

T.P. N°2 en Programmation Orientée Objet JAVA

L'outil Javadoc (<https://www.oracle.com/technical-resources/articles/java/javadoc-tool.html>)

- Javadoc est un outil, développé par Sun Microsystems et c'est aussi la documentation générée par cet outil.
- Cet outil permet, en inspectant le code Java des classes, de produire une documentation très complète du code
- Il se trouve dans le répertoire **bin** du dossier jdk
- Il génère des pages HTML contenant au minimum la liste des classes, la liste des méthodes et la liste des variables
- Il existe trois types de commentaires en Java. Les commentaires en lignes, les commentaires sur plusieurs lignes, et les commentaires Javadoc
- Toutes les informations que nous mettrons se trouveront dans ce commentaire Javadoc. Il doit se situer sur la ligne immédiatement avant le nom de la classe, de la méthode, ou de la variable.

▪ Les tags Javadoc

Tag	Rôle	Exemple
@param	sert à renseigner le ou les paramètres de la méthode. Après le tag, insérer le nom du paramètre (son type sera inclus automatiquement dans la Javadoc)	<pre> /** met à jour quantite * @param quantiteNew * valeur pour modifier quantite */ void setQuantite(int quantiteNew){ quantite=quantiteNew ;} </pre>
@return	sert à renseigner l'objet retourné par la méthode.	<pre> /** retourne le prix de vente * @return retourne le prix de vente d'un * produit donné*/ double calculerVente(double pu, int qte){ return pu*qte ;} </pre>
@author	renseigne le nom de l'auteur de la classe.	<pre> package chapitre 1; /** * la classe représente une introduction à javadoc * * @author DlsiADBD * @version 1.0 */ public class IntroductionJD { } </pre>
@version	indique le numéro de version de la classe ou interface	
@see	permet de faire une référence à une autre méthode, classe, etc.	<pre> /** * "prixVente" peut être calculé. * * @see calculerVente */ private prixVente; </pre>

Il existe aussi les tags :

- @throws : les exceptions propagées
- @since : permet de dater la présence d'une méthode, d'un paramètre. Derrière ce tag, il faut noter un numéro de version de la classe.
- @deprecated : indique qu'une méthode est dépréciée. Doit décrire la version depuis laquelle cette méthode / classe est dépréciée et ce qu'il faut utiliser à la place.

T.P

Démarrer Eclipse et ouvrir votre workspace (PrenomNomDlsiADBd) existant dans l'unité d:

Activité 1

1) Créer la classe suivante dans le package chapitre1

```
Class EssaiJD{
int b ;
int calculer(int a){
return a*a*a ;
}
void modifier(int b1){
b=b1;
}
```

Ajouter des commentaires javadoc

- 1) La description de la classe suivante « c'est un essai d'utilisation de commentaire Javadoc »
- 2) Donner votre nom comme nom d'auteur
- 3) La date de la classe
- 4) Le paramètre et le retour de la méthode calculer
- 5) Le paramètre de la méthode modifier

Générer le fichier javadoc rattaché à la classe EssaiJD

Activité 2

1) Créer le un package cert et une classe d'accès par défaut Beverage dans cert à l'aide de Eclipse

```
package cert;
class Beverage { }
```

2) Créer le package exam.stuff et une classe Tea qui hérite de Beverage dans exam.stuff

```
package exam.stuff;
import cert.Beverage;
class Tea extends Beverage { }
```

Compiler les deux classes, qu'est-ce que vous constatez?

la classe Beverage ..., mais a la compilation de la classe Tea il s'affiche le message suivant : "The Type cert.Beverage is not visible"

.....

.....

.....

=> pour que Tea compile il faut:

.....

.....

Faite la compilation dans chaque cas en utilisant javac

Activité 3

En utilisant Eclipse, créer un package food et un package heritfood. Créer, dans le package food, une classe abstract Fruit déclarant une méthode caracteristique et dans le package heritfood, une classe concrète Apple qui hérite de la classe Fruit (Apple **extends** Fruit) et développer la méthode caracteristique qui affiche un message « apple »

```
.....class Fruit{
.....
.....
.....
```

.....classe Apple

.....

.....

.....

.....

.....

.....

.....

.....

3. explorer les package food et heritfood et les autres packages que vous avez créées, qu'est-ce que vous remarquez ?

.....

.....