



## Javadoc

# Création de documentation en Java

# Plan

- 1 Introduction à la Javadoc
- 2 Exemples
- 3 Création javadoc : ligne de commande / Eclipse

# Introduction Javadoc

## Le programme javadoc

- **javadoc** est le 3e programme essentiel fourni avec le SDK (java, javac).
- C'est l'outil qui sert à générer la documentation au format de Sun (API).
- Il permet donc à un tiers d'utiliser vos programmes et vos librairies de manière simple et efficace.
- Cela permet aussi au programmeur de s'y retrouver plus rapidement !
- Utiliser cet outil est fondamental pour programmer en Java.

## Les commentaires pour la javadoc : `/** ... */`

- `/**`  
\* documentation pour méthode/classe/attributs/paramètres/etc.  
\*  
`*/`

## Exemple de javadoc

### Personne.java

```
package cours.javadoc;
/**
 * La classe personne modelise une personne.
 * Elle sert pour...
 *
 * @see NegativeAgeException
 * @author fab
 */
public class Personne implements Cloneable{
    /**
     * l'age de la personne, doit etre positif
     */
    private int age;
    /**
     * le nom de la personne, une simple chaine de caracteres
     */
    private String name;
```

# Exemples

## Personne.java

```
/**
 * Constructeur pour la classe Personne.
 * Blabla ...
 *
 * @param age un entier >= 0
 * @param name le nom de la personne
 * @throws NegativeAgeException si age < 0
 */
public Personne(int age, String name) throws NegativeAgeException {
    this.setAge(age);
    this.name = name;
}
```

# Exemples

## Personne.java

```
/**
 * Modifie l'age de la personne.
 *
 * @param age un entier >= 0
 * @throws NegativeAgeException si age < 0
 */
public void setAge(int age) throws NegativeAgeException {
    if (age < 0){
        throw(new NegativeAgeException("erreur : l'age doit etre > 0, "
            +" valeur entree : "+age)); //sort de la methode
    }
    this.age = age;
}
```

# Exemples

## Personne.java

```
@Override
/**
 * cree une representation sous forme de string de l'objet
 * personne
 * @return une chaine de caracteres
 */
public String toString() {
    return name+" age : "+age;
}
}
```

# Exemples

## NegativeAgeException.java

```
package cours.javadoc ;

/**
 * Exception generee lors d'une mauvaise
 * utilisation de la methode {@link Personne#setAge(int)}.
 *
 * @author fab
 * @see Personne
 */
public class NegativeAgeException extends Exception {

    /**
     * constructeur pour l'exception
     * @param message le message d'erreur
     */
    public NegativeAgeException(String message) {
        super(message);
    }
}
```

## Javadoc pour un package

Il faut ajouter, dans le répertoire du package, un fichier appelé **package-info.java** et structuré comme suit :

### package-info.java

```
/**
 * Give the summary of what the package allows to do.
 * And more details here. bla bla...
 *
 * @since Project 1.0
 * @author fab
 * @version 1.1
 */
package cours.javadoc;
```

# Utilisation du programme javadoc

- En ligne de commande :
  - option **-sourcepath** : racine des sources (= packages)
  - option **-d** : répertoire destination
  - arguments : noms de package (prog.nompackage) et/ou noms de fichiers .java

## Exemple

```
javadoc -sourcepath src -d doc cours.javadoc
```

## Plusieurs niveaux de visibilité possibles

- option `-public` : seuls les membres *public*
- option `-protected` : seuls les membres *public/protected*
- option `-package` : seuls les membres *public/protected/package*
- option `-private` : tout !

### Exemple

```
javadoc -private -sourcepath src -d doc cours.javadoc
```

## Liaison avec d'autres javadoc

Objet : créer des hyperliens avec d'autres javadoc

### option `-link` : liaison avec des API externes

```
javadoc -link http://download.oracle.com/javase/7/docs/api  
-private -sourcepath src -d doc cours.javadoc
```

## Options liées à l'encodage des caractères

- **-docencoding encodingType** : encodage utilisé pour les fichiers html (i.e. ISO-8859-1 ou UTF-8). Valeur de l'OS si absent.
- **-encoding encodingType** : spécifie l'encodage des fichiers sources (valeur de l'OS si absent)
- **-charset encodingType** : ajoute dans le source HTML l'header Content-Type avec l'encodage utilisé.

