

Eclipse IDE (Integrated Development Environment)

La spécificité d'Eclipse IDE (Integrated Development Environment) vient du fait de son architecture totalement développée autour de la notion de plugin (en conformité avec la norme OSGi) : toutes les fonctionnalités de cet atelier logiciel sont développées en tant que plug-ins qui seront ensuite regroupés dans des features et des catégories.

Ouvrez l'IDE Eclipse, on vous demandera de choisir l'espace de travail dans lequel vous allez sauvegarder vos projets.

Tapez le chemin du répertoire que vous avez créé pour désigner l'espace de travail (le cas échéant Eclipse le créera pour vous).

Eclipse affiche plusieurs icônes (Overview, Tutorials, Samples, What's New) qui donnent accès à différentes documentations. Sélectionner Workbench et vous allez vous retrouver dans votre espace de travail.

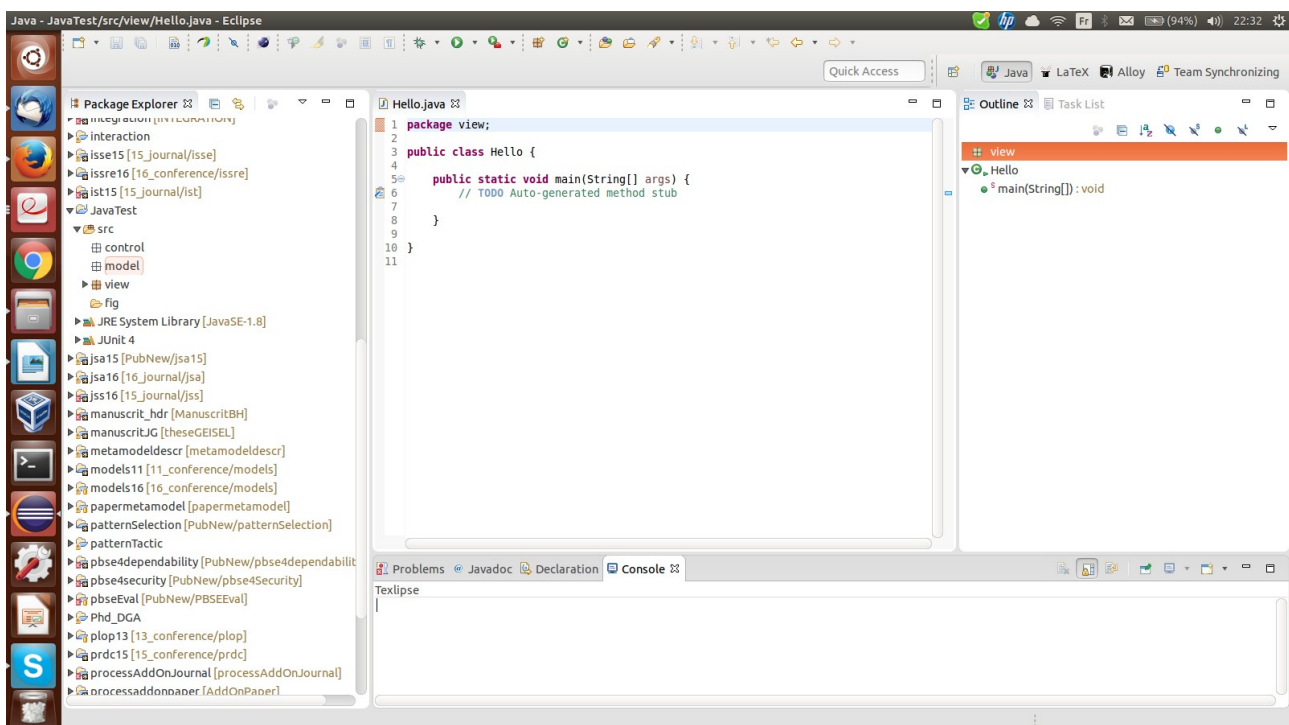
Une perspective décrit une configuration de travail d'Eclipse. Pour changer la perspective, on peut faire Window / Open Perspective. Un exemple de perspective est Java pour l'édition de programmes Java, Java Browsing pour la consultation de projet Java, Debug pour la mise au point de programmes...

Sélectionner successivement les perspectives Java Browsing, Debug puis Java et constater que les éléments affichés changent.

