
Aide-mémoire de javadoc

1. Introduction

Javadoc est un utilitaire de jdk (inclus dans les distributions les plus récentes) qui permet de générer la documentation des classes développées, sous réserve d'avoir commenté le code java suivant les règles énoncées ci-dessous.

La documentation complète (installation, génération, format...) de javadoc est sur le site : d'Oracle javadoc.

La documentation sur les classes des API de java est générée par javadoc.

2. Les indispensables

- Format du commentaire

Chaque déclaration de package, classe, attribut, méthode... qui sera précédée du commentaire `/** commentaire */` sera pris en compte.

- Résumé et précisions

La première phrase du commentaire (jusqu'au premier '.') est utilisée comme description résumée de l'entité commentée. Les suivantes apparaissent dans la section des précisions.

- Règles de commentaires

Commencer les commentaires de méthodes par des verbes, ceux des classes et attributs peuvent ne pas avoir de sujet. Exemple pour une classe :

```
/**
 * Cumul des notes des etudiants par matiere
 * @author moi-meme
 * @version 1.0
 */
public class CumulNotes {
    ...
    /** Insérer une note. */
    public void insererNote(float n) throws BadNoteValueException { ...
    }
}
```

En règle général, selon le contexte dans lequel on travaille, écrire en anglais peut-être utile...

- HTML

Il est possible d'insérer des balises HTML directement dans les commentaires.

```
/** Regroupement des matieres en un module
 * On prendra comme exemple le module "Java" dont les matieres sont :
 * <li>Programmation</li>
 * <li>Java Reseau</li>
 * <li>Projet</li>
 * <p>
 * Les matieres sont ponderees par des coefficients.
 */
public class Module { ...
}
```

- Quelques tags

- `@param p commentaire` pour les paramètres de méthode

- `@return` commentaire pour le retour de méthodes
- `@throws class-name description` pour indiquer les exceptions levées
- Hiérarchie
Pour obtenir une hiérarchie avec plusieurs packages il suffit d'appliquer javadoc à l'ensemble de la hiérarchie. Pour commenter les packages, il faut ajouter un fichier `package-info.java` dans lequel on met les informations voulues. Le fichier `package-info.java` doit être dans l'arborescence du package.

```
/**
 * Procure les classes nécessaires pour gerer les notes d'un module.
 * <p>
 * Utilise principalement la classe { @link java.util.ArrayList }.
 * @author moi-meme
 * @version 1.0
 */
package note;
```

