

Ing1 - TD 1 - Programmation Java

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 (`xemacs` par exemple). Vous utiliserez les outils en ligne `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 l'environnement de développement Eclipse.

1. Vérifier la version de votre compilateur et 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 de Sun : <http://java.sun.com>. Pensez à mettre un signet sur l'index de l'API dans votre navigateur.
3. Cherchez la documentation de la classe `System` que nous allons utiliser dans notre premier exemple.
4. Pour pouvoir développer indépendamment d'éventuels problèmes réseaux, il est fortement conseillé de télécharger cette documentation et de l'installer sur votre machine. Pensez toujours à prendre la version correspondant à votre installation.

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 bases de la manipulation des outils de production Java.

1. Créez un répertoire de travail
2. Récupérez le code de ce programme élémentaire. Faites-en un fichier `Hello.java`
3. Compilez le fichier. Que se produit-il ?
4. Renommez le fichier source en `HelloWorldApp.java` et recompilez.
5. Exécutez le programme `.class` produit.
6. 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 du répertoire de vos sources `src`.

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

7. Effacez le fichier compilé produit précédemment et recompilez vers le répertoire de sortie `build/classes`.
8. Comment peut-on exécuter le programme ?
9. Ajouter les commentaires habituels pour être en conformité avec les règles de programmation de l'école.

Exercice 3 : Automatisation avec Ant

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

1. 1 cible `compile` pour compiler les sources du répertoire `src` et placer le bytecode compilé dans le répertoire `build/classes`. Si le répertoire de production n'est pas créé, `ant` devra le faire.
2. 1 cible `run` pour exécuter le programme
3. 1 cible `clean` 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 lesquelles vous stockerez respectivement les heures, les minutes et les secondes obtenues à partir de l'heure système.

Afficher ces 3 informations après avoir dit bonjour :

```
Bonjour tout le monde ! Il est 15h45'23.
```

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

```
System.out.println("une chaîne " + uneVariableDUnTypeQuelconque + " une autre chaîne");
```

Exercice 6 : Hello World ne dit pas bonjour

Vous allez trouver des versions modifiées de `HelloWorldApp` qui ne compilent pas :

```
- HelloWorldApp2.java :
// INTENTIONALLY UNCOMPILABLE!
/**
 * 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.
    }
}

- HelloWorldApp3.java :
// INTENTIONALLY UNCOMPILABLE!
/**
 * 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.
    }
}

- HelloWorldApp4.java :
// INTENTIONALLY UNCOMPILABLE!
/*
 * The HelloWorldApp class implements an application that
 * simply prints "Hello World!" to standard output.
```

```

*/
class HelloWorldApp4 {
    void public static main(string[] args) {
        System.out.println("Hello World!"); // Display the string.
    }
}
- HelloWorldApp5.java :
// INTENTIONALLY UNCOMPILABLE!
/**
 * 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.
    }
}

```

Ces programmes ont tous une erreur. Trouvez la (en compilant) et vérifiez?