

# TP: Consignes générales et rappels de Java

## 1 Consignes générales pour les TP

### 1.1 Edition

- Veuillez utiliser les outils indiqués dans les sujets et **respecter** les notations données pour les identifiants<sup>1</sup>.
- Vous travaillerez sur les machines de l'IUT, en utilisant l'éditeur *Notepad++*. Les sources seront sur l'emplacement réseau **T:\RCPI01** dans lequel vous aurez un répertoire par séance de TP (**TP1**, **TP2**, **TP3**, etc.)
- Convention Java : un programme dans un fichier nommé **Abcd.java** est défini par une classe nommée **Abcd** : la première lettre de la classe (et donc du fichier) est en majuscules. Les fichiers sources auront l'extension **.java** (voir page suivante).
- Une **indentation** conforme aux consignes est impérative! Vous appliquerez le style "Allman", conformément à cet exemple :
  - Aucun caractère (verticalement) entre accolade ouvrante et fermante.
  - Les accolades seront le seul caractère sur leur ligne.
  - Une tabulation (caractère ASCII **0x09**) pour chaque ligne d'un bloc.

```

if( condition1 )
{
    if( condition2 )
    {
        faireuntruc();
    }
    else
    {
        if( condition3 )
        {
            fairececi();
        }
        else
        {
            fairecela();
        }
    }
}

```

S'il un bloc n'est composé que d'une seule instruction, il est toléré de supprimer les accolades :

```

if( condition )
{
    faireuntruc();
}

```

⇒

```

if( condition )
    faireuntruc();

```

### 1.2 Compilation et exécution

- La compilation en *bytecode* Java sera faite via une console ouverte dans le dossier en cours, avec la commande `javac`
- Par exemple, ceci : `javac *.java` va compiler tous les fichiers sources du dossier courant.
- Pour lancer un programme : `java Nomduprog` (respectez la casse!)
- A l'exécution, pour sortir d'une boucle infinie ou d'une attente quelconque : CTRL-C

## 2 Problèmes divers

### 2.1 Caractères invalides

**Problème :** Si le compilateur vous envoie un message bizarre à propos de caractères invalides tout au début de votre fichier, et que ces caractères sont invisibles dans l'éditeur, c'est très probablement parce que *Notepad++* utilise l'encodage UTF avec **BOM**<sup>2</sup>

1. Une partie de l'évaluation portera sur ce respect.

2. *Byte Order Mark* : deux octets ajoutés automatiquement en début de fichier pour identifier l'*Endianess* du fichier.

**Solution :** modifier l'encodage en spécifiant UTF8 (sans BOM), et enregistrez le fichier.

## 2.2 Chemin du compilateur

**Problème :** à la compilation, vous avez le message "'javac' n'est pas reconnu en tant que commande interne ou externe, ..."

Ceci signifie que le compilateur (**javac.exe**) n'est pas trouvé dans les chemins par défaut définis sur la machine.

**Solution manuelle :** Il faut ajouter le chemin du compilateur dans le "PATH" par défaut :

1. Ouvrir un explorateur (touche "Windows" + E)
2. Naviguer jusqu'à **C:\Program Files\Java\jdk1.7.XXXX\bin** (ou **jdk1.8 ...**)
3. Sélectionner cette ligne dans la barre d'adresse de l'explorateur, puis CTRL-C
4. Ouvrir un deuxième explorateur
5. Sélectionner dans le volet de gauche "Poste de travail", puis clic-droit → Propriétés  
⇒ Ouverture d'une boîte de dialogue "Propriétés Système"
6. Sélectionner l'onglet "Avancé", puis le bouton "Variables d'environnement"
7. Dans la partie "variables utilisateur", descendre jusqu'à la variable **Path**
8. bouton "Modifier"
9. Aller tout à la fin de la ligne, ajouter un ";", puis coller (CTRL-V)
10. Valider pour fermer les dialogues (deux fois)

Il faut ensuite fermer la console ouverte et en ré-ouvrir une autre (la valeur de la variable PATH n'est lue que au lancement de la console).

## 3 Rappels de Java

1. Un programme Java correspond à une classe publique de même nom que le fichier qui la contient, et dans laquelle on déclare une méthode **main()**. Voir l'exemple ci-dessous.
2. Arguments de la ligne de commande  
En ligne de commande, l'utilisateur peut passer des arguments à un programme via des arguments, avec l'espace comme séparateur :

```
> java MonProg un "deux trois" 567
```

Le programme pourra accéder à ces chaînes via un tableau de **String** en paramètre de la fonction **main()** :

- Fichier **MonProg.java** :

```
public class MonProg
{
    public static void main( String[] args )
    {
        System.out.println( "nb args=" + args.length );
        for( int i=0; i<args.length; i++ )
            System.out.println( "arg " + i + "=" + args[i] );
        ...
    }
}
```

Ce programme affichera ici :

```
nb args=3
arg 0=un
arg 1=deux trois
arg 2=567
```