

Breve introduzione al Javadoc

Premessa

- Javadoc è uno strumento che permette di documentare i sorgenti di un programma all'interno dei sorgenti stessi.
- Anziché scrivere la documentazione di un programma in un file separato, il programmatore **inserisce** nel codice sorgente dei commenti **in un formato particolare**.
- Tali commenti vengono estratti dal programma javadoc che li converte in un formato più semplice per la consultazione (HTML o PDF, ad esempio)
- In questa breve guida, daremo alcune direttive di come scrivere dei commenti Javadoc minimi.

Breve introduzione al Javadoc

Formato di un Javadoc

In generale un commento Javadoc è un testo **HTML** racchiuso tra i tag **/**** e ***/**.

Esempio Javadoc

```
/**  
 * Questo &egrave; un commento Javadoc.  
 * Gli spazi e gli asterischi a inizio riga  
 * sono sempre ignorati.  
 */
```

L'effetto è il seguente:

Questo è un commento *Javadoc*. Gli spazi e gli asterischi a inizio riga **sono** sempre ignorati.

Breve introduzione al Javadoc

Cosa si commenta?

- In generale, si dovrebbe commentare tutto ciò che utile per far comprendere il proprio codice ai colleghi e, per gli aspetti più generali, agli utenti.
- Un commento Javadoc è pensato solo per descrivere le funzionalità o i principi di un package/classe/interfaccia/metodo.
- Un commento Javadoc non è pensato per “spiegare” pezzi di codice. Questi commenti, anche se sono in formato Javadoc, non sono mai estratti dal comando javadoc.
- La documentazione **minima** dovrebbe comprendere la descrizione di ciascuna **package**, **classe**, **interfaccia**, **metodo pubblico**, **attributo pubblico**.

Breve introduzione al Javadoc

Come si commenta?

Un commento Javadoc:

- è sempre posto subito prima della dichiarazione della **classe**, **interfaccia**, **metodo**, **attributo**.
- deve descrivere in modo sintetico lo scopo dell'oggetto che si sta documentando.
- può contenere tag HTML per aiutare la formattazione, anche se si sconsiglia di adottare tag di *presentazione* come ``, `<i>`, ecc in favore di tag di *struttura* come ``, ``, ecc
- può contenere tag speciali (prossime slide) che aiutano a organizzare meglio il commento.
- Un commento Javadoc non deve essere una lezione. Deve aiutare a comprendere e non far perdere tempo.

Breve introduzione al Javadoc

Esempio di Javadoc

Ecco un esempio completo

```
/**
 * Returns an Image object that can then be painted on the screen.
 * The url argument must specify an absolute {@link URL}. The name
 * argument is a specifier that is relative to the url argument.
 * <p>
 * This method always returns immediately, whether or not the
 * image exists. When this applet attempts to draw the image on
 * the screen, the data will be loaded. The graphics primitives
 * that draw the image will incrementally paint on the screen.
 *
 * @param url an absolute URL giving the location of the image
 * @param name the location of the image
 * @return the image at the specified URL
 * @see Image
 */
public Image getImage(URL url, String name);
```

Breve introduzione al Javadoc

Esempio di Javadoc

Il commento precedente è composto da

- Una prima parte composta da un paragrafo che riassume lo scopo del metodo.
- Eventuali paragrafi successivi sono suddivisi da `<p>`.
- Una riga vuota che separa la parte descrittiva dalla parte dei tag Javadoc. Deve esserci solo 1 riga vuota in tutto.
- Una seconda parte in cui sono inseriti tag Javadoc (ad esempio, `@param`, spiegati nelle prossime slide) che identificano le componenti dell'oggetto che si sta commentando.

Breve introduzione al Javadoc

I tag

- Formato generale di un tag: **@name comment**
dove nome specifica quale tipo di informazione si sta dando e il commento è l'informazione.
Esempio: **@author William Shakespeare**
- Ogni tag deve essere su una riga nuova.
- I commenti possono estendersi su più righe, ma non ci devono essere righe vuote!

