

# Documentando con Javadoc

## Introducción

Diseño de Sistemas Informáticos 2007/08

MADS Group - Departamento de Computación

Marzo de 2008

Contenido

Introducción

Tags Javadoc

Extensión de Javadoc

Anotaciones

Bibliografía

- 1 **Introducción**
- 2 **Tags Javadoc**
- 3 **Extensión de Javadoc**
- 4 **Anotaciones**
- 5 **Bibliografía**

## Contenido

Introducción

Tags Javadoc

Extensión de Javadoc

Anotaciones

Bibliografía

- ¿A **quienes** interesa el código fuente?
  - Autores del propio código
  - Otros desarrolladores del proyecto
  - Clientes de la API del proyecto
- ¿**Por qué** documentarlo?
  - Mantenimiento
  - Reutilización
- ¿**Qué** documentar?
  - Obligatorio  $\left\{ \begin{array}{l} \text{Clases y paquetes} \\ \text{Constructores, métodos y atributos} \end{array} \right.$   
`/** Javadoc */`
  - Conveniente  $\left\{ \begin{array}{l} \text{Fragmentos no evidentes} \\ \text{Bucles, algoritmos...} \end{array} \right.$   
`// una sola línea`  
`/* varias líneas */`

- Generar documentación de una API a mano es tedioso y propenso a errores
    - Gran cantidad de pequeños detalles
    - Sincronización de código fuente y documentación
    - Duplicidad de esfuerzos (tipos, nombres...)
- ⇒ Combinar código fuente con documentación
- + Generar documentación desde el código
  - + La documentación embebida en el código tiende a ser más correcta

## Javadoc

- Genera documentación en HTML
- Usa la información de nombres, tipos...
- Explicaciones adicionales y referencias cruzadas
- Otras herramientas se apoyan en Javadoc para ayudar a los desarrolladores (p.e. Eclipse)

- Comentarios con una sintaxis concreta, que se ubican antes de las clases, interfaces, constructores, metodos y atributos a documentar

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

Contenido

Introducción

Tags Javadoc

Extensión de Javadoc

Anotaciones

Bibliografía

```
/**
 * Returns the index of the first occurrence of the specified element in
 * this vector, searching forwards from index, or returns -1 if
 * the element is not found.
 *
 * @param o element to search for
 * @param index index to start searching from
 * @return the index of the first occurrence of the element in
 *         this vector at position index or later in the vector;
 *         -1 if the element is not found
 * @throws IndexOutOfBoundsException if the specified index is negative
 * @see   Object#equals(Object)
 */
public int indexOf(Object o, int index) ...
```

## indexOf

```
public int indexOf(Object o,
                  int index)
```

Returns the index of the first occurrence of the specified element in this vector, searching forwards from `index`, or returns -1 if the element is not found.

### Parameters:

`o` - element to search for  
`index` - index to start searching from

### Returns:

the index of the first occurrence of the element in this vector at position `index` or later in the vector; -1 if the element is not found.

### Throws:

[IndexOutOfBoundsException](#) - if the specified index is negative

### See Also:

[Object.equals\(Object\)](#)

- La primera frase de cada comentario Javadoc debe ser una frase resumen con una **descripción concisa y completa**, terminada en punto, y seguida de un espacio, tabulador o retorno de carro
- Usar la **etiqueta** `<code>` para palabras clave y nombres
- Preferible el uso de la **tercera persona**

\* Devuelve el índice del primer elemento...

\* Devolvemos el índice del primer elemento... ← Evitar

- Empezar con un **verbo** la descripción de los métodos
- **Omitir el sujeto** cuando es obvio

\* @param peer nombre del peer

\* @param peer parámetro con el nombre del peer ← Evitar

- Palabras clave gestionadas de forma especial
- Dos tipos,
  - **Block tags**
    - Ubicados después de la descripción principal
    - @tag
  - **Inline tags**
    - Ubicados en la descripción principal o en las descripciones de los *block tags*
    - {@tag}

```
/**  
 * ...  
 * @param id hash del objeto, calculado según {@link #hash(Object)}  
 * ...  
 */
```

- @param, @return, @throws, @author, @version, @see, @since...
  - *Case sensitive*



- Clases, interfaces, constructores, métodos, atributos  
⇒ \*.java
- Paquetes  
⇒ package.html
- Generación  $\left\{ \begin{array}{l} \text{javadoc *.java} \\ \text{Ant} \end{array} \right.$