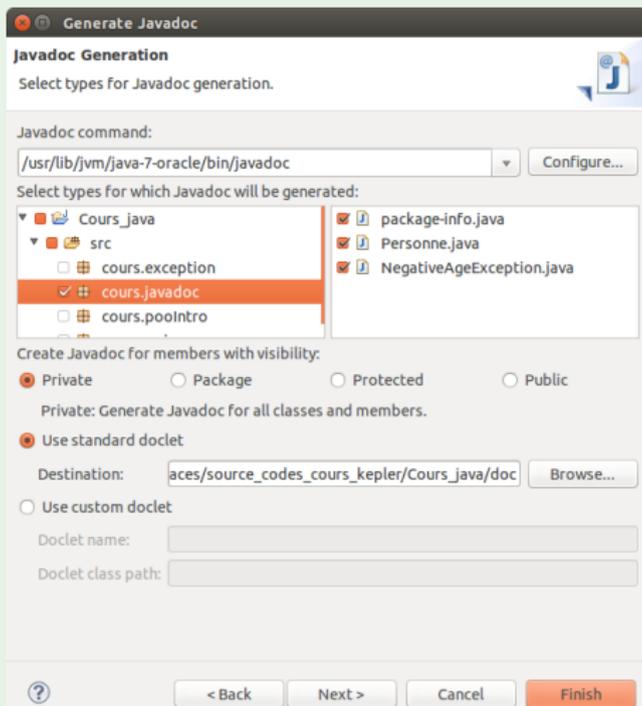
### Exemple sous linux (UTF-8 par défaut pour l'OS)

```
javadoc -charset UTF-8 -sourcepath src -d doc cours.javadoc
```

▶ Résultat

# Sous Eclipse

File → Export → Javadoc →



# Sous Eclipse

→ Next

**Generate Javadoc**

**Javadoc Generation**  
Configure Javadoc arguments for standard doclet.

Document title:

**Basic Options**

- Generate use page
- Generate hierarchy tree
- Generate navigator bar
- Generate index
- Separate index per letter

**Document these tags**

- @author
- @version
- @deprected
- deprected list

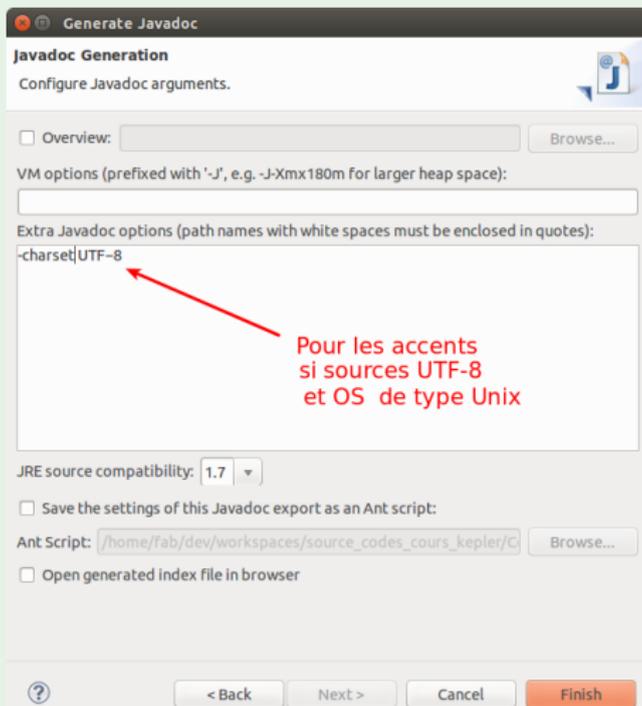
Select referenced archives and projects to which links should be generated:

- charsets.jar - http://download.oracle.com/javase/7/docs/api/
- Cours\_java - not configured
- dnsns.jar - not configured
- jce.jar - http://download.oracle.com/javase/7/docs/api/
- jfr.jar - http://download.oracle.com/javase/7/docs/api/
- jsse.jar - http://download.oracle.com/javase/7/docs/api/

Style sheet:

# Sous Eclipse

→ Next



**Generate Javadoc**

**Javadoc Generation**  
Configure Javadoc arguments.

Overview:

VM options (prefixed with '-J', e.g. -J-Xmx180m for larger heap space):

Extra Javadoc options (path names with white spaces must be enclosed in quotes):  
`-charset|UTF-8`

JRE source compatibility:  ▾

Save the settings of this Javadoc export as an Ant script:

Ant Script:

Open generated index file in browser

Pour les accents  
si sources UTF-8  
et OS de type Unix

## Quelques tags pour la javadoc

Tag	Description
@author	Nom du développeur
@deprecated	Marque la méthode comme dépréciée. Certains IDEs génèrent un avertissement à la compilation si la méthode est appelée.
@exception	Documente une exception lancée par une méthode — voir aussi @throws.
@param	Définit un paramètre de méthode. Requis pour chaque paramètre.
@return	Documente la valeur de retour. Ce tag ne devrait pas être employé pour des constructeurs ou des méthodes définis avec un type de retour <i>void</i> .
@see	Documente une association à une autre méthode ou classe.
@since	Précise à quelle version une méthode a été ajoutée à la classe.
@throws	Documente une exception lancée par une méthode. Un synonyme pour @exception disponible depuis Javadoc 1.2.
@version	Donne la version d'une classe ou d'une méthode.

▶ plus d'information