TP 2 : javadoc

Introduction à l'informatique et à la programmation (IF1)

Semaine du 3 octobre 2011

La *documentation* d'un code est destinée aux futurs utilisateurs de ce code. Ce ne sont pas des commentaires liés aux détails de programmation [1] : cette documentation doit permettre d'utiliser le code sans avoir à le regarder.
En Java, cette documentation peut être mise en place *dans* le code au moyen de commentaires entre /** et */ et d'une syntaxe appropriée. Ces commentaires permettent de générer une documentation au format html en utilisant la commande javadoc avec le nom du fichier source en argument. A noter que par défaut javadoc ne génère que la documentation des classes publiques : il ne s'agit pas de vous expliquer ici ce que cela signfie, mais ajoutez le mot-clef public devant les en-têtes de vos classes.
A toutes fins utiles, on rappelle l'url de la documentation officielle actuelle de Java :
http://download.oracle.com/javase/6/docs/api/

Tous les documents à récupérer sur didel se trouvent dans l'arborescence de racine le répertoire tp2 de Documents et liens.

► **Exercice 1** :

1. Recopiez le fichier suivant :

```
/**
 * classe inutile
 */
public class Inutile{
    /**
     * fonction qui ne fait rien
     * @param n un entier quelconque
     * @return n
     */
    public static int neFaitRien(int n){
        return n;
    }
}
```

2. Lancez la commande javadoc sur ce fichier : quels sont les fichiers d'extension html créés ?
3. Tous ces fichiers ont été créés dans votre répertoire courant, ce qui rend peu lisible son contenu. Nettoyez votre répertoire courant et créez-y un sous-répertoire docInutile. Relancez la commande javadoc pour que les fichiers soient créés dans ce sous-répertoire (option -d).

1. Ces commentaires sont très important pour permettre modifications et maintenance du code, ne les négligez pas.

4. Regardez le fichier `docInutile/index.html` dans un navigateur web et naviguez. Vous remarquez que la classe `Inutile` est présentée comme "provenant" de la classe `Object` (on dit qu'elle *étend* la classe `Object` ; ce qui est le cas de toutes les classes en `Java`), mais il n'y a pas de lien vers la classe `Object` dans la documentation. Pour y remédier, recréez la documentation en utilisant l'option `-link` avec comme argument l'url de la doc officielle.

Prenez l'habitude de documenter vos classes et vos fonctions au format standard `javadoc`, cela permet d'avoir facilement une documentation agréable et facilite la réutilisation d'anciens codes dans lesquels on n'a pas forcément envie de se replonger.

► **Exercice 2** :
Téléchargez sur `didel` les documents suivants du sous-répertoire `exercice_2` : `CanvasTP2.class`, `FenetreTP2.class`, `ImageTP2.class` et `img.jpg` et placez-les dans votre répertoire de travail.
Suivez maintenant le lien `javadoc.url` (toujours au même endroit sur didel) : c'est la documentation des classes `FenetreTP2` et `ImageTP2`.

Écrivez un programme `Java` qui permet d'afficher l'image `img.jpg` à l'écran en vous servant des classes `FenetreTP2` et `ImageTP2`.

► **Exercice 3** (inspiré du partiel 2004/2005, exercice 4) : Téléchargez sur `didel` les fichiers suivant du sous-répertoire `exercice_3` : `Grille.class` et `MonPoint.class`. La documentation de ces classes est disponible au même endroit en suivant le lien `javadoc.url`.
Pour toutes les questions qui suivent, il s'agit d'écrire le code *et de documenter* les fonctions. Dans une même classe :

1. Écrire une fonction qui teste si un point est sur une grille (c'est-à-dire renvoie `true` s'il est dessus et `false` sinon).
2. Écrire une fonction qui teste si un point est sur la bordure d'une grille.
3. Écrire une fonction qui teste si un point est un des coins de la grille.
4. Écrire une fonction qui teste si 3 points sont alignés.
   Rappel : 3 points $A(x_A, y_A)$, $B(x_B, y_B)$ et $C(x_C, y_C)$ sont alignés si et seulement si les vecteurs $\overrightarrow{AB}(x_{AB} = x_B - x_A, y_{AB} = y_B - y_A)$ et $\overrightarrow{AC}(x_{AC}, y_{AC})$ sont colinéaires, c'est-à-dire si et seulement si $x_{AB}y_{AC} = x_{AC}y_{AB}$.
5. Générer la documentation de votre classe.

► **Exercice 4** : Votre point de départ dans l'API : la classe `java.util.Date`.
Écrivez une fonction `afficheDateCourante` qui affiche la date courante au format JJ/MM/AAAA.