

Utilisation d'Eclipse

Ce document présente le minimum à connaître pour développer en Java avec Eclipse¹, une plate-forme de développement libre et gratuite, développée dans le cadre d'une association fondée par IBM.

Exercice 1 : Démarrage Eclipse

Voici quelques éléments à connaître pour lancer Eclipse.

1.1 Lancer Eclipse. Pour lancer Eclipse à l'ENSEEIH, taper `eclipse-jee` dans un terminal. La dernière version de la plate-forme Eclipse se lance alors.

Lors du premier lancement, Eclipse affiche plusieurs icônes (*Overview*, *Tutorials*, *Samples*, *What's New*) qui donnent accès à différentes documentations. Sélectionner *Workbench*. Le *workbench* correspond à l'espace de travail dans lequel seront conservés les projets créés sous Eclipse. Il correspond à un répertoire dans le système de gestion de fichiers. On peut garder le répertoire proposé (*workspace* dans le répertoire racine). La fenêtre de la plate-forme Eclipse apparaît.

1.2 Sélectionner une perspective. Une perspective décrit une configuration d'Eclipse, c'est-à-dire les éléments affichés par Eclipse. Le nom de la perspective courante est affiché en haut à droite.

Pour changer la perspective, on peut faire *Window / Open Perspective*. On peut également cliquer sur l'icône à côté du nom de la perspective courante. Les perspectives disponibles incluent *Java* pour l'édition de programmes Java, *Java Browsing* pour la consultation de projet Java, *Debug* pour la mise au point de programmes...

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

1.2.2 Sélectionner la perspective *Java*. Elle est composée des éléments suivants (figure 1) :

- 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.

Exercice 2 : Faire un projet Java simpliste

Voyons comment créer un projet Java, y ajouter une classe, la compiler et l'exécuter.

2.1 Créer un projet Java. Faire *File / New / Project*², sélectionner *Java Project* et faire *Next*. On peut alors saisir le nom du projet *bonjour*, par exemple. Faire *Finish*. Le projet est créé et apparaît dans l'explorateur (partie gauche d'Eclipse).

