Les éléments de la programmation Java

Remarque:

Que signifie:

public static void main(String[] args) { }

- **public**: la méthode main est accessible depuis l'extérieur de la classe.
- **static**: Le système Java peut appeler la méthode main sans créer une instance de la classe.
- **void**: la méthode main ne renvoie pas de valeur.
- **main:** le nom de la méthode
- **(String[] args):** le nom et le type des arguments de la méthode. Il est utilisé pour passer des arguments en ligne de commande au programme Java.

Les éléments de la programmation Java

- **Niveau de visibilité/ contrôle de visibilité/modificateurs d'accès**

Visibilité des classes:

- **Visibilité par défaut:** La classe n'est visible que de son propre paquetage.
- **Classe publique:** La classe est visible de partout.
- **Classe privée:** une classe privée doit être déclarée comme classe interne dans une autre classe externe ***publique***.

Les éléments de la programmation Java

- **Niveau de visibilité/ contrôle de visibilité/modificateurs d'accès**

Visibilité des champs (attributs et méthodes):

- **Privé (private):** le champ est accessible uniquement dans sa propre classe.
- **Protégé (protected):** le champ est accessible dans son package ou depuis les sous-classes des autres paquets (si sa classe est publique).
- **Publique (public):** le champ est accessible dans tout son package et si sa classe est publique, il est accessible partout.

Les éléments de la programmation Java

- **Getters et Setters**

Pour accéder aux propriétés privée (**private)** d'une classe dans une autre classe, on utilisera des getters et des setters;

- Les getters nous permettent d'accéder au contenu des propriétés privées.
- Les setters nous permettent de modifier les contenus des propriétés privées.

Les éléments de la programmation Java

Getters et setters

Déclaration:

```
public <type> getAtt() {
        return att;
    }

public void setAtt(<type> att) {
        this.att= att;
    }
```

Les éléments de la programmation Java

Getters et setters

Exemple:

Créer une classe Personne avec les attributs privées suivants:

Nom, prenom et age. Ensuite, nous désirons accéder à ces attributs à travers la classe Main.

Les éléments de la programmation Java

Getters et setters

Solution de l'exemple:

Code la classe Personne:

```
package tp;
public class Personne {

private String nom;
private String prenom;
private int age;

public String getNom() {
return nom;
}
public void setNom(String nom) {
this.nom = nom;
}
public String getPrenom() {
return prenom;
}
public void setPrenom(String prenom) {
this.prenom = prenom;
}
public int getAge() {
return age;
}
public void setAge(int age) {
this.age = age;
} }
```

Les éléments de la programmation Java

Getters et setters

Solution de l'exemple:

Code la classe Main:

```
package tp;
public class Main {

public static void main(String[] args) {
// TODO Auto-generated method stub
Personne pers = new Personne();
pers.setNom("AITAALI");
pers.setPrenom("Nawal");
pers.setAge(32);
System.out.println("Je suis " + pers.getNom() + " " +
pers.getPrenom() + " et j'ai " + pers.getAge() + " ans");
}
}
```

Les éléments de la programmation Java

- **Les structures de contrôle**

La structure IF – ELSE

```
if (condition logique)
{
        //instructions
}
else
{
        //instructions dans les autres cas
}
```

Ecrire un programme java qui vérifie si un entier n est paire ou impaire et affiche le résultat

Les éléments de la programmation Java

- **Les structures de contrôle**

La structure SWITCH – CASE

```
switch (expression )
{
    case constatnte_1 :   Suite d'instructions;    break;
    case constatnte_2 :   Suite d'instructions;    break;
    default :   Suite d'instructions;
}
```

Ecrire un programme java qui affiche :
- « Très bien » si la note est 'A'
- « Bien » si la note est 'B'
- « Insuffisant » si la note est 'C'
- Un message d'erreur sinon

Les éléments de la programmation Java

- **Les structures de contrôle**

La boucle For

```
for (initialisation; condition; mise à jour de valeurs)
    {
        // instructions
    }
```

Exemple:

```
for (int i=0 ; i<10 ; i++) {
    System.out.println("La valeur de i est : " + i);
}
```

Les éléments de la programmation Java

- **Les structures de contrôle**

La boucle While

```
while (test logique)
{
    //instructions
}
```

La boucle DO – While

```
do {
    // instructions;
} while (test logique)
```

Les éléments de la programmation Java

- **Les structures de contrôle**

TP n°2: Application de strucutres de contrôle

- **Ex1 :** Ecrire un programme java qui vérifie si un entier n est pair ou impair et affiche le résultat
- **Ex2 :** Ecrire un programme java qui affiche :

  « Très bien » si la note est 'A'

  « Bien » si la note est 'B'

  « Insuffisant » si la note est 'C'

  Un message d'erreur sinon
- **Ex3 :** Afficher la valeur d'un entier « i » tant que sa valeur ne dépasse pas la valeur 10.

