

TP JAVA : Les bases

I. Hello World en ligne de commande

- Ouvrez un éditeur de texte comme notepad, gedit, kate ou autre Geany et écrivez le code du programme `Hello World` vu en cours.
- Sauvez le fichier sous le nom `Hello.java` dans le répertoire de votre choix
- Ouvrez un interpréteur de commande type bash ou command prompt ou cmd.
- Allez dans le répertoire où est sauvé le fichier `Hello.java` et compilez ce fichier avec la commande `javac Hello.java`.
- Vérifiez le nom du fichier généré et ce qu'il contient.
- Exécutez le programme Hello en lançant la commande `java Hello`.
- Vérifiez que le texte « `Hello world` » s'affiche à l'écran.

I. Installation et utilisation d'Éclipse avec Hello World

Lisez le tutoriel « `Eclipse_premiers_pas.pdf` » et créer le projet `HelloWorld` avec Eclipse. Vous trouverez en prime quelques trucs et astuces très utiles pour utiliser Éclipse efficacement.

À partir de là, vous pouvez créer un projet par exercice ou par session d'exercices. Par ailleurs, vous ferez attention à ne créer qu'une seule classe par fichier.

II. Calculatrice simple

Écrire un programme prenant un nombre sur l'entrée standard et affichant celui-ci. Pour cela, on utilisera un objet (une instance) de la classe **Scanner** et particulièrement sa **méthode `nextInt()`**.

```
import java.util.Scanner;

public class Calc
{
    public static void main(String[] args)
    {
        Scanner sc;
        sc = new Scanner(System.in);
        System.out.print("Saisir un entier : ");
        int value;
        value = sc.nextInt();

        // compléter ici

        sc.close() ; // fermeture de l'entrée

    }
}
```

Afin de comprendre le programme, il est utile de regarder la **Javadoc** et les **tutoriels de Sun**.

1. Recopier le programme précédent et le compléter pour qu'il affiche le nombre saisi par l'utilisateur.
2. Indiquer dans le programme où sont les variables et quel est leur type associé.
3. Modifier le programme pour déclarer et initialiser les variables en une seule ligne
4. Modifier le programme pour qu'il demande deux entiers et affiche la somme de ceux-ci.
5. Afficher en plus de la somme, la différence, le produit, le quotient et le reste.

III. Quelques exemples sur les structures de base

6. Écrire un programme qui saisit 4 valeurs et affiche la plus grande.
7. Écrire un programme qui affiche la factorielle d'un nombre saisi : $n! = 1 \times 2 \times 3 \times \dots \times n$
8. Écrire un programme qui teste si un nombre entier positif saisi au clavier est premier.

IV. Afficher les arguments d'un programme Java

Écrire une classe **PrintArgs** qui affiche les arguments d'un programme Java.

Indication: Dans le menu contextuel de votre classe PrintArgs, sélectionnez *Exécutez en tant que* → *Configurations*. Puis dans l'onglet *Arguments*, entrer les arguments de votre programme

Les arguments de la ligne de commande sont stockés dans le tableau de chaînes de caractères passé en argument à la méthode **public static main(String[] args)**.

9. Dans un premier temps, afficher le premier argument de la ligne de commande. Que se passe-t-il si l'on ne passe pas d'argument lors de l'exécution du programme ?
10. Écrire une boucle affichant le contenu du tableau en sachant qu'en Java les tableaux possèdent un champ (un attribut, une variable d'instance) `length` qui renvoie la taille du tableau.
11. Écrire une seconde boucle utilisant la syntaxe dite 'foreach' `for(Type value:array)`

Exemple pour un tableau d'objets de type String:

```
String [] names = { "Le", "langage", "Java", "me", "plait", "beaucoup" };
for (String name : names)
    System.out.println(name.toString());
```

Ainsi en Java, dans un parcours de **conteneur** d'objets de type **Type**, un **objet** de type **Type** peut servir de variable de boucle et donc être utilisé à la place d'un entier.

V. Conversion d'un objet de type String en un entier

On souhaite écrire un programme affichant la somme d'entiers pris en paramètre sur la ligne de commande.

Ce programme est décomposé en plusieurs méthodes :

12. Écrire une méthode qui prend un tableau de chaînes de caractères en argument et renvoie un tableau d'entiers de même taille contenant les entiers issus des chaînes de caractères.

La méthode statique **parseInt(String s)** de la classe **java.lang.Integer** permet de récupérer la valeur d'un entier stockée dans une chaîne de caractères.

13. Écrire une méthode qui prend un tableau d'entiers en argument et renvoie la somme de ceux-ci.

14. Écrire la méthode main qui utilise les deux méthodes précédentes pour afficher le tableau d'entiers ainsi que sa somme.

Pour afficher le tableau d'entiers, deux solutions sont à écrire: la première avec un **foreach**, la seconde avec une méthode adéquate de la classe **java.util.Arrays**.