

Atelier Programmation Orientée Objet Avancée –JAVA–

TP1

ISET Zaghouan



Enseignant

Boukchim_mossaab@yahoo.fr

I. Avant de programmer en Java

Le JDK de Sun (Java Development Kit) est l'outil essentiel pour programmer en Java. Il permet la compilation, le débogage et l'exécution d'applications et d'applets Java. Il comprend également des fonctions avancées comme les signatures numériques, la création d'archives et de documentation ou l'intégration de code natif C. La documentation de JDK est en fait l'API (Application Programming Interface) de Java (le détail de toutes les fonctions du langage). Après avoir installé le JDK, il faut définir un chemin de recherche des exécutables « *PATH* ».

Pour le faire, Lancer une session **DOS** et taper :

Set Path=C:\JDK\bin ou bien directement **Path=C:\JDK\bin**

(Donne la valeur **C:\JDK\bin** à la variable d'environnement path) Ou encore

Set Path=%Path%;C:\JDK\bin

(Ajoute le chemin **C:\JDK\bin** à la valeur de la variable d'environnement path)

Ceci ne fonctionne que si **C:\JDK** est le répertoire d'installation du JDK. Si ce n'est pas le cas, remplacer-le par votre vrai répertoire.

II. La compilation d'un code source

Pour compiler un fichier source il suffit d'invoquer la commande **javac** avec le nom du fichier source avec son extension .java :

javac NomFichier.java

Le nom du fichier doit correspondre au nom de la classe principale en respectant la classe même si le système d'exploitation n'y est pas sensible.

```
Public class HelloWorld {  
//Code}}
```

Le nom du fichier sera **HelloWorld.java**

Une fois le fichier enregistré, ouvrir une fenêtre DOS et aller au répertoire où le fichier est enregistré (**Rappel** : utiliser **cd nomrépertoire** pour monter d'un niveau et **cd..** pour descendre).

Pour le compiler taper :

Javac HelloWorld.java

Suite à la compilation, le pseudo-code Java est enregistré sous le nom HelloWorld.class, ce fichier est compilé et sera interprété par la machine virtuelle.

III. L'exécution d'un programme java

Une classe ne peut être exécutée que si elle contient une méthode `main()` correctement définie. Pour exécuter un fichier contenant du bytecode, toujours dans le même répertoire (dans lequel est créé le fichier `.class`), il suffit d'invoquer la commande `java` avec le nom du fichier source avec ou sans son extension `.class`

java NomFichier et dans notre exemple **java HelloWorld**

Les Exercices

Exercice 1 :

On se propose de faire fonctionner un programme Java dont le rôle est d'afficher « J'aime JAVA ! »

```
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello World!");  
    }  
}
```

Exercice 2 :

On se propose de faire fonctionner un programme Java dont le rôle est d'afficher le premier mot qu'on lui passe comme paramètre d'exécution.

```
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println(args[0]);  
    }  
}
```

Exercice 3:

Créer un fichier `JouerAvecMethode.java` contenant la classe `JouerAvecMethode`

2- Ajouter une méthode `void Afficher ()` qui fait l'affichage d'une chaîne de caractères exemple : « Bienvenue à l'ISSET Zaghuan ! »

3- Ajouter la méthode `main` dans laquelle vous faites instancier la classe `JouerAvecMethode` et référence l'instance ainsi créée à une variable `s` qu'on déclare comme suit :

`JouerAvecMethode s=new JouerAvecMethode () ;`

4- Dans la méthode `main` faites un appel à la méthode `Afficher ()` de l'objet `s` comme suit :

`s.Afficher() ;`

```
public class HelloWorldWithMethod{  
    public void Hello(){  
        System.out.print("You're Welcome!");  
    }  
    public static void main(String[] args) {  
        HelloWorldWithMethod s=new HelloWorldWithMethod();  
        s.hello();  
    }  
}
```

Exercice 4:

- 1- Créer un fichier JouerAvecAttribut.java contenant la classe JouerAvecAttribut
- 2- Ajouter un attribut String maChaine à la classe
- 3- Ajouter une méthode void Afficher () qui fait l'affichage de la chaîne de caractères attribut de la classe comme suit System.out.println(this.maChaine);
- 4- Ajouter la méthode main dans laquelle vous faite instancier la classe JouerAvecAttribut et référence l'instance ainsi créée à une variable s qu'on déclare comme suit :
JouerAvecAttribut s=new JouerAvecAttribut () ;
- 5- Dans la méthode main faite affecter l'attribut de l'objet s comme suit s.maChaine='Salut '
- 6- Dans la méthode main faite un appel à la méthode Afficher () de l'objet s comme suit
s. Afficher () ;

```
public class HelloWorldWithAttribut {  
    String maChaine;  
    public void Hello(){  
        System.out.print(maChaine);  
    }  
    public static void main(String[] args) {  
        HelloWorldWithAttribut s=new HelloWorldWithAttribut ();  
        s.maChaine="salut";  
        s.hello();  
    }  
}
```