3. Exemples

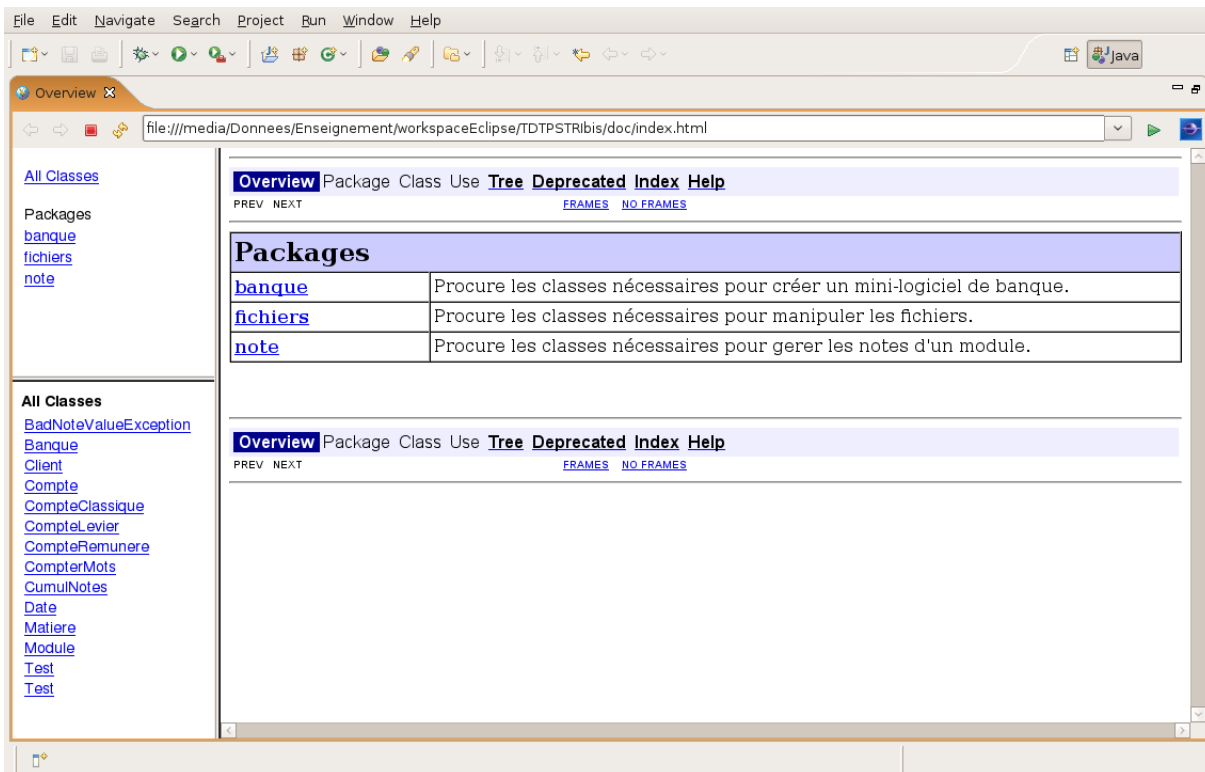


Figure 1: Documentation générée pour les packages du projet

[All Classes](#)

Packages
[banque](#)
[fichiers](#)
[note](#)

All Classes
[BadNoteValueException](#)
[Banque](#)
[Client](#)
[Compte](#)
[CompteClassique](#)
[CompteLevier](#)
[CompteRemunere](#)
[CompterMots](#)
[CumulNotes](#)
[Date](#)
[Matiere](#)
[Module](#)
[Test](#)
[Test](#)

[Overview](#) [Package](#) [Class](#) [Use Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV PACKAGE](#) [NEXT PACKAGE](#)

[FRAMES](#) [NO FRAMES](#)

Package note

Procure les classes nécessaires pour gerer les notes d'un module.

See: [Description](#)

Class Summary	
CumulNotes	Permet de cumuler une serie de notes et d'en calculer la moyenne.
Matiere	Matiere definie par son coefficient et le cumul de ses notes.
Module	Regroupement des matieres en un module.
Test	Teste les classes du package note.

Exception Summary	
BadNoteValueException	Signale une note en dehors de l'intervalle considérée

Package note Description

Procure les classes nécessaires pour gerer les notes d'un module.

Utilise principalement la classe ArrayList.

Author:
moi-même

[Overview](#) [Package](#) [Class](#) [Use Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV PACKAGE](#) [NEXT PACKAGE](#)

[FRAMES](#) [NO FRAMES](#)

Figure 2: Documentation générée pour le package note

[All Classes](#)

Packages
[banque](#)
[fichiers](#)
[note](#)

All Classes
[BadNoteValueException](#)
[Banque](#)
[Client](#)
[Compte](#)
[CompteClassique](#)
[CompteLevier](#)
[CompteRemunere](#)
[CompterMots](#)
[CumulNotes](#)
[Date](#)
[Matiere](#)
[Module](#)
[Test](#)
[Test](#)

[Overview](#) [Package](#) [Class](#) [Use Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

[SUMMARY: NESTED | FIELD | CONSTR | METHOD](#)

[FRAMES](#) [NO FRAMES](#)

[DETAIL: FIELD | CONSTR | METHOD](#)

note

Class CumulNotes

java.lang.Object
└─ **note.CumulNotes**

```
public class CumulNotes
extends java.lang.Object
```

Permet de cumuler une serie de notes et d'en calculer la moyenne.