Breve introduzione al Javadoc

I tag per le classi

I tag che si devono mettere sono:

@author chi ha scritto il programma. Formato: Nome
Cognome (no nick!)

Se ci sono più autori, si mettono più tag su righe separate, uno di seguito all'altro.

Tag minimo per le classi

```
/** ...  
 *  
 * @author William Shakespeare  
 * @author Christopher "Kit" Marlowe  
 */  
public class Drama {...}
```


Breve introduzione al Javadoc

I tag per i metodi

I tag che si devono mettere sono (in ordine di apparizione):

@param <nome parametro> breve descrizione del parametro. Se ci sono più parametri, rispettare l'ordine con cui sono dichiarati.

@return breve descrizione di ciò che è tornato dal metodo.
Esempio: `@return true if the value was found in the array, false otherwise.`

@exception <nome eccezione> descrizione delle circostanze che determinano il lancio dell'eccezione.

Breve introduzione al Javadoc

I tag per i metodi

Tag minimo per i metodi

```
/**
 * Gets item.
 *
 * @param item searched element
 * @return the node that contains item
 *         if it is present, null otherwise.
 * @exception NullPointerException if item
 *         is null.
 */
BinaryNode get(Comparable item);
```

Breve introduzione al Javadoc

I tag per le variabili/attributi

Non ci sono tag speciali per descrivere le variabili.

Breve introduzione al Javadoc

Come si compila

- Il comando per estrarre la documentazione è `javadoc` presente in tutte le Java Development Kit (JDK)
- Il comando `javadoc` può produrre la documentazione in diversi formati, ciascuno dei quali può richiedere dei file di istruzioni speciali (i cosiddetti `doclet`)
- Il formato standard è HTML e non richiede nessun file ausiliario.

Breve introduzione al Javadoc

Come si compila

- Si supponga di voler ottenere la documentazione di un proprio package (labASD) i cui sorgenti sono nella directory `$HOME/labASD/src/`.
- Si supponga di voler salvare la documentazione (che sarà composta da più file HTML) nella directory `$HOME/labASD/doc/`.
- Si supponga di voler estrarre la documentazione di tutti i metodi/attributi (anche i privati).
- Ci si posizioni nella directory `$HOME/labASD/`.
- Il comando da lanciare è il seguente:

```
javadoc -private -sourcepath ./src -d ./doc  
labASD
```

Breve introduzione al Javadoc

Come si compila

- È quasi inevitabile che nei propri sorgenti si faccia riferimento a classi del package `java.lang`, `java.util`.
- Se si compila il javadoc con le opzioni della precedente slide, i riferimenti a tali classi saranno scritti per esteso ma senza alcun collegamento

Esempio: `java.lang.Object get(int k)`

- Se, viceversa, si vuole avere un link alla documentazione Javadoc di tali classi, è necessario specificare l'indirizzo base del loro Javadoc

Esempio: Object `get(int k)`

- Per i package standard (`java.*`) è sufficiente specificare l'ulteriore parametro:

`-link http://java.sun.com/javase/6/docs/api/` (per la versione 6 della JDK)

Breve introduzione al Javadoc

Come si compila... in PDF

- Si supponga di voler la documentazione nel formato PDF.
- È necessario specificare una doclet in grado di istruire Javadoc come produrre un PDF.
- Tra le molteplici disponibili, la doclet PdfDoclet <http://pdfdoclet.sourceforge.net/> è una delle più semplici e complete.
- Una volta scaricata la libreria (pdfdoclet-1.0.2-all.jar), è sufficiente specificare la libreria tra le opzioni del comando:

```
javadoc -private -sourcepath ./src \  
-doclet com.tarsec.javadoc.pdfdoclet.PDFDoclet \  
-docletpath ./pdfdoclet-1.0.2-all.jar \  
-pdf <nome file pdf> labASD
```
- Per ulteriori opzioni si rimanda alla documentazione di PdfDoclet.