

# Java : Javadoc

**Achref El Mouelhi**

Docteur de l'université d'Aix-Marseille  
Chercheur en Programmation par contrainte (IA)  
Ingénieur en Génie logiciel

[elmouelhi.achref@gmail.com](mailto:elmouelhi.achref@gmail.com)



- 1 Introduction
- 2 Syntaxe
- 3 Javadoc et Eclipse
- 4 Doxygen : installation et génération de documentation

# Javadoc

## Définition

Il s'agit d'un document PDF, HTML ou autre qui permet de décrire le fonctionnement d'une application Java.

# Javadoc

## Définition

Il s'agit d'un document PDF, HTML ou autre qui permet de décrire le fonctionnement d'une application Java.

Ce n'est pas un rapport à rédiger.

# Javadoc

## Définition

Il s'agit d'un document PDF, HTML ou autre qui permet de décrire le fonctionnement d'une application Java.

Ce n'est pas un rapport à rédiger.

C'est plutôt un rapport à générer à partir de notre code-source.

# Javadoc

## Pourquoi ?!

- Avoir une idée sur la structure de l'application
- Permettre de bien comprendre le fonctionnement
- Faciliter la maintenance
- Éviter de lire plusieurs (milliers voire plus) de lignes de code pour comprendre le fonctionnement de l'application

# Javadoc

## Principe

- Bien commenter le code (respecter certaines règles)
- Générer la documentation en faisant l'extraction des commentaires et en établissant les différents liens entre éléments

# Javadoc

## 3 types de commentaires

- Commentaire mono-ligne

```
// commentaire sur une seule ligne
```

- Commentaire multi-ligne

```
/* commentaire sur  
plusieurs lignes */
```

- Commentaire pour annotation et documentation

```
/** commentaire pour  
documentation */
```

# Javadoc

## Règles

- Un commentaire de documentation concerne un élément de notre code source
  - une classe
  - un attribut de classe
  - une variable
  - une méthode
  - ...
- Pas de ligne vide entre un commentaire de documentation et l'élément commenté
- Un commentaire de documentation peut avoir un (ou même plusieurs) attributs (appelés aussi balises)

# Javadoc

## Attributs

- `@author` : permet d'indiquer l'auteur (ou les auteurs) participant au code source

```
/**  
 * @author Wick  
 */  
public class Personne{ }
```

- `@see` : effectue un renvoi vers une autre classe ou fonction...

```
/**  
 * @see Adresse  
 */  
private Adresse adresses[] = new Adresse[5];
```

# Javadoc

## Attributs

- `@version` : permet de définir la version d'un fichier

```
/**  
 * @version 3.2beta  
 */
```

- `@deprecated` : permet d'indiquer que le code marqué est obsolète
- `@class` : permet de décrire une classe
- `@exception` : permet de décrire une classe exception

# Javadoc

## Attributs

- `@return` : permet de décrire la valeur de retour d'une méthode.
- `@throws` : indique qu'une méthode peut lever une exception
- `@param` : permet de lister et décrire les paramètres d'une méthode

```
/**  
 * @param var1 description1  
 * @param ...  
 * @param varN descriptionN  
 */
```

# Javadoc

## Sous Eclipse

- Pour commenter un élément : cliquer sur l'élément et faire alt+shift+j
- Pour générer la Javadoc : aller dans le menu Project et choisir l'option Generate Javadoc

# Javadoc

## Doxygen pour générer la documentation JAVA

- Supporte plusieurs langages de programmations C, C++, Java, C#...
- Plusieurs formats de documentation possible HTML, PDF, XML, LATEX...
- Disponible sous Windows, Mac et Linux
- Même syntaxe ou presque pour les autres langages et pour les autres générateurs

# Javadoc

## Téléchargement

- Télécharger Doxygen :

[http://www.stack.nl/~dimitri/doxygen  
/download.html#latesrc](http://www.stack.nl/~dimitri/doxygen/download.html#latesrc)

- Télécharger Graphviz qui permettra de visualiser certaines propriétés (telles que l'héritage) via des graphes :

<http://www.graphviz.org/>

- Télécharger HTML WorkShop :

[https://www.microsoft.com/en-us/download/  
details.aspx?displaylang=en&id=21138](https://www.microsoft.com/en-us/download/details.aspx?displaylang=en&id=21138)

- Télécharger Miktex **si vous voulez générer de la documentation PDF ou PS** : <https://miktex.org/2.8/setup>

# Javadoc

## Installation et lancement

- Installer tous les fichiers téléchargés
- Lancer `doxywizard.exe`

# Javadoc

## Utilisation : l'onglet Wizard

- Dans le menu Project, il faut préciser les données relatives au projet (répertoire code source, répertoire destination...)
- Dans le menu Mode, on choisit l'option JAVA
- Dans le menu Output, on choisit le format de documentations à générer
- Dans le menu Diagrams, on choisit les types de diagrammes à générer...

# Javadoc

## Utilisation : l'onglet Expert

- Dans le menu **Project**, on peut forcer l'utilisation d'un codage, modifier la langue du document à générer
- Dans le menu **Build**, on peut afficher la documentation pour les fonctions/méthodes/membres ayant le qualificateur `private`, `static`...
- Dans le menu **Input**, ça nous permet de définir le codage utilisé pour nos fichiers sources
- ...

# Javadoc

## Utilisation : l'onglet Run

- Le bouton Run doxygen permet de générer la documentation

# Javadoc

## Plus de détails

- <http://www.stack.nl/~dimitri/doxygen/index.html>