Contenido

Introducción

Tags Javadoc

Extensión de Javadoc

Anotaciones

Bibliografía

- Aplicable a parámetros de constructores y métodos
- Describe los parámetros del constructor/método
- name: idéntico al nombre del parámetro

```
/**
 * Removes from this List all of the elements whose index is between
 * fromIndex, inclusive and toIndex, exclusive. Shifts any succeeding
 * elements to the left (reduces their index).
 * This call shortens the ArrayList by (toIndex - fromIndex) elements. (If
 * toIndex==fromIndex, this operation has no effect.)
 *
 * @param fromIndex index of first element to be removed
 * @param toIndex index after last element to be removed
 */
protected void removeRange(int fromIndex, int toIndex) {
    ...
}
```

Contenido

Introducción

Tags Javadoc

Extensión de Javadoc

Anotaciones

Bibliografía

- Aplicable a métodos
- Describe el valor de retorno del método
- Incluir descripción de valores de retorno especiales
  - `null`
  - ...

```
/**
 * Tests if this vector has no components.
 *
 * @return <code>>true</code> if and only if this vector has
 *         no components, that is, its size is zero;
 *         <code>>false</code> otherwise.
 */
public boolean isEmpty() {
    return elementCount == 0;
}
```

Contenido

Introducción

Tags Javadoc

Extensión de Javadoc

Anotaciones

Bibliografía

- Aplicable a constructores y métodos
- Describe las posibles excepciones del constructor/método
- Un @throws por cada posible excepción
- Si es de ayuda para el usuario, también se pueden documentar las *unchecked exceptions*

```
/**
 * Parses the string argument as a signed decimal
 * <code>long</code>. The characters in the string must all be
 * decimal digits, except that...
 *
 * @param s a <code>String</code> containing the <code>long</code>
 * representation to be parsed
 * @return the <code>long</code> represented by the argument in
 * decimal.
 * @exception NumberFormatException if the string does not contain a
 * parsable <code>long</code>.
 */
public static long parseLong(String s)
    throws NumberFormatException {
    ....
}
```

Contenido

Introducción

Tags Javadoc

Extensión de Javadoc

Anotaciones

Bibliografía

- Aplicable a clases, interfaces, constructores, métodos, atributos y paquetes
- Añade enlaces de referencia a otras partes de la documentación
- Variantes,
  - @see string
  - @see <a href="URL">label</a>
  - @see package.class#member label

```
/**
 *
 * ...
 *
 * @see "Design Patterns: Elements of Reusable OO Software"
 * @see <a href="http://www.w3.org/WAI/">Web Accessibility Initiative</a>
 * @see String#equals(Object) equals
 */
public void method() {
    ...
}
```

# {@link package.class#member label}

- Aplicable a clases, interfaces, constructores, métodos, atributos y paquetes
- Equivalente a @see package.class#member label
- *Inline tag*
- No genera sección de referencias

```
/**
 * Returns the component at the specified index.
 *
 * This method is identical in functionality to the {@link #get(int)}
 * method (which is part of the {@link List} interface).
 *
 * @param index an index into this vector
 * @return the component at the specified index
 * @throws ArrayIndexOutOfBoundsException if the index is out of range
 *         (<code>index < 0 || index >= size()</code>)
 */
public Object elementAt(int index) {
    ...
}
```

- Copiado de documentación { Implícito  
Explícito
- **Implícito** (automático)
  - Cuando en un comentario Javadoc de un método no se incluye una descripción general o un *tag* `@return`, `@param` y/o `@throws`, la herramienta de generación de la documentación toma la descripción o el *tag* de la clase o interfaz de nivel superior
  - Para el caso del tag `@throws` solo se copia el *tag* de la superclase si se trata de una *checked exception*
- **Explícito**
  - ⇒ {`@inheritDoc`}
  - Aplicable a clases, interfaces, constructores y métodos
  - *Inline tag*

Contenido

Introducción

Tags Javadoc

Extensión de Javadoc

Anotaciones

Bibliografía

- Copia la documentación del elemento de nivel superior inmediato
  - ⇒ Permite crear documentación más general en las superclases y completarla en las subclases escribiendo alrededor del texto heredado

```
/**
 * (superclase) ...
 *
 * @throws NullPointerException if <code>dst</code> is <code>>null</code>
 */
public void getChars(int srcBegin, int srcEnd, char dst[], int dstBegin) {
    ....
}
```

```
/**
 * (subclase) ...
 *
 * @throws NullPointerException {@inheritDoc}
 */
public void getChars(int srcBegin, int srcEnd, char dst[], int dstBegin) {
```



