INSTITUT SUPERIEUR DES ETUDES TECHNOLOGIQUES DE NABEUL

TRAVAUX PRATIQUES N° 2 : LES CLASSES ET LES OBJETS

Classes : DSI **Enseignante : Sana REFAI**

1. Objectifs

Le but de ce TP est d'abord de se familiariser avec un IDE dédié au développement avec le langage Java en l'occurrence Eclipse, ensuite, d'apprendre à définir des classes contenant des attributs et des méthodes et à créer des objets et les manipuler.

2. Introduction

A partir de ce TP, nous allons travailler avec un IDE dédié à Java. Il existe plusieurs autres éditeurs pour Java, tels que JBuilder, NetBeans, IntelliJ, jDeveloper, …
Mais notre choix s'est porté sur Eclipse.
Il s'agit d'un environnement de développement intégré (Integrated Development Environment) open source créé par IBM en 2001. Son code est écrit en Java et son utilisation principale est le développement d'applications Java, mais il peut également être utilisé pour d'autres langages de programmation via des plug-ins (extensions).

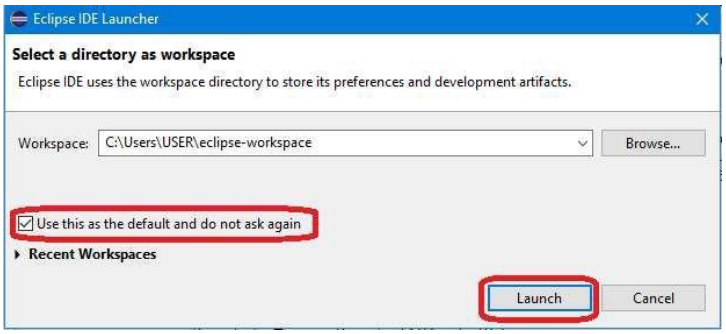
3. Débuter avec Eclipse

3.1 Téléchargement et exécution d'Eclipse

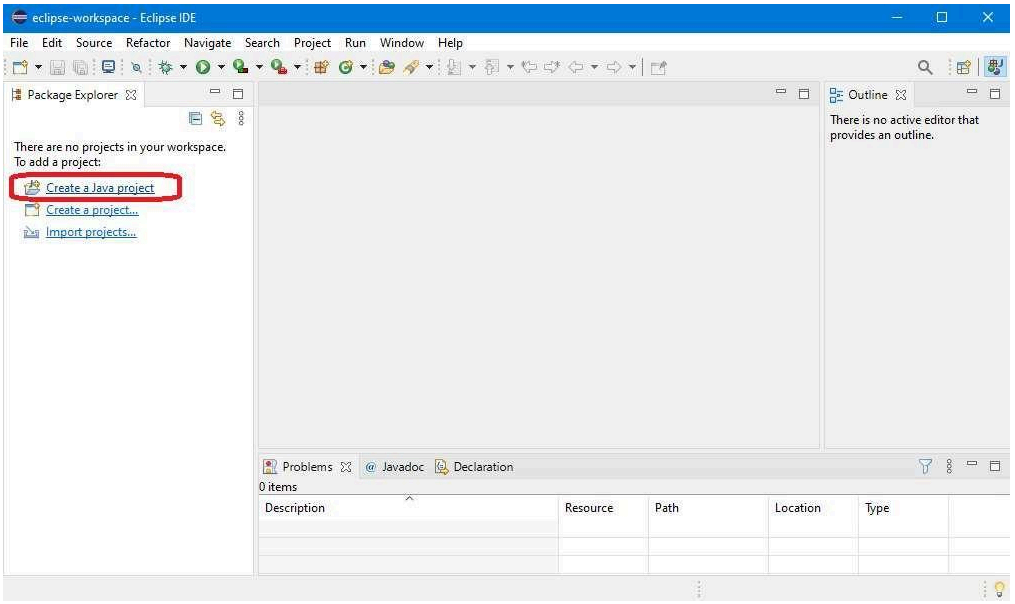
On commence par télécharger la dernière version en date intitulée « 2021-09 » qui est disponible sur le site officiel : https://www.eclipse.org/downloads/packages/.
Ensuite, il suffit de décompresser le fichier obtenu sous « c:\ ». Puis, ouvrir le dossier Eclipse et cliquer sur l'application Eclipse.exe ⇨ Aucun installation n'est requise.