Exemple: La perspective Java

Elle est composée des éléments suivants :

- un Package Explorer (à gauche) : il liste les projets créés et leur contenu sous la forme d'une arborescence ;
- un Outline (à droite) : il donne la vue synthétique d'un élément par exemple, les attributs et les méthodes de la classe sélectionnée ;
- des onglets (Problems, javadoc et Declaration) dans la partie inférieure ;
- une partie centrale utilisée pour éditer les fichiers Java.



Créer un premier projet Java sous Eclipse :

Faire File / New / Project, sélectionner Java Project et faire Next. On peut alors saisir le nom du projet Hello, par exemple.

Cochez la case "create separate folders for source and class files" qui vous permet d'avoir un répertoire src pour vos sources et bin pour les fichiers compilés.

Faire Finish. Le projet est créé et apparaît dans l'explorateur (partie gauche d'Eclipse).

Pour créer une classe, on peut sélectionner le projet et choisir, ensuite New puis Class dans le menu contextuel. On doit alors donner le nom de la classe, ici Hello.

Vous pouvez générer automatiquement la méthode main en cliquant sur la case public static void main(String[] args).

Faire Finish. Le fichier Hello.java apparaît dans le projet (dans le paquetage par défaut) et dans l'éditeur (partie centrale d'Eclipse) comme sur la figure.

Le squelette de la classe est engendré (y compris la documentation).

Notons également que la classe Hello et la méthode main apparaissent dans le Outline.

La classe générée apparaîtra sous le package que vous avez choisi. Vous n'aurez qu'à terminer le reste du code à la main.

La compilation se fait en temps réel, après chaque sauvegarde par exemple.

De plus, les erreurs de syntaxe seront affichées pendant l'écriture du code, avec souvent des propositions de corrections.

Où se trouve votre projet sur l'ordinateur (espace de travail) ? Que contient les autres fichiers dans votre espace de travail ?

Refaire les mêmes instructions pour créer un autre projet contenant un package myPack contenant une classe Hello qui contient une méthode main.

Dans la méthode main, écrire System.out.println("Hello!") en oubliant le e de system et le ;

Corriger, compiler ensuite exécuter l'application :

Pour exécuter le programme, il est conseillé de sélectionner la classe que l'on souhaite exécuter (et qui doit contenir une méthode principale). On peut la sélectionner dans l'explorateur, l'éditeur ou le outline. Ici, nous n'avons que la classe Hello (qui contient un main).

En utilisant ensuite le menu contextuel, sélectionner Run As / Run. Sélectionner Java Application et faire New. Ceci provoque la création d'une nouvelle configuration pour l'exécution.

Elle porte le nom de la classe sélectionnée et la Main class est initialisée avec le nom de cette classe. Il suffit ensuite de faire Run pour lancer l'exécution.

Les résultats de l'exécution sont affichés dans l'onglet Console (en bas).

Chaque configuration correspond à une exécution particulière. Pour les exécutions suivantes, il suffit de choisir la configuration souhaitée et lancer son exécution.

Ajout d'arguments

Modifier le code de la méthode main pour avoir : System.out.println("Hello "+args[0]+"!");

Sélectionner Run Configurations, et dans le cadre Program Arguments de l'onglet Arguments tapez goodbye , ensuite exécuter.

Packages

Le package permet de structurer vos programmes, pour simplifier la lisibilité, la réutilisation et la maintenance.

Créez un package nommé control. Déplacez-y la classe Hello.
Observer la première ligne du fichier Hello.java

Utiliser une bibliothèque connue d'Eclipse :

Pour utiliser une bibliothèque existante, par exemple JUnit, on peut faire un clic droit sur le projet, sélectionner Build Path puis Add Libraries.... Dans celles proposées, il y a JUnit. La sélectionner et faire Next. Choisir la bonne version de JUnit (JUnit 4) et faire Finish.

Debug sous Eclipse

Rajouter une variable entière i initialisé à 0, puis une boucle permettant d'incrémenter cette variable de 2 à chaque itération pour atteindre 22.

Placez un point d'arrêt sur la ligne `i=i+2`

Lancez le code en mode debug.

Regardez les valeurs de cette variable. Utilisez la pile d'appel pour remonter d'un cran et regarder les valeurs de la variable.