

# TP1 JAVA

## Exercice 1

- 1) Écrire le programme suivant
- 2) Compiler le programme en utilisant le commande javac puis vérifier que le fichier .class correspondant existe bien
- 3) Exécuter le programme avec la commande java

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

## Exercice 2

- 1) Dans un premier temps, afficher le premier argument de la ligne de commande (dans notre exemple Voici). Que se passe-t-il si l'on ne passe pas d'argument lors de l'exécution du programme ? -Lorsqu'il n'y a pas d'argument le programme n'affiche rien.

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

- 2) Écrire une boucle affichant le contenu du tableau, sachant qu'en Java, les tableaux possèdent un champ (un attribut) length qui correspond à la taille du tableau.

```
public class PrintArgs {  
    public static void main(String[] args) {  
        for(int i=0;i<args.length;i++){  
            System.out.println(args[i]);  
        }  
    }  
}
```

- 3) Changer votre programme pour utiliser la syntaxe dite du 'for deux points', c'est à dire for(Type value: array)

```
public class PrintArgs {  
    public static void main(String[] args) {  
        for(String i : args){  
            System.out.println(i);  
        }  
    }  
}
```

## Exercice 3

1) Recopier le programme précédent et le compléter pour qu'il affiche le nombre saisi par l'utilisateur.

```
public class Calc {  
    public static void main(String[] args) {  
        Scanner scanner;  
        scanner = new Scanner(System.in);  
        int value;  
        value = scanner.nextInt();  
        System.out.println(value);  
    }  
}
```

2) Indiquer dans le programme où sont les variables et quel est leur type associé. Modifier le programme pour déclarer et initialiser les variables en une seule ligne - value est une variable de type int, scanner est une variable de type java.util.Scanner

```
public class Calc {  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
        int value = scanner.nextInt();  
        System.out.println(value);  
    }  
}
```

3) Pourquoi nextInt() n'est pas une fonction ? Qu'est nextInt() alors -nextInt() n'est pas une fonction car les fonctions ne dépendent pas d'un objet spécifique, nextInt() est une méthode

4) Expliquer la ligne : import java.util.Scanner; -import java.util.Scanner permet d'importer dans notre code une bibliothèque extérieure pour pouvoir utiliser ses méthodes 5) Modifier le programme pour qu'il demande deux entiers et affiche la somme de ceux-ci.

```
public class Calc {  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
        int value = scanner.nextInt();  
        Scanner scanner2 = new Scanner(System.in);  
        int value2 = scanner2.nextInt();  
        System.out.println(value+value2);  
    }  
}
```

6) Afficher en plus de la somme, la différence, le produit, le quotient et le reste.

```
public class Calc {  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
        int value = scanner.nextInt();
```

```

        int value2 = scanner.nextInt();
        scanner.close();
        System.out.println(value+value2);
        System.out.println(value-value2);
        System.out.println(value*value2);
        System.out.println(value/value2);
        System.out.println(value%value2);
    }
}

```

## Exercice 4

1) Dans un premier temps, créer un record Point (dans un fichier Point.java) avec deux composants x et y, tous les deux de type int. Quelle doit être la ligne de commande pour compiler le fichier Point.java ? -pour compiler le fichier Point.java on utilise : javac Point.java

```

public record Coordonnées(int x, int y) {
    public static void main(String[] args) {
    }
}

```

2) Écrire une méthode main qui prend les entiers x et y sur la ligne de commande (3 et 4 dans l'exemple ci-dessus), les convertit en valeurs entières et affiche celles-ci (la ligne x=3, y=4 dans l'exemple).

```

public record Coordonnées(int x, int y) {
    public static void main(String[] args) {
        int x=Integer.parseInt(args[0]);
        int y=Integer.parseInt(args[1]);
        System.out.print("x=");
        System.out.print(x);
        System.out.print(", ");
        System.out.print("y=");
        System.out.println(y);
    }
}

```

3) Que veut dire ``static'' pour une méthode ? -La methode peut être appelé sans avoir besoin de créer un instance de cette classe

4) Que se passe-t-il lorsque l'un des arguments n'est pas un nombre ? -lorsque l'un des arguments n'est pas un nombre, une exception est levée 5) Dans le main, ajouter des instructions pour créer un instance du record Point, avec le deux entiers x et y et afficher celui-ci.

```

public record coordonnées(int x, int y) {
    public static void main(String[] args) {
        int x=Integer.parseInt(args[0]);
        int y=Integer.parseInt(args[1]);
        System.out.print("x=");
        System.out.print(x);
    }
}

```

```

        System.out.print(", ");
        System.out.print("y=");
        System.out.println(y);

        var coordonnées = new coordonnées(x, y);
        System.out.print(coordonnées);
    }
}

```

6) On souhaite ajouter au record Point une méthode d'instance (une méthode non statique) nommée distance qui calcule la distance entre deux points sous forme d'un nombre à virgule flottante. Quels sont les paramètres et le type de retour de la méthode distance ? -la methode "distance n'a pas de paramètre et son type de retour est "double". Sachant que calculer la distance entre deux points est équivalent à utiliser Pythagore sur la distance en x et la distance en y entre les deux points et qu'il existe une méthode statique Math.sqrt qui renvoie la racine carrée d'un nombre, écrire la méthode distance et afficher dans le main la distance entre le point (x, y) et un point à l'origine (0, 0).

```

public record Point(int x, int y) {
    public double Distance(){
        return Math.sqrt(x*x+y*y);
    }
    public static void main(String[] args) {
        int x=Integer.parseInt(args[0]);
        int y=Integer.parseInt(args[1]);
        var point = new Point(x, y);
        System.out.println("x="+point.x+", y="+point.y);

        System.out.println(point);

        System.out.print("dist = ");
        System.out.print(point.Distance());
    }
}

```

## Exercice 5

1) Compiler (gcc pascal.c) et exécuter le programme a.out en demandant au système le temps d'exécution du programme. (time a.out). -les temps d'exécution sont : time real/ 0,194s user/ 0,177s sys/ 0,017s

```

#include <stdio.h>
#include <stdlib.h>

int pascal (int nBut, int pBut){
    int * tab;
    unsigned int n, i;

    tab = (int *)malloc ((nBut+1)*sizeof(int));
    if(tab==NULL){

```

```

    fprintf(stderr, "Pas assez de place\n");
    exit(0);
}

tab[0] = 1;

for(n=1; n<=nBut; n++){
    tab[n] = 1;

    for(i=n-1; i>0; i--){
        tab[i] = tab[i-1] + tab[i];
    }

    int result=tab[pBut];
    free(tab);
    return result;
}

int main(int argc, char * argv[]) {
    printf(" Cn, p = %d\n", pascal (30000, 250));
    return 0;
}

```

2) Écrire le programme (Pascal.java) équivalent en Java. Pour une fois, servez-vous du copier/coller. Compiler le programme puis l'exécuter en mesurant le temps (toujours avec time). -les temps d'exécution sont : time real/ 0,194s user/ 0,177s sys/ 0,017s

```

public class Pascal {

    public static int methodePascal(int nBut, int pBut){

        int tab[];
        tab = new int[nBut+1];

        int n, i;
        tab[0] = 1;

        for(n=1; n<=nBut; n++){
            tab[n] = 1;

            for(i=n-1; i>0; i--){
                tab[i] = tab[i-1] + tab[i];
            }

            int result=tab[pBut];
            return result;
        }

        public static void main(String[] args) {
            System.out.print(" Cn, p = ");
            System.out.print(methodePascal(30000, 250));
        }
    }
}

```

```
}  
}
```