3.2 Démarrage avec Eclipse

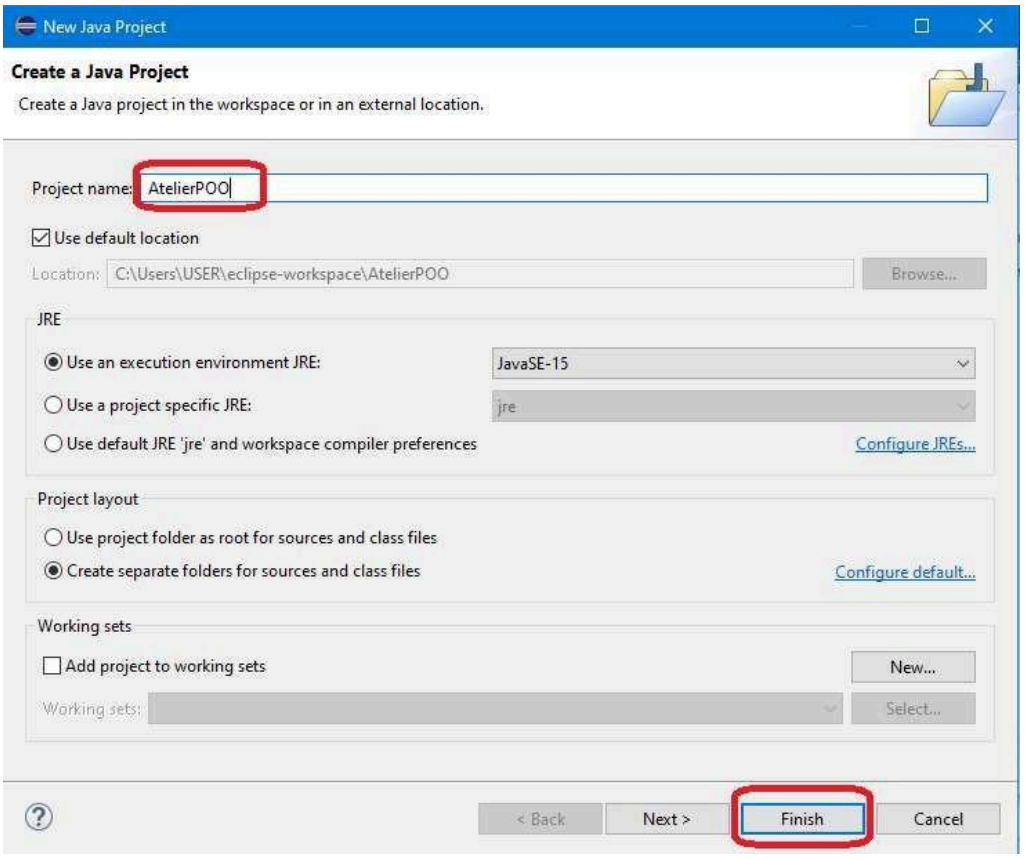
Dès la 1ère exécution, Eclipse demande à configurer le dossier du workspace.



Une fois le chemin du Workspace fixé, on commence par créer un projet Java.

