



L'outil Javadoc

un générateur de documentation

1



...\bin\javadoc.exe

- un générateur de documentation
- génère une série de pages web décrivant l'API de classes Java dont le code source comporte des commentaires spéciaux
- Le plus souvent: classes public, méthodes public, ...

XH

2

Résultat un site web

The screenshot shows the Java API documentation for the `Adresse` class. The interface includes a sidebar with navigation links (All Classes, Adresse, Etudiant, Personne), a main content area with tabs for Package, Class, Tree, Deprecated, Index, and Help, and a search bar at the bottom. The `Adresse` class is detailed, showing its inheritance from `java.lang.Object`, its version (4), and its author (Turturleon). The `Method Summary` section lists the `formeTextuelle(boolean prefix)` method, which returns a `java.lang.String`. The `Method Detail` section provides the full signature and a description of the method's purpose: to return a textual representation of the address, with a prefix 'F' for postal addresses.

XH

3

Si vous codez des classes

- une documentation est nécessaire
- « contrat » passé avec l'utilisateur de votre code
- décrit les fonctionnalités et leurs conditions d'utilisation

XH

4



Conseils sur les commentaires Javadoc

- Chez Oracle
- [codeconventions-programmingpractices.html](https://www.oracle.com/java/technologies/javase/codeconventions-codeexamples.html)
- Exemple avec la classe Blah
- <https://www.oracle.com/java/technologies/javase/codeconventions-codeexamples.html>
- Chez Google
- <https://google.github.io/styleguide/javaguide.html#s7-javadoc>

XH

5



Format des commentaires Javadoc

- commencent par `/**`, se terminent par `*/` et toutes les lignes intermédiaires commencent par `*`
- écrits en HTML
- précèdent la déclaration d'une classe, d'un attribut ou d'une méthode
 - Pas les variables locales
- composés de 2 parties :
 - une description explicative
 - un bloc de « tags » décrivant des éléments contractuels »

XH

6



Principaux tags pour une classe

- @author
- @version

exemple :

```
/**
 * Cette classe représente une <b>adresse
 * <font color=red>postale</font> </b>
 * @author Tartempion
 * @version 4
 */
```

```
public class Adresse
{
```

XH

7



Principaux tags pour une méthode

- @param nomParametre description du paramètre
- @return description de la donnée retournée par la méthode
- @throws typeException description du cas où cette exception est
générée (ou propagée)
- @see référence à une autre documentation ou à une autre méthode
- @since version du produit où cette méthode est apparue
- @author le nom du développeur

exemple :

```
/**
 * fournit une représentation textuelle de l'adresse
 *
 * @param prefix vrai pour que le code postal soit préfixé par un
 * 'F', faux pour ne pas avoir de préfixe
 * @return une chaîne de caractères représentant l'adresse sous le
 * format habituellement employé sur les enveloppes
 * @throws BadAddressFormat si prefix est vrai et que l'adresse
 * n'est pas en France
 * @since 1.3
 */
public String formeTextuelle(boolean prefix) throws
BadAddressFormat
{
    . . .
```

XH

8



Production Javadoc avec la CLI

- fichiers sources commentés
- Avec la CLI, la commande `javadoc.exe`
- exemples :
 - `javadoc *.java`
 - `javadoc Rectangle.java Cercle.java`
- Voir les options à la page suivante:
 - `javadoc -d doc Rectangle.java`

XH

9



Quelques options javadoc CLI

- **-d** : pour éviter que les pages web soient générées dans le répertoire courant
- **-sourcepath** *path* : pour indiquer où trouver les fichiers source
- **-author** : pour tenir compte des tags « @author »
- **-version** : pour tenir compte des tags « @version »
- **-private** : pour inclure toutes les classes, attributs et méthodes (seuls les public et protected sont inclus par défaut)

XH

10



Production de la Javadoc avec Eclipse

- Dans vos sources, Eclipse peut générer le squelette des commentaires Javadoc :
 - placer le curseur sur la déclaration de méthode ou de classe
 - clic droit, « source », « Generate element comment »
- Pour produire la documentation d'un projet:
 - menu <file><export><Java><Javadoc>
 - Boîtes de dialogue « Javadoc Generation »
 - choisir la destination ..\doc, choisir les classes à documenter, choisir la visibilité « Package »
 - cliquer <Finish>
- Visualisez dans FireFox
 - index.html, et vérifiez que vous n'avez pas de pb avec les accents

XH

11



Autres outils



- www.doxygen.org
- outil semblable à Javadoc, permettant de générer de la documentation non seulement à partir de code source commenté Java mais aussi C, C++, Python... Ses formats de sortie sont également plus variés que javadoc : LaTeX, RTF, PostScript, PDF, ...
- Voir la page Wikipédia
- Il existe des plugins javadoc pour Maven, AntMod...

XH

12



Documenter ET commenter

Attention : documenter le code ne dispense pas de le commenter !

- Documenter :
 - quelles sont les fonctionnalités ?
 - pour un utilisateur externe au projet
- Commenter :
 - comment sont réalisées les fonctionnalités ?
 - pour une personne interne au projet (éventuellement soi-même)

XH

13



END



XH

14