

Programmation Orientée Objet

Programmation JAVA

FISE 3A ICy / FISA 3A Informatique

TP1 : Introduction à Java

Mohamed Amine BOUDIA

UPHF, CNRS, UMR 8201 - LAMIH, F-59313 Valenciennes, France

Email : mohamedamine.boudia@uphf.fr



Objectif Zéro Papier



Partie 1

Console / IDE

- ▶ En Java chaque classe publique doit être éditée dans un fichier d'extension «.java » . Le nom de ce fichier est le même nom de la classe
- ▶ Nous allons commencer par quelques petits programmes afin de se familiariser avec JAVA.
 - ▶ Crée un nouveau sous répertoire TP1 dans le répertoire Java.

```
public class HelloINSAHdF {  
    public static void main(String[] args) {  
        System.out.println("Hello INSA HdF");  
    }  
}
```

- ▶ Recopier ce programme java dans un fichier texte et enregistrer le.
- ▶ Lancer l'invite de commande CMD
- ▶ Compiler ce programme en utilisant la commande javac il prends comme paramètre le nom de fichier tp1.java
- ▶ vérifier que le fichier .class a été créé.
- ▶ Exécuter le programme avec la commande java qui prends en paramètre le nom de la classe

Remarque: Quand marche virtuelle est démarré par la commande java, elle cherche d'elle même le fichier dont le nom est celui qui est fourni en lui ajoutant le suffixe .class.

- ▶ En Java chaque classe publique doit être éditée dans un fichier d'extension «.java » . Le nom de ce fichier est le même nom de la classe
- ▶ Nous allons commencer par quelques petits programmes afin de se familiariser avec JAVA.
 - ▶ Crée un nouveau sous-répertoire TP1 dans le répertoire Java.

```
public class ArgsTest {  
    public static void main(String[] args) {  
        System.out.println("Nom : " + args[0] + " Prénom : " + args [1]);  
    }  
}
```

- ▶ Recopier ce programme java dans un fichier texte et enregistrer le.
- ▶ Lancer l'invite de commande CMD
- ▶ Compiler ce programme en utilisant la commande javac il prends comme paramètre le nom de fichier tp1.java
- ▶ vérifier que le fichier .class a été créé.
- ▶ Dans l'entête d'une méthode main apparaît ce qui s'appelle un argument nommé args. Cet argument est de type tableau de String, le premier est accessible par args[0], le deuxième par args[1] et ainsi de suite
- ▶ Exécuter le programme avec la commande " java ArgsTest Dupont Phillippe "

IDE : Eclipse

- ▶ [Voir le fichier Initiation Java-Eclipse sur moodle](#)

Partie 2

Syntaxe de base du langage Java

Exercice 1

- ▶ Quelles erreurs ont été commises dans chacun des groupe d'instruction suivants. On suppose que les variable concernées sont d'un type primitif numérique et qu'elles ont été correctement déclarées

```
If(a<b) System.out.println ("ascendant")  
else    System.out.println ("non ascendant")
```

```
If(a<b) { System.out.println ("ascendant"); max = b }
```

```
int n, p ;  
....  
switch (n) {  
case 2 : System.out.println ("petit"); break;  
case p : System.out.println ("limite"); break;  
}
```

```
int n;  
final int LIMITE = 20  
....  
switch (n) {  
case LIMITE - 1 : System.out.println ("trop petit"); break;  
case LIMITE : System.out.println ("OK");  
case LIMITE + 1 : System.out.println ("trop grand"); break;  
}
```

Exercice 2

► Quelles erreurs ont été commises dans chacun des instructions suivantes ?

- `Do n++ while (n<10); // instruction 1`
- `Do while (n!=10); // instruction 2`
- `Do ; while (true); // instruction 3`
- `Do { } while (false); // instruction 4`

Exercice 3

- ▶ Quelles résultats fournit le programme suivant?

```
Public class exo3 {  
    public static void main (String [] args) {  
        int n=0, p ;  
        while (n < = 5) n++;  
        System.out.println ("A : n = " + n);  
        n=p=0;  
        while (p < = 8) p++;  
        System.out.println ("B : n = " + n);  
        n=p=0;  
        while (p < = 8) n+= ++p;  
        System.out.println ("C : n = " + n);  
    }  
}
```

Exercice 4

- ▶ Quels résultats fournit le programme suivant?

```
Public class exo4 {  
    public static void main (String [] args) {  
        int n=0;  
        do {  
            if (n%2 == 0) {  
                System.out.println (n + " est pair " );  
                n+=3; continue;  
            }  
            if (n%3 == 0) {  
                System.out.println (n + " est multiple de 3 " );  
                n+=5;  
            }  
            if (n%5 == 0) {  
                System.out.println (n + " est multiple de 5 " ); break;  
            }  
            n+=1  
        } while (true);  
    }  
}
```

The background features abstract, overlapping geometric shapes in various shades of blue, ranging from light sky blue to deep navy blue. These shapes are primarily located on the right side of the frame, creating a modern, layered effect.