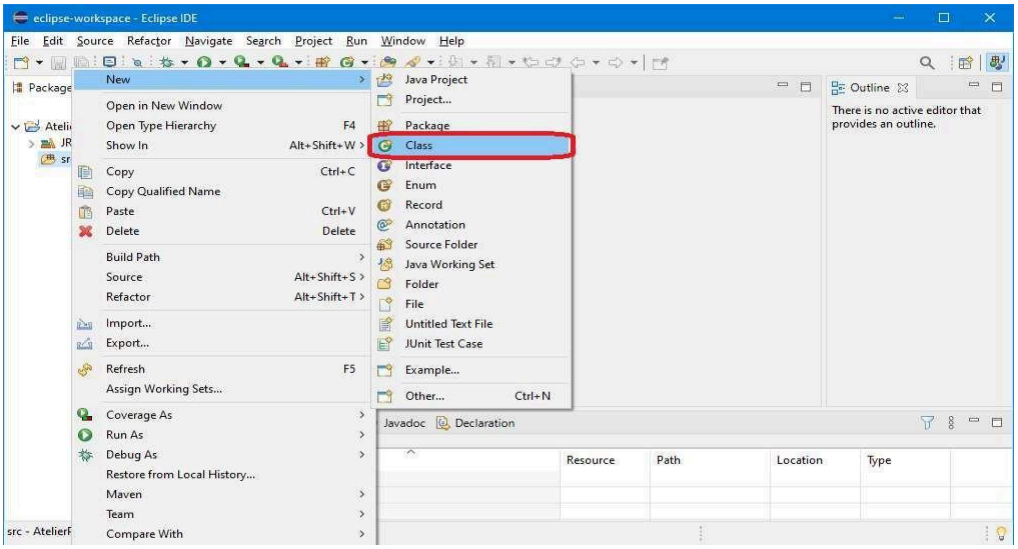
- 

Indiquez le nom du projet « AtelierPOO » puis cliquez sur le bouton « Finish » en bas afin de procéder à la création du dossier du projet.

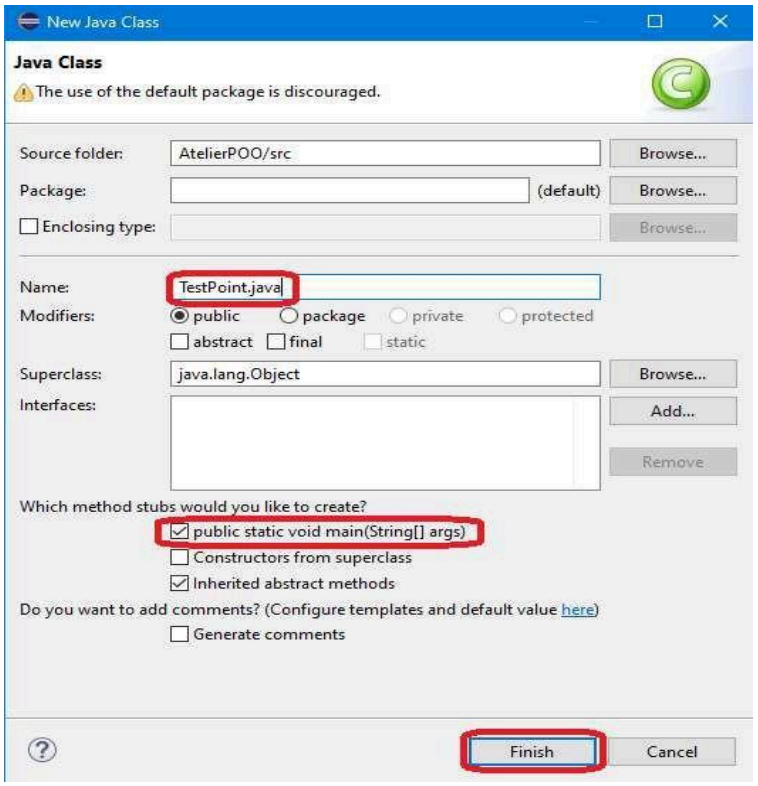
- 

Remarque : Par convention, les noms des projets doivent être en minuscule, néanmoins, il est possible de garder le nom qu'on a déjà indiqué.

