

Tecnología de la Programación

Documentación

Departamento de Computación

Facultad de Informática
Universidade da Coruña

Curso 2006/2007

- Características de la documentación:
 - Comentarios malos, peor que no tener comentarios.
 - Documentación obsoleta, peor que no tener documentación.
 - Deseable: comentarios breves y precisos.
 - Lo que expresa el código no necesita comentario.
 - Incrustar la documentación en el código.
 - Automatizar con herramientas.
 - Más fácil que se actualize.
- Anotación = documentación.

- Características de la documentación:
 - Comentarios malos, peor que no tener comentarios.
 - Documentación obsoleta, peor que no tener documentación.
 - Deseable: comentarios breves y precisos.
 - Lo que expresa el código no necesita comentario.
 - Incrustar la documentación en el código.
 - Automatizar con herramientas.
 - Más fácil que se actualize.
- Anotación = documentación.

Documentación: contraejemplos (I)

```
/*
 * Start of function S Q R T (= "square root")
 * S Q R T takes an argument :
 * - double X: the number to be squared (X >= 0)
 * S Q R T return:
 * - the square root of X
 * (c) 2003 H. Becker, Burbach - all rights reserved
 */
```

- Más importancia al aspecto que al contenido.
- Engorroso de mantener el aspecto.
- Muy “verboso”, se oculta la info. importante.
- Alternativa:

$\{X \geq 0\}$

```
function raiz_cuadrada(float X):float { ... }
```

$\{\text{return} = \text{sqrt}(X)\}$

Documentación: contraejemplos (II)

```
n += 1;      // Incrementar n
a *= n;      // Multiplicar a por n
return a;    // Devolver el valor de a
```

```
int g;       // Número de gallifantes
g = num_g(); // Calcular el número de gallifantes
assert(g == 2); // Esperamos dos gallifantes
```

- **Alternativa:**

```
int num_gallifantes;
num_gallifantes = numero_de_gallifantes();
assert(num_gallifantes == 2);
```

Documentación: JAVADOC (I)

- Herramienta para Java.
- Extrae documentación del código.
- Documentación = comentarios con sintáxis especial.
 - Comienzan por “/ **”.
 - Incluye etiquetas que comienzan por “@”.
 - @param
 - @return
 - ...
- Genera HTML.
- Admite etiquetas HTML.

Documentación: JAVADOC (I)

- Herramienta para Java.
- Extrae documentación del código.
- Documentación = comentarios con sintáxis especial.
 - Comienzan por “/ **”.
 - Incluye etiquetas que comienzan por “@”.
 - @param
 - @return
 - ...
- Genera HTML.
- Admite etiquetas HTML.

Documentación: JAVADOC (I)

- Herramienta para Java.
- Extrae documentación del código.
- Documentación = comentarios con sintáxis especial.
 - Comienzan por “/ **”.
 - Incluye etiquetas que comienzan por “@”.
 - @param
 - @return
 - ...
- Genera HTML.
- Admite etiquetas HTML.

Documentación: JAVADOC (I)

- Herramienta para Java.
- Extrae documentación del código.
- Documentación = comentarios con sintáxis especial.
 - Comienzan por “/ **”.
 - Incluye etiquetas que comienzan por “@”.
 - @param
 - @return
 - ...
- Genera HTML.
- Admite etiquetas HTML.

Documentación: JAVADOC (I)

- Herramienta para Java.
- Extrae documentación del código.
- Documentación = comentarios con sintáxis especial.
 - Comienzan por “/ **”.
 - Incluye etiquetas que comienzan por “@”.
 - @param
 - @return
 - ...
- Genera HTML.
- Admite etiquetas HTML.

Documentación: JAVADOC (II)

```
$ vi HolaMundo.java
/**
 * El mítico primer ejemplo.
 */

public class HolaMundo
{
    public static int main(String argv[]) {
        System.out.println(";Hola mundo!\n");
    }
}
```

```
$ javadoc HolaMundo.java
```

```
...
```

```
$ ls
```

```
HolaMundo.html
```

```
HolaMundo.java
```

```
allclasses-frame.html
```

```
allclasses-noframe.html
```

```
constant-values.html
```

```
deprecated-list.html
```

```
help-doc.html
```

```
index-all.html
```

```
index.html
```

```
overview-tree.html
```

```
package-list
```

```
packages.htm
```

```
resources
```

```
stylesheet.c
```

- Algunas etiquetas:

`@author` Autor/es de la clase.

`@version` Versión de la clase.

`@deprecated` Marca métodos como deprecated.

`@since` Cuando se incluyo el elemento.

`@see`, `@link` Referencia a otro elemento.

`@param` Parámetro del método.

`@return` Valor de retorno del método.

`@exception` Excepción que puede arrojar el método.

- Podemos crear nuevas etiquetas para anotaciones.
- Problema con anotaciones: fórmulas lógicas.
- Opciones:
 - Usar MathML.
 - Generar \LaTeX , PDF, Postscript.

- Nuevas etiquetas (simple):

```
javadoc -tag reviewed:m:"Reviewed by:"
```

- Nuevas etiquetas con Taglets (clase java):

```
public class TodoTaglet implements Taglet {  
    ...  
    public static void register(Map tagletMap) { ... }  
  
    public String toString(Tag tag) { return "<b>" + ...  
javadoc -taglet TodoTaglet *.java
```

- Formato de salida diferente: Doclets

```
javadoc -doclet ListClass -docletpath . *.java
```

- DOXYGEN: C, C++, IDL, Java.
- OCAMLDOC: caml.
- PERLDOC: Perl.
- ...
- Literate Programming:
 - Extraer el programa de la documentación.
 - WEB (T_EX).