

Langages de programmation

1. Que va afficher à la console le programme ci-dessus ?

```
int a = 3, b = 2;
```

```
a = a + b;
```

```
a = a+++1;
```

```
System.out.println(a);
```

a. 5 b. 6 c. 7 d. 8

2. Que va afficher à la console le programme ci-dessus ?

```
int a = 3, b = 2;
```

```
a = a + b;
```

```
a = ++a+1;
```

```
System.out.println(a);
```

a. 5 b. 6 c. 7 d. 8

3. Que va afficher à la console le programme ci-dessus ?

```
int a = 5;
```

```
for (int i = 0; i < 3; i++) {  
    a = a++;  
    System.out.println(a);  
}
```

4. Que va afficher à la console le programme ci-dessus ?

```
int a = 5;
```

```
for (int i = 0; i < 3; i++) {  
    a++;  
    System.out.println(a);  
}
```

5. Qu'est-ce imprime à l'écran le programme suivant?

```
int a = 2;
```

```
String b = "3";
```

```
int c = 5;
```

```
int d = 7;
```

```
System.out.println(a + c + b + d);
```

Considérez la matrice suivante



00	01	02	03	04
10	11	12	13	14
20	21	22	23	24
30	31	32	33	34
40	41	42	43	44

Quelle est la condition pour sélectionner des éléments sur la diagonale principale?

Considérez la matrice suivante

00	01	02	03	04
10	11	12	13	14
20	21	22	23	24
30	31	32	33	34
40	41	42	43	44

Quelle est la condition pour sélectionner les éléments ci-dessus et en dessous de la diagonale principale?

Considérez la matrice suivante



00	01	02	03	04
10	11	12	13	14
20	21	22	23	24
30	31	32	33	34
40	41	42	43	44

Quelle est la condition pour sélectionner des éléments sur la diagonale secondaire?

Considérez la matrice suivante

00	01	02	03	04
10	11	12	13	14
20	21	22	23	24
30	31	32	33	34
40	41	42	43	44

00	01	02	03	04
10	11	12	13	14
20	21	22	23	24
30	31	32	33	34
40	41	42	43	44

Quelle est la condition pour sélectionner les éléments ci-dessus et en dessous du diagonale secondaire ?

La lecture du clavier Scanner

1. Créer un objet de type Scanner:

```
Scanner sc = new Scanner(System.in);
```

2. Prendre le contenu qu'on veut lire:

```
String str = sc.nextLine();
```

La classe Scanner fonctionne avec plusieurs types de variables:

```
Scanner sc = new Scanner(System.in);
```

```
int i = sc.nextInt();
```

```
double d = sc.nextDouble();
```

```
long l = sc.nextLong();
```

```
byte b = sc.nextByte();
```

```
String str = sc.nextLine();
```

Exercice 1

Lisez un tableau à partir du clavier et l'affichez.

Considérez la matrice suivante

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

00	01	02	03	04
10	11	12	13	14
20	21	22	23	24
30	31	32	33	34
40	41	42	43	44

Exercice 2

Écrivez une application qui va afficher la somme des chiffres pour chaque zone hachurée . Utilisez une fonction pour chaque zone hachurée.

```
int[][] A = {  
    {1, 2, 3, 4, 5},  
    {6, 7, 8, 9, 10},  
    {11, 12, 13, 14, 15},  
    {16, 17, 18, 19, 20},  
    {21, 22, 23, 24, 25},  
};
```

$$2 + 3 + 4 + 8 = 17$$

$$18 + 22 + 23 + 24 = 87$$

$$10 + 14 + 15 + 20 = 59$$

$$6 + 11 + 12 + 16 = 45$$

Exercice 3

Écrivez une fonction qui vérifie si un mot est palindrome ou non; e.g. “ana” est palindrome;

```
public static boolean palindrome(String str)
```

Exercice 4

Créer un programme qui déplace tous les 0 d'un tableau à la fin. Maintenez l'ordre relative des autres éléments. Utilisez une fonction pour changer le tableau. Ex: {0, 0, 2, 0, 3, 5, 7, 0, 9, 0} devient {2, 3, 5, 7, 9, 0, 0, 0, 0, 0}

Exercice 5

Écrivez un programme qui contient une méthode merge qui prend comme paramètre deux tableaux d'entiers int d'une dimension (tab1 et tab2) de longueurs quelconques.

Par exemple,
l'appel merge pour {1, 2, 3, 4} et { 5, 6} renvoie le tableau [1, 5, 2, 6, 3, 4].

Exercice 6

Ecrivez un programme qui pour un n donné (entier positif) détermine si n est premier ou non.

Exercice 7

Ecrivez un programme qui pour un n donné (entier positif) génère et imprime tous les nombres premiers inférieurs à n .

Théorie

Les tableaux dynamiques et hétérogènes - ArrayList:

- Combinaison entre listes et tableaux => les avantages des deux.
- Def: liste dynamique – on peut modifier le nombre d'éléments ajoutés dans la liste
- Def: liste hétérogène – on peut avoir plusieurs types de données dans la liste

Declaration:

```
ArrayList <Integer> list = new ArrayList<Integer>();
```

Théorie

On doit importer le paquet:

```
import java.util.ArrayList;
```

Ajouter des éléments dans l'ArrayList:

```
list.add(10);
```

```
list.add(15);
```

```
list.add(20);
```

Obs: on peut aussi ajouter un élément en spécifiant aussi la position (l'index).

Théorie

Fonctions utiles:

Dimension:

```
int count = list.size();
```

Parcours des elements:

```
for (int i = 0; i < list.size(); i++) {  
    int value = list.get(i);  
    System.out.println("Element: " + value);  
}
```

Exemple

```
public static String plusLongNom(ArrayList<String> list) {
    String nom = "";
    int longueur = nom.length();

    for(int i = 0; i < list.size(); i++) {
        if(list.get(i).length() > nom.length()) {
            nom = list.get(i);
            longueur = list.get(i).length();
        }
    }

    return nom;
}

public static void main(String[] args) {
    ArrayList<String> etudiants = new ArrayList<String>();
    etudiants.add("Martin");
    etudiants.add("Bernard");
    etudiants.add("Dubois");
    etudiants.add("Thomas");

    System.out.println(plusLongNom(etudiants));
}
```

Devoir

Exercice 1

Modifiez deux exercices des 7 premiers laboratoires et ajoutez une nouvelle tâche (ou changez une tâche).