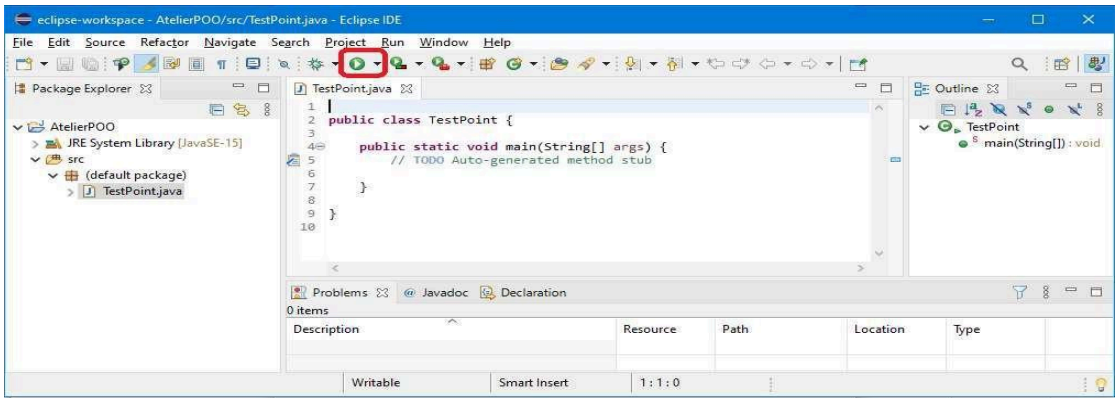
Une fois le projet créé, on passe à la création de fichiers java dedans. On fait un clic droit sur le dossier « src », on choisit New > Class



Il suffit d'indiquer le nom du fichier à créer et cocher la case entourée afin de générer la méthode main (voir figure). Exemple, pour notre 1er exercice, nous allons créer un fichier intitulé « TestPoint.java ». Attention à bien respecter la casse (majuscule & minuscule) sinon vous aurez des problèmes de compilation.



Eclipse permet de créer le fichier et générer une partie du code. C'est à vous de compléter le programme et exécuter en cliquant sur le triangle vert en haut (voir figure)



4. Les Exercices

4.1 Exercice 1

1) Ecrivez le programme suivant :

```
class Point {
 private char id ;                    // identifiant du point
 private double abs ;                 // abscisse du point

 public Point (char c, double x) {     //
 constructeur
   id = c ;
    abs = x ;
  }
 public void affiche ()
 {        // A compléter          }
 public void translate (double dx)
 { abs += dx ; }
}
 public class TestPoint {
  public static void main (String args[]) {
     Point a = new Point ('A', 2.5) ;
     a.affiche() ;
     Point b = new Point ('B', 5.25) ;
     b.affiche() ;
     b.translate(2.25) ;
     b.affiche() ;
   }
}
```

2) Complétez le code pour que la méthode permette d'afficher un message sous la forme :

Point identifié par A d'abscisse 2.5

3) Exécutez ce programme, que fait-il ? Identifier les différentes parties du code.

4) Déterminez les attributs et les méthodes de la classe Point ainsi que leurs rôles.

4.2 Exercice 2

La classe Point est caractérisée par :

✔ Les attributs suivants privés :

- Nom de type chaîne caractères
- Abscisse de type entier
- Ordonnée de type entier

✔ Et les méthodes suivantes :

- 3 Constructeurs avec 3, 2 et 1 paramètres.
- La méthode void Affiche () qui permet d'afficher les coordonnées d'un point sous le format suivant :

nom (abscisse, ordonnée).

- La méthode void TranslHoriz (int a) qui permet de translater le point horizontalement.
- La méthode void TranslVert (int a) qui permet de translater le point Verticalement.
- La méthode void Translation (int a, int b) qui permet de translater le point dans les deux sens.
- La méthode boolean Coïncide (Point p) qui permet de tester si 2 points coïncident.
- La méthode String getNom() qui retourne le nom du point
- La méthode int getAbscisse() qui retourne l'abscisse du point
- La méthode int getOrdonnées() qui retourne l'ordonnée du point
- La méthode void setNom(String ch) qui modifie le nom du point

- La méthode void setAbscisse(int a) qui modifie l'abscisse du point
- La méthode void setOrdonnée(int a) qui modifie l'ordonnée du point

Travail demandé

1- Ecrire en Java la classe Point.