Partie 3

Exercices

Exercice 1

- ▶ La suite Fibonnacci est définie par la formule de récurrence suivante :
 - $U_1 = 1$;
 - $U_2 = 2$;
 - $U_n = U_{n-1} + U_{n-2}$ ($n \geq 3$)
- ▶ Ecrire un programme Fibo1.java qui permet de calculer le nième terme de la suite de Fibonacci, n étant fixé par l'utilisateur
- ▶ Ecrire un programme Fibo2.java qui permet d'obtenir la valeur et la rang du premier terme de cette suite supérieure à une valeur donnée par l'utilisateur,

Exercice 2

► Ecrire un programme Java pour chaque tâches:

- Lire deux entiers afficher leur somme.
- Lire une somme en Francs, un taux de change des Francs en Euros, et afficher la somme correspondante à la somme lue en Euros.
- Lire 3 valeurs réelles et afficher la moyenne des 2 plus grandes
- Lire 3 entiers représentant une heure donnée sous la forme " heure, minute, seconde " et afficher le triplet représentant l'heure à la seconde suivante
- Lire 10 entiers afficher leur somme et leur moyenne
- Lire une suite d'entiers terminée par 0, et afficher le nombre d'entiers lus, la somme des entiers lus et la moyenne des entiers lus.

Exercice 3

- ▶ Ecrire un programme qui affiche un carré composé de caractères '*', dont la taille est entrée par l'utilisateur. Par exemple, pour une taille 4, ce sera:

```
****  
****  
****  
****
```

- ▶ Ecrire un programme qui affiche un triangle avec la pointe tournée vers le haut, on demandera à l'utilisateur de rentrer le nombre de lignes:

```
*  
***  
*****  
*****
```

Exercice 4

► Ecrire un programme pour chaque tâches:

- Lire une chaîne et l'afficher à l'envers.
- Lire une chaîne de caractères, et changer toutes les lettres minuscules en lettres majuscules, et afficher la chaîne de caractères obtenue.
- Lire un caractère, puis une chaîne de caractères, et afficher le nombre d'occurrences du caractère dans la chaîne de caractères.
- Lire un caractère C, puis une chaîne de caractères, et afficher la chaîne de caractères lue après avoir ôté de la chaîne de caractères toutes les occurrences du caractère C
- Lire une chaîne de caractères qui contient un nombre entier en notation binaire, calculer et afficher sa valeur en notation décimale.

Exercice 5

- ▶ Ecrire un programme, qui affiche les n premiers nombres pairs où n est un entier entré par l'utilisateur. Par exemple, les 4 premiers nombres pairs sont: 2, 4, 6 et 8.

Exercice 6

- ▶ Ecrire un programme qui saisit un nombre entier et l'affiche à l'envers. Par exemple, si on rentre le nombre 78151, le programme affichera 15187. Vous utiliserez pour cela une boucle qui extrait un chiffre à afficher.

Les opérations de division et de modulo ou reste de la division entière (opérateur `%`) vous aideront.

Exercice 7

- ▶ Soit un tableau de 10 entiers. Ecrire un programme.java qui affiche
 - La moyenne des éléments de ce tableau
 - La valeur minimal de ce tableau
 - La valeur maximale de ce tableau

Exercice 8

- ▶ Ecrire un programme Java nommée `SuppressionDoublons` qui permet à l'utilisateur de remplir un tableau de taille n (n étant donnée par l'utilisateur).
- ▶ Le programme se charge alors de créer un second tableau qui contient les éléments du premier sans leurs doublons
- ▶ Voici l'affichage souhaité
 - Le tableau original est : `[3, 5, 7, 2, 3, 4, 5, 8, 6, 9, 9]`
 - Le tableau sans doublons est : `[3, 5, 7, 2, 4, 8, 6, 9]`

Exercice 9

- ▶ Ecrire un programme Java qui lit un mot au clavier et qui indique combien de fois sont présentes chacune des voyelles a, e, i, o, u et y, que celles-ci soient écrites en majuscules ou en minuscules et l'affiche .

Exercice 10

- ▶ Ecrire un programme Java qui lit au clavier un verbe du premier groupe et qui en affiche la conjugaison au présent de l'indicatif. On supposera qu'il s'agit d'un verbe régulier.
- ▶ Si le verbe n'est pas du premier groupe votre programme devra afficher :

« Le verbe saisi ne se termine pas par er, donnez-en un autre verbe »

Et attendre que l'utilisateur entre un autre verbe,

BONUS

- ▶ Reprenez la fiche TD 1 et faites les exercices sur machine