TD1 – Prise en main de l'environnement de développement

Exercice 1 : Environnement de développement

Pour les premières séances de Java, nous vous demandons de garder vos habitudes du 1er semestre. Vous continuerez à éditer vos programmes sources avec votre éditeur préféré (gvim parexemple). Vous utiliserez les outils en ligne de commande `javac` et `java` ou le gestionnaire de projet `ant` pour compiler et exécuter vos programmes. Il est fortement conseillé d'ouvrir un navigateur web pour consulter la documentation en ligne de Java pendant le développement de vos applications.

Quand vous maîtriserez ces différents éléments, vous serez autorisés à utiliser un IDE de type Eclipse.

1. Vérifier la version de votre compilateur et celle de la machine virtuelle avec l'option –version. Sont-elles cohérentes ?
2. Trouver la documentation de l'API correspondant à votre version de Java sur le site d'Oracle : http://www.oracle.com/technetwork/java/index.html
   Pensez à mettre un signet sur l'index de l'API dans votre navigateur. Il est même conseillé de télécharger cette documentation en local sur sa machine pour pouvoir développer indépendamment de la disponibilité du réseau.
3. Cherchez la documentation de la classe `System` que nous allons utiliser dans notre premier exemple. Comment s'appellent l'entrée standard, la sortie standard et la sortie d'erreur dans cette classe ?

Exercice 2 : Premier programme Java : Hello World

Soit le programme du cours :

```
/**
 * The HelloWorldApp class implements an application that
 * simply prints "Hello World!" to standard output.
 */
public class HelloWorldApp {
   public static void main(String[] args){
      System.out.println("Hello World!");  //Display the string.
   }
}
```

L'objectif de ce premier exercice est de compiler et exécuter un premier programme fourni, en découvrant les techniques de base de la manipulation des outils de production Java.

1. Créez un répertoire de travail
2. Recopiez le code de ce programme dans un fichier Hello.java.
3. Compilez le fichier. Que se passe-t-il ?
4. Renommez le fichier source en HelloWorldApp.java et recompilez.
5. Exécutez le programme .class produit.
6. Effacez le fichier compilé produit précédemment.
7. En examinant le compilateur et ses options (avec l'option `–help`), trouvez le moyen de produire les fichiers compilés dans un répertoire `build/classes` à côté de vos sources dans le répertoire `src`.

```
ProjetHelloWorld
   |
   |___src
   |     |___HelloWorldApp.java
   |
   |___build
         |___classes
               |___HelloWorldApp.class
```

8. Comment exécuter le programme produit à partir du répertoire racine du projet.

Exercice 3 : Automatisation avec Ant

Automatiser les tâches de l'exercice précédent avec un fichier de configuration `ant` avec :

1. une cible `compile` pour compiler les sources du répertoire `src` et placer le bytecode compilé dans le répertoire `build/classes` ; les répertoires `build` et `classes` sont à créer s'ils n'existent pas ;
2. une cible `run` pour exécuter le programme ;
3. une cible `cleanall` pour nettoyer le projet en supprimant le répertoire `build`.

Exercice 4 : Hello World en français

Modifiez le programme HelloWorldApp.java pour qu'il affiche `Bonjour le Monde !` au lieu de `Hello World !`.

1. éditez le source ;
2. recompilez ;
3. exécutez pour vérifier.

Exercice 5 : Hello World à l'heure

Trouver dans la documentation de la classe `System` comment récupérer l'heure du système. Déclarer trois variables entières dans lequelles vous stockerez respectivement les heures, les minutes et les secondes obtenues à partir de l'heure système. Afficher ces trois informations après avoir dit bonjour :

```
Bonjour tout le monde ! Il est 15h21'23.
```

Vous pouvez utiliser les mêmes opérateurs sur les entiers qu'en C (+, -, *, /, %) pour vos calculs ainsi que l'opérateur de concaténation de chaînes de caractères + :

```
System.out.println("texte 1 " + uneVariableTypeQuelconque + " texte 2");
```

Exercice 6 : Hello World ne dit pas bonjour

Etudiez les versions modifiées de HelloWorldApp suivantes. Ces programmes ont tous une erreur. Trouvez là puis compilez pour vérifier.

1. HelloWorldApp2.java :

```
/**
 * The HelloWorldApp class implements an application that
 * simply prints "Hello World!" to standard output.
 */ class HelloWorldApp2 {
   public static void main(String[] args){
      System.out.println("Hello World!");  //Display the string.
   }
}
```

2. HelloWorldApp3.java :

```
/**
 * The HelloWorldApp class implements an application that
 * simply prints "Hello World!" to standard output.
 */
class HelloWorldApp3 {
   public static void main(String[] args){
   System.out.println("Hello World!");  //Display the string.
}
```

3. HelloWorldApp4.java :

```
/**
 * The HelloWorldApp class implements an application that
 * simply prints "Hello World!" to standard output.
 */
class HelloWorldApp4 {
   public static void main(string[] args){
      System.out.println("Hello World!");  //Display the string.
```

```
    }
}
```

4. HelloWorldApp5.java :

```
/**
 * The HelloWorldApp class implements an application that
 * simply prints "Hello World!" to standard output.
 */
public class HelloWorldApp4 {
    public static void main(String[] args){
        System.out.println("Hello World!");  //Display the string.
    }
}
```