Les éléments de la programmation Java

- **Package et importation:**
  - Les packages offrent une organisation structurée des classes
  - La répartition des packages correspond à l'organisation physique
  - Les packages conditionnent les importations de classes
  - Les packages permettent la coexistence de classes de même nom
  - Les mots-clé associés sont « **package** » et « **import** »

Exemple:

```
Package Bibliothèque
    class Livre
    class Employé
Package Ecole
    Import Bibliothèque. Employé
```

Les éléments de la programmation Java

➢ La classe String

- String n'est pas un type primitif, c'est une classe
- Déclaration de deux String:

```
String s1, s2;
```

- Initialisation :

```
s1 = "Bonjour";
s2 = "le monde";
```

- Déclaration et initialisation :

```
String s1 = "Bonjour";
```

- Concaténation :

```
String s3 = s1 + " " + s2;
```

Les éléments de la programmation Java

➢ La classe String

- Longueur d'un objet String

  Méthode length() : renvoie la longueur de la chaîne

```
String str1 = "bonjour";
int n = str1.length(); // n vaut 7
```

- Sous-chaînes

  Méthode **String substring(int debut, int fin)**

```
String str2 = str1.substring(0,3); // str2 contient la valeur "bon"
```

Les éléments de la programmation Java

➢ La classe String

- Récupération d'un caractère dans une chaîne

Méthode char charAt(int pos) : renvoie le caractère situé à la position pos dans la chaîne de caractère à laquelle on envoie ce message.

```
String str1 = "bonjour";
char unJ = str1.charAt(3); // unJ contient le caractère 'j
```

- Modification des objets String
  - Les String sont inaltérables en Java : on ne peut modifier individuellement les caractères d'une chaîne.

```
str1 = str1.substring(0,3) + "soir"; /* str1 contient maintenant la chaîne "bonsoir" */
```

Les éléments de la programmation Java

➢ La classe String

Les chaînes de caractères sont des objets :

```
String str1 = "BonJour";
String str2 = "bonjour";
String str3 =   "bonjour";
boolean   a, b, c, d;
a = str1.equals("BonJour"); //a contient la valeur true
b = (str2 == str3); //b contient la valeur true
c = str1.equalsIgnoreCase(str2);//c contient la valeur true
d = "bonjour".equals(str2); //d contient la valeur true
```

Les éléments de la programmation Java

➢ La classe String

- Quelques autres méthodes utiles
  - **boolean startsWith(String str)** : pour tester si une chaine de caractère commence par la chaine de caractère str
  - **boolean endsWith(String str)** : pour tester si une chaîne de caractère se termine par la chaine de caractère str

```
String str1 =   "bonjour ";
boolean a = str1.startsWith("bon");//a vaut true
boolean b = str1.endsWith("jour");//b vaut true
```

Méthodes la classe String	Rôle
charAt(int)	renvoie le nieme caractère de la chaine
concat(String)	ajoute l'argument à la chaîne et renvoie la nouvelle chaîne
endsWith(String)	vérifie si la chaîne se termine par l'argument
equalsIgnoreCase(String)	compare la chaîne sans tenir compte de la casse
indexOf(String)	renvoie la position de début à laquelle l'argument est contenu dans la chaine
lastIndexOf(String)	renvoie la dernière position à laquelle l'argument est contenu dans la chaine
length()	renvoie la longueur de la chaine
replace(char,char)	renvoie la chaîne dont les occurrences d'un caractère sont remplacées
startsWith(String int)	Vérifie si la chaîne commence par la sous chaîne
substring(int,int)	renvoie une partie de la chaine
toLowCase()	renvoie la chaîne en minuscule
toUpperCase()	renvoie la chaîne en majuscule

Les éléments de la programmation Java

- **La classe String**

TP3: Utilisation de la classe String

Ecrire un programme en JAVA qui permet de:

- Créer deux chaînes de caractères et les initialiser par ton nom et prénom
- Vérifier si les deux chaines sont égaux
- Calculer la longueur des deux chaines
- Renvoyer le 2ème caractère dans votre nom et prénom
- Changer le contenu de ton nom en commençant par Mr, Mlle ou Mme.