1. <http://www.eclipse.org>

2. On peut aussi utiliser le menu contextuel (clic droit) de l'explorateur et faire *New / Project*.

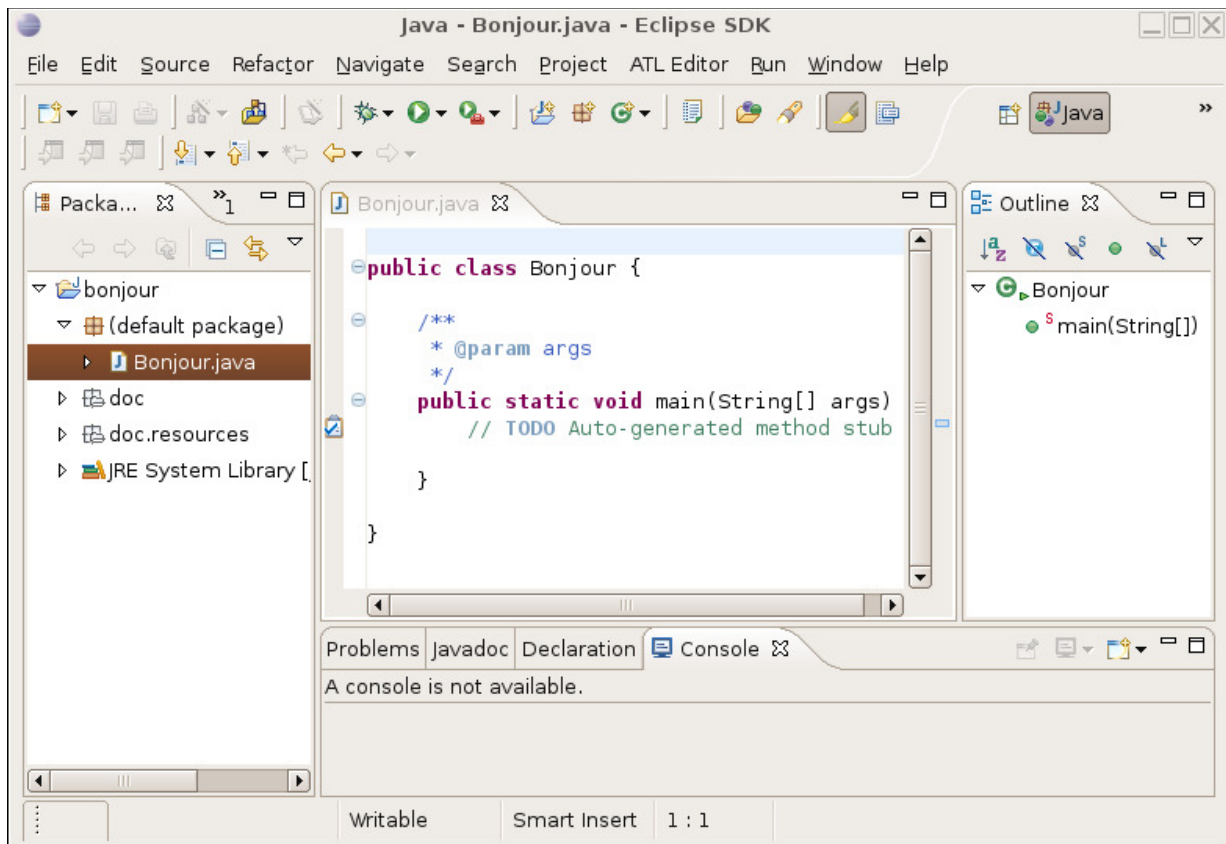


FIGURE 1 – Eclipse 3.1 avec la perspective Java.

2.2 Créer une classe Bonjour. Pour créer une classe, on peut sélectionner le projet et choisir *New* puis *Class* dans le menu contextuel. On doit alors donner le nom de la classe, ici *Bonjour*. Comme cette classe contiendra une méthode principale, on peut cocher l'option correspondante. Faire *Finish*. Le fichier *Bonjour.java* apparaît dans le projet (dans le paquetage par défaut) et dans l'éditeur (partie centrale d'Eclipse) comme sur la figure 1.

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

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

Remarque : il est possible de sélectionner le bouton marqué **C** de la barre de boutons pour construire une classe.

2.3 Compléter le code de main. Écrivons maintenant le code de la méthode *main*.

2.3.1 Écrire le code de la méthode *main* avec l'instruction suivante (en oubliant le « s » de *System* et le point-virgule final).

```
Sytem.out.println("Bonjour")
```

Lors de la sauvegarde, Eclipse vérifie le programme et signale une erreur en mettant une croix dans la ligne. Un message d'erreur est affiché dans l'onglet *Problems*.

2.3.2 L'erreur indiquée indique le point-vigule manquant. Corriger l'erreur. Lors de la sauvegarde, une nouvelle erreur est détectée : *Sytem* n'est pas connu. Il manque le « s ». Corriger

l'erreur. Il ne doit plus y avoir d'erreur !

2.4 Compiler l'application. L'application est en fait compilée après chaque sauvegarde. Il n'est donc pas nécessaire de lancer explicitement une compilation.

2.5 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 *Bonjour*.

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.

Exercice 3 : Exécuter avec le raccourci

Une fois les configurations créées, il est possible de relancer l'exécution de la dernière configuration en cliquant sur la flèche « lecture » dans le bouton vert de la barre de boutons. En cliquant sur la flèche à droite, on peut sélectionner une autre configuration ou faire *Run* pour obtenir la boîte de dialogue consacrée aux configurations.

Exercice 4 : Importer les classes d'une archive

Pour intégrer dans un projet Eclipse les classes d'une archive contenant des fichiers source (.tar, .jar ou .zip), il faut commencer par créer un projet Java (s'il n'existe pas), sélectionner ce projet, puis dans le menu contextuel faire *Import*, sélectionner *General*, puis *Archive file*, faire *Next*, sélectionner l'archive (avec *Browse*), faire *Finish*. Les fichiers de l'archive ont été ajoutés au projet.

Attention : Si l'archive contient des répertoires, ils sont conservés. Ceci signifie qu'ils seront traités comme des paquetages dans le projet Java. Si le répertoire n'était là que pour regrouper les fichiers de l'archive, il faudra déplacer son contenu dans le paquetage par défaut (ou la racine du projet).

Exercice 5 : Faire du *refactoring* avec Eclipse

Le *refactoring* consiste à modifier une partie d'une application en répercutant les modifications sur l'ensemble de l'application. Par exemple, lorsque l'on change le nom d'une classe, il faut répercuter ce changement dans toutes les classes qui l'utilisent. Ce type de modification est pris en charge par Eclipse. Pour illustrer les possibilités du *refactoring*, nous allons modifier l'application qui affiche graphiquement un schéma mathématique avec des points nommés.

5.1 Renommer la classe *Point* en *Point2D*.

5.2 Renommer la classe *PointNommé* en *Point2DNommé*.

5.3 Ajouter un paramètre couleur aux constructeurs de *Point2D* et *Point2DNommé*.