2- Ecrire et exécuter la classe Test Point qui permet de tester la classe Point et qui contient le code suivant :

```
public class Test_Point
{
	public static void main (String [] args)
	{
		point p1;
		p1 = new point (3, 5);
		point p2 = new point ("a");
		point p3 = new point ("b", 3,5);
		System.out.println("\n ----------------------\n");
		System.out.println("les points créés sont :");

		p1.Affiche ();
		p2.Affiche ();
		p3.Affiche ();

		System.out.println("\n
		---------------------------\n"); if
		(p1.Coincide(p3) == true)
			System.out.println("Les 2 points p1 et p3
			coïncident");
		else
			System.out.println("Les 2 points ne coïncident pas");

		System.out.println("\n ---------------------------\n");
		System.out.println("translation des point ");
		p1.TranslHoriz (4);
		p2.TranslVert (3);
```

```
                p3.Translation (5,2);
                p1.Affiche ();
                p2.Affiche ();
                p3.Affiche ();

                System.out.println("\n ---------------------------\n");
                System.out.println("modification des attributs des points") ;
                p1.setNom("SRI21");
                p2.setAbscisse(25);
                p3.setOrdonnée(50);
                p1.Affiche ();
                p2.Affiche ();
                p3.Affiche ();

                System.out.println("\n ---------------------------\n");
                System.out.println("utilisation des méthodes get");
                String x=p1.getNom();
                int y=p1.getAbscisse();
                int z=p1.getOrdonnée();
                System.out.println(" le nom du point p1 est : " + x);
                System.out.println(" son abscisse est : " + y);
                System.out.println(" son ordonnée est : " + z);
        }
}
```

Question :

Ajouter les instructions suivantes :

p3.Nom="Test" ;

p3.Abscisse=2 ;

p3.Ordonnée=4 ;

Qu'est-ce que vous remarquez ?

4.3 Exercice 3

Considérons une classe java appelée MaDate ayant les trois attributs suivants[1] :

- Jours : un attribut privé de type entier
- Mois : un attribut privé de type entier
- Année : un attribut privé de type entier

Travail demandé

1) Créez la classe MaDate
2) Définissez un constructeur avec 3 paramètres MaDate(int jour, int mois,int annee)
3) Définissez un constructeur avec 1 seul paramètres MaDate(int annee)qui permet d'initialiser l'année de la date courante
4) Générez les getters et les setters des trois attributs
5) Redéfinissez la méthode toString pour que nous puissions afficher une date sous format « jour/mois/année »

6) Ecrivez la méthode ajouterUnJour qui permet d'ajouter un jour à notre date et faire des modifications, si nécessaire, pour les deux attributs mois et année. Remarque : il faut traiter tous les cas.

7) Utilisez la méthode précédente pour écrire une méthode ajouterPluseursJours(int n), **n** étant le nombre de jours à ajouter à la date enregistrée dans les trois attributs
8) Définissez les méthodes ajouterUnMoi()s et ajouterUnAn().

9) Ecrire une méthode main dans laquelle vous allez

- Créer 3 Dates, l'une parmi elles est instancié avec le constructeur avec un seul paramètre
- Utiliser les setters pour modifier ou initialiser les attributs des dates
- Affichez les trois dates.
- Testez et affichez si d1 et d2 sont identiques.

- Ecrire un menu qui demande laquelle des opérations à effectuer
    - Ajout d'un jour
    - Ou ajout de plusieurs jours
    - Ajout d'un mois
    - Ajout d'une année

Remarque : utilisez la bibliothèque java.util.Scanner