- Nuevos *tags* y salidas alternativas
- **Doclets**
  - `com.sun.javadoc.*`
  - Definen el contenido y tipo de salida de la herramienta Javadoc
  - `-doclet`
  - UMLGraph (UMLGraphDoc), PDF Doclet...
- **Taglets**
  - `com.sun.tools.doclets.Taglet`
  - Permiten incorporar nuevos *block* e *inline tags* a los ya existentes en Javadoc
  - `-tag`

## Contenido

Introducción

Tags Javadoc

Extensión de Javadoc

Anotaciones

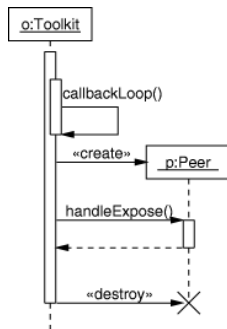
Bibliografía

- *“UMLGraph allows the declarative specification and drawing of UML class and sequence diagrams”*

```
# Define the objects
object(0,"o:Toolkit");
placeholder_object(P);
step();

# Activation and messages
active(0);
message(0,0,"callbackLoop()");
create_message(0,P,"p:Peer");
message(0,P,"handleExpose()");
active(P);
return_message(P,0,"");
inactive(P);
destroy_message(0,P);
inactive(0);

# Complete the lifeline of 0
step();
complete(0);
```



Contenido

Introducción

Tags Javadoc

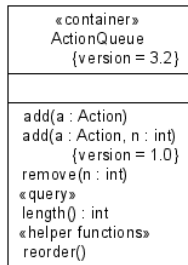
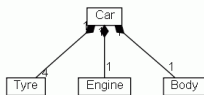
Extensión de Javadoc

Anotaciones

Bibliografía

```
class Tyre {}
class Engine {}
class Body {}
/**
 * @composed 1 - 4 Tyre
 * @composed 1 - 1 Engine
 * @composed 1 - 1 Body
 */
class Car {}
```

```
/** @hidden */
class Action {}
/**
 * @stereotype container
 * @tagvalue version 3.2
 */
class ActionQueue {
    void add(Action a) {};
    /** @tagvalue version 1.0 */
    void add(Action a, int n) {};
    void remove(int n) {};
    /** @stereotype query */
    int length() {};
    /** @stereotype "helper functions" */
    void reorder() {};
}
```



## Contenido

## Introducción

## Tags Javadoc

## Extensión de Javadoc

## Anotaciones

## Bibliografía

- Introducidas en 1.5
- Metadatos del código fuente
- Posibles usos,
  - Información para el compilador
  - Información para herramientas de procesado
  - Procesamientos en tiempo de ejecución
- Aplicables a clases, atributos, métodos. . .

```
/**  
 * @deprecated explanation of why it was deprecated  
 */  
@Deprecated  
public void deprecatedMethod() {  
    ...  
}
```

## Contenido

Introducción

Tags Javadoc

Extensión de Javadoc

Anotaciones

Bibliografía

```
import java.lang.annotation.*; // import this to use @Documented and @Retention

@Documented // Make the annotation appear in Javadoc generated doc
@Retention(RetentionPolicy.RUNTIME) // Make annotation available at runtime
@interface ClassPreamble {
    String author();
    String date();
    int currentRevision() default 1;
    String lastModified() default "N/A";
    String lastModifiedBy() default "N/A";
    String[] reviewers(); // Note use of array
}
```

```
@ClassPreamble (
    author = "John Doe",
    date = "3/17/2002",
    currentRevision = 6,
    lastModified = "4/12/2004",
    lastModifiedBy = "Jane Doe"
    reviewers = {"Alice", "Bob", "Cindy"} // Note array notation
)
public class Generation3List extends Generation2List {
    ...
}
```

Contenido

Introducción

Tags Javadoc

Extensión de Javadoc

Anotaciones

Bibliografía

- [apt08] Annotation Processing Tool.  
<http://java.sun.com/j2se/1.5.0/docs/guide/apt/>, 2008.
- [jav08] Javadoc 5.0 Tool.  
<http://java.sun.com/j2se/1.5.0/docs/guide/javadoc/>, 2008.
- [uml08] UMLGraph - Declarative Drawing of UML Diagrams.  
<http://www.umlgraph.org>, 2008.
- [wri08] How to Write Doc Comments for the Javadoc Tool.  
<http://java.sun.com/j2se/javadoc/writingdoccomments/>, 2008.

Contenido

Introducción

Tags Javadoc

Extensión de Javadoc

Anotaciones

Bibliografía