Author:
bonenfant

Field Summary	
static int	MAX_VALUE Note maximale
static int	MIN_VALUE Note minimale

Constructor Summary	
CumulNotes()	Creates new CumulNotes

Method Summary	
float	calculerMoyenne() Calcule la moyenne
void	insérerNote(float n) Insère une note

Figure 3: Documentation générée pour la classe CumulNotes

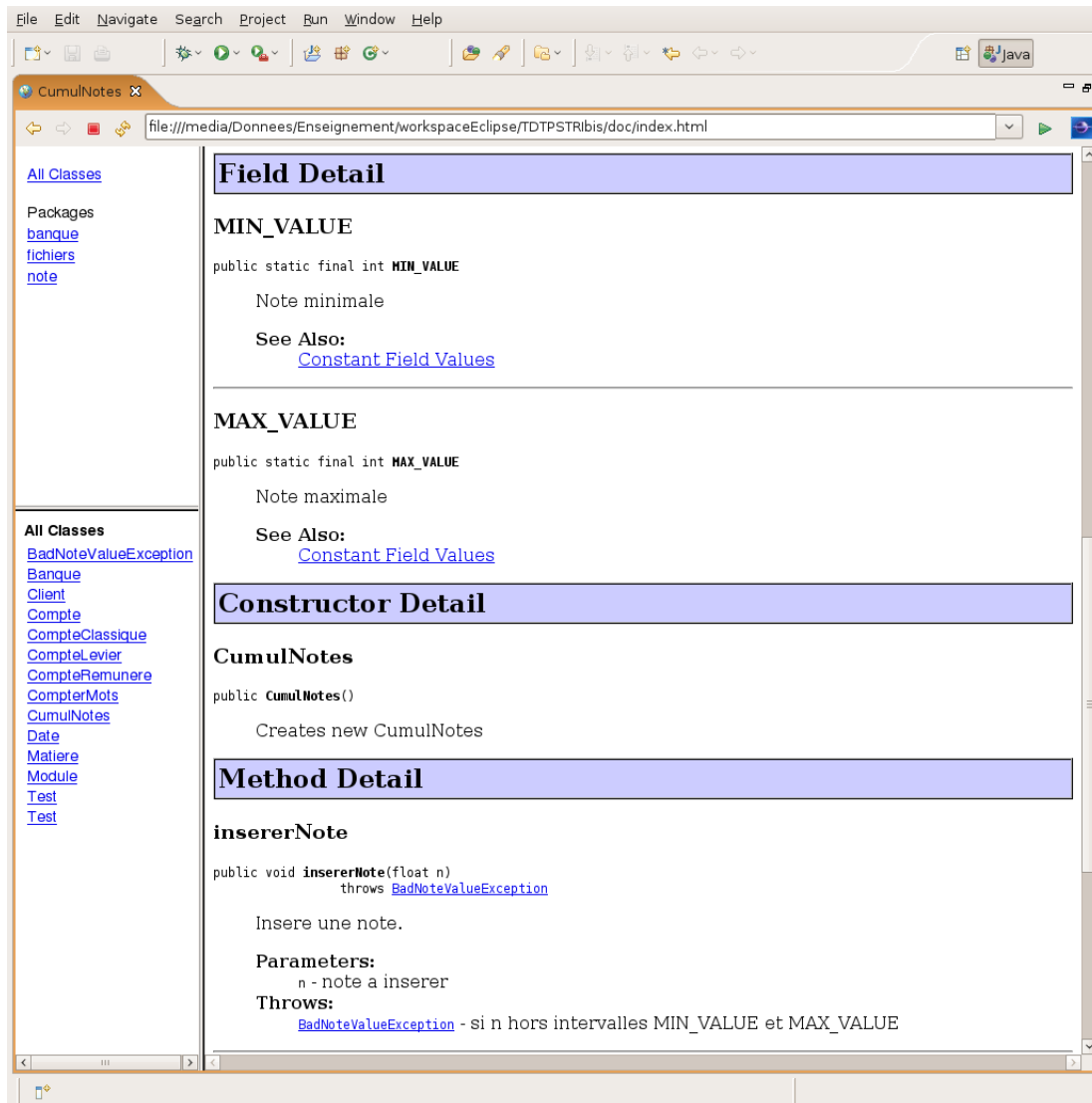


Figure 4: Documentation g n r e pour les attributs et m thodes de CumulNotes