir de los ficheros de código.
- Contrato entre el programador de la clase/método y el usuario, siendo la forma de referencia más rápida para ver que **funcionalidades ofrece una clase, paquete o interfaz.**
- Javadoc realza algunos comentarios, de los que exige una sintaxis especial. Deben comenzar por "/**" y terminar por "*/", incluyendo una descripción y algunas etiquetas especiales

MarkDown

- Markdown, es una forma sencilla de dar otro aspecto a la documentación utilizando lenguaje de texto plano. Es decir, que con sencillos símbolos de texto plano, podremos definir un texto en negrita, en cursiva, como encabezado, crear un enlace, etc, sin necesidad de utilizar un editor de texto más complejo.

- Existen una serie de elementos básicos que podemos utilizar para dar formato a nuestra documentación: cursiva, negrita, encabezado, enlaces, imágenes…

- **NOTA1**: Visor online Markdown -> https://dillinger.io/
- **NOTA2**: Extensión VSC -> Markdown Preview Enhanced

MarkDown (cursiva y negrita)

- Para poner un texto en *cursiva*, es necesario incluir dicho texto entre símbolos de subrayado: _textoEnCursiva_
  - En este párrafo, _este texto está en cursiva_.
- Para poner un texto en **negrita**, es necesario incluir dicho texto entre dobles asteriscos:
- **textoEnNegrita**
  - En este párrafo, **este texto está en negrita**.

NOTA: Es posible mezclar ambos elementos

MarkDown (encabezados)

- Para poner un texto en forma de encabezado, es necesario preceder dicho texto (la línea completa) por tantas almohadillas como queramos (de 1 a 6):

  #textoEncabezadoTipo1

  ##textoEncabezadoTipo2

  …

  ######textoEncabezadoTipo6

NOTA1: No es posible poner un encabezado en negrita, pero sí poner parte del encabezado (o todo) en cursiva.

NOTA2: Este elemento es similar a los H1, H2, … de HTML

MarkDown (enlaces)

- En MarkDown existen dos tipos de enlaces (aunque ambos funcionan de igual forma):
  - **Enlace en línea**: para hacerlo hay que encerrar el texto del enlace entre corchetes y después el enlace entre paréntesis

    [textoDelEnlace](direcciónDelEnlace)

    NOTA1: Es posible combinar este elemento con los anteriores.

    NOTA2: Es posible crear links automáticos metiendo la url entre los símbolos de menor y mayor: <URLdelEnlace>

MarkDown (enlaces)

- En MarkDown existen dos tipos de enlaces (aunque ambos funcionan de igual forma):
  - **Enlace de referencia**: se usa para hacer referencia a otro lugar del documento. Se usan para que varios enlaces, al mismo sitio, se definan en un solo sitio.

```
Aquí hay [un enlace a otro lugar][otro lugar].
Aquí hay [otro enlace más][otro-enlace].
Y de nuevo regresamos al [primer enlace][otro lugar].

[otro lugar]: https://www.github.com
[otro-enlace]: https://www.google.com
```

MarkDown (imágenes – enlaces)

- En MarkDown los enlaces a imágenes se configuran de la misma forma que un enlace simple pero anteponiendo un símbolo de exclamación al texto alternativo:

```
![logoDeMarkdown](https://upload.wikimedia.org/wikipedia/commons/4/48/Markdown-mark.svg)
```

MarkDown (citas)

- Una cita es una frase o párrafo a la que se la ha dado un formato específico para llamar la atención del lector.
- En MarkDown los se crean anteponiendo a la línea donde está el texto de la cita el símbolo >:

>Aquí ponemos el texto para darle el formato de cita.
>
>>Aquí ponemos el texto de la cita anidada

NOTA1: Si la cita ocupa varios párrafos hay que ponerle el símbolo “>” al comienzo de todos los párrafos.

NOTA2: Es posible mezclarlo con otros elementos.

NOTA3: Para crear citas anidadas se puede usar el “>>”

MarkDown (listas)

- En MarkDown existen 2 tipos de listas:
  - Sin orden: se antepone a cada línea un asterisco, guión (segundo nivel) o más (tercer nivel)

    * Primer elemento
    * Segundo elemento

  - Ordenadas: se antepone a cada línea su número de línea y un punto (pueden combinarse con las desordenadas)

    1. Primer elemento
    2. Segundo elemento

    NOTA: Es posible mezclarlo con otros elementos.

MarkDown (párrafos)

- En MarkDown para dar formato a un párrafo se ponen dos espacios al final de cada párrafo (los .. son dos espacios)

  Primer párrafo..
  Segundo párrafo..

NOTA2: Es posible mezclarlo con otros elementos.

MarkDown (bloque de código)

- En MarkDown para introducir bloques de código hay que encerrar el código entre dos líneas formadas por 3 virgulillas (~) o con ```

~~~
Bloque de código
~~~

```
Bloque de código
```

MarkDown (reglas horizontales)

- En MarkDown para crear una regla horizontal que nos permita separar secciones, de una forma visual, crearemos una línea compuesta por 3 de los siguientes elementos: asteriscos, guiones o guiones bajos

```
Sección 1
***
Sección 2
---
Sección 3

---
Sección 4
```

MarkDown (escape de caracteres)

- En MarkDown para poder escribir aquellos caracteres que se usan para dar formato, hay que anteponerles la barra invertida “\”

\-
*
_

MarkDown (Tablas)

- En MarkDown para poder crear tablas tenemos que crear la estructura de la siguiente forma:

- | Left columns | Right columns |
- | ------------ |:-------------:|
- | left foo | right foo |
- | left bar | right bar |
- | left baz | right baz |