

Attention :

Le nom du programme est identique au nom de la classe principale, Celle qui contient la méthode main()

PROCEDURE :

- + Lancer Eclipse, cliquer sur **workbench** (en haut à droite)
 - + **File > new > javaProject** nom : `ProjetJ1`
 - + Clic droit sur `ProjetJ1` **new > package** nom : `packageJ1`
 - + Clic droit sur `packageJ1` (à gauche) **> new > class** : nom : `nom de la classe principale`
- Cocher **public static void main**

Pour chaque exercice, on crée **new class** qui peut contenir plusieurs classes.

1. Exercice

Ecrire un programme Java avec une seule classe nommée **Bienvenue**, cette classe contient la méthode **main()**, ce programme permet d'afficher :

Bienvenue au cours de java

1. On affecte à la variable **nom** la valeur **Dubois**.
En utilisant la variable **nom**, afficher : **Bonjour, je m'appelle Dubois**

/*****

AVEC DEUX CLASSES

- + Afin d'appeler une méthode d'une classe, on instancie la classe grâce au mot clé **new** suivi du nom de la classe et se termine par une parenthèse, c'est une fonction spéciale appelée un **constructeur**
Cette instanciation crée une image de la classe en mémoire qu'on récupérera dans une **variable** de **type le nom de la classe**.

Toto toto = new Toto() ;

- + Les méthodes de la classe peuvent être appelées grâce à la variable d'instance **toto** :

Toto.methode() ;

2. Exercice 1

1. Reprendre l'exercice précédent, en considérant 2 classes.
 - a. La classe principale contenant la méthode **main()**, nommé **GestionBonjour** qui permet juste d'appeler les méthodes des autres classes.
Instancier la classe **Bonjour** afin d'appeler n'importe quelle méthode de sa classe.
 - b. L'autre classe qui permet d'affecter à la variable **nom** la valeur **Dubois**.
En utilisant la variable **nom**, afficher : **Bonjour, je m'appelle Dubois**

Affichage saisie : Affectation en dynamique

<div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;">Scanner</div> <div style="border: 1px solid black; padding: 5px;"> next() nextInt() nextFloat() </div>	→Chaine caractère →Entier →Réel
--	---------------------------------------

```

package packageJ1 ;
import java.util.Scanner ;           // en-tête du programme

Scanner scan = new Scanner (System.in);
System.out.println("Veuillez saisir un entier ") ;
int x = scan.nextInt() ;

```

If et switch/case

<pre> if (condition){ instruction1; else if (condition){ instruction2; } else{ instruction3; } </pre>	<pre> switch (variable){ case valeur1 : instruction; break ; . . . case valeurN : instruction; break ; default : instruction; break ; } </pre>
---	--

3. Exercice

Ecrire un programme qui demande à l'utilisateur sa note → next(), nextInt() ...

On affichera selon la valeur saisie une phrase du type : → if

Vous avez la moyenne

Vous n'avez pas la moyenne

4. Exercice : avec if

Ecrire un programme qui demande à l'utilisateur de choisir :

" une multiplication ou une addition "

De saisir :

"2 nombres entiers "

Chaque opération sera gérée dans des classes différentes,

Chaque fonction admet 2 arguments, ne retourne un résultat.

Afficher le résultat de la multiplication ou de l'addition.

pas pour une chaîne de caractères	Le bon syntaxe
a == "Paris"	a.equals("Paris")

5. Exercice bis: Test multiple → switch()

Reprendre le même exercice avec switch/case

6. Exercice : Test avec String

Ecrire un programme Java **GestionSaisiePays** qui demande à l'utilisateur de saisir une capitale parmi **Paris** et **Londres**.

Tout ceci sera géré dans la classe **SaisiePays** de méthode **saisie()**.

On affichera selon la saisie :

Vous êtes Français
 Vous êtes Anglais
 Ce pays n'est pas pris en compte

A tester	La vraie syntaxe
a == "Paris"	a.equals("Paris")

ville = ville.toUpperCase();	La méthode toUpperCase() permet de passer en majuscule
------------------------------	---