

## Chapitre 2

# Prise en main d'Eclipse

### 1. De l'importance de s'organiser

Ce chapitre permet de se familiariser avec l'environnement de développement Eclipse.

Créer un logiciel implique de modifier des fichiers texte. Au fur et à mesure de cette création, les fichiers s'accumulent et il devient de plus en plus important de s'organiser afin de pouvoir retrouver facilement le code.

Pour cela, Java propose un système d'organisation par espace de noms ou package, qui permet de regrouper différentes classes. Conceptuellement, il est plus simple d'y penser comme à différents dossiers de l'ordinateur.

Au plus haut niveau d'organisation, le code d'un logiciel est organisé dans un projet par Eclipse. Le projet regroupe tous les codes nécessaires à la réalisation des fonctionnalités demandées. Chaque élément de code est ensuite regroupé dans des dossiers différents selon qu'ils soient utiles à la fonctionnalité proprement dite ou à leurs tests.

## 2. Premier projet

Pour se familiariser avec le **workbench**, une classe très simple est créée.

Il faut auparavant créer un projet puis un package, toute classe d'une application un tant soit peu professionnelle devant être rangée dans un package.

Pour rappel, un package est physiquement un dossier, et symboliquement un espace de nommage pour rassembler les classes d'une même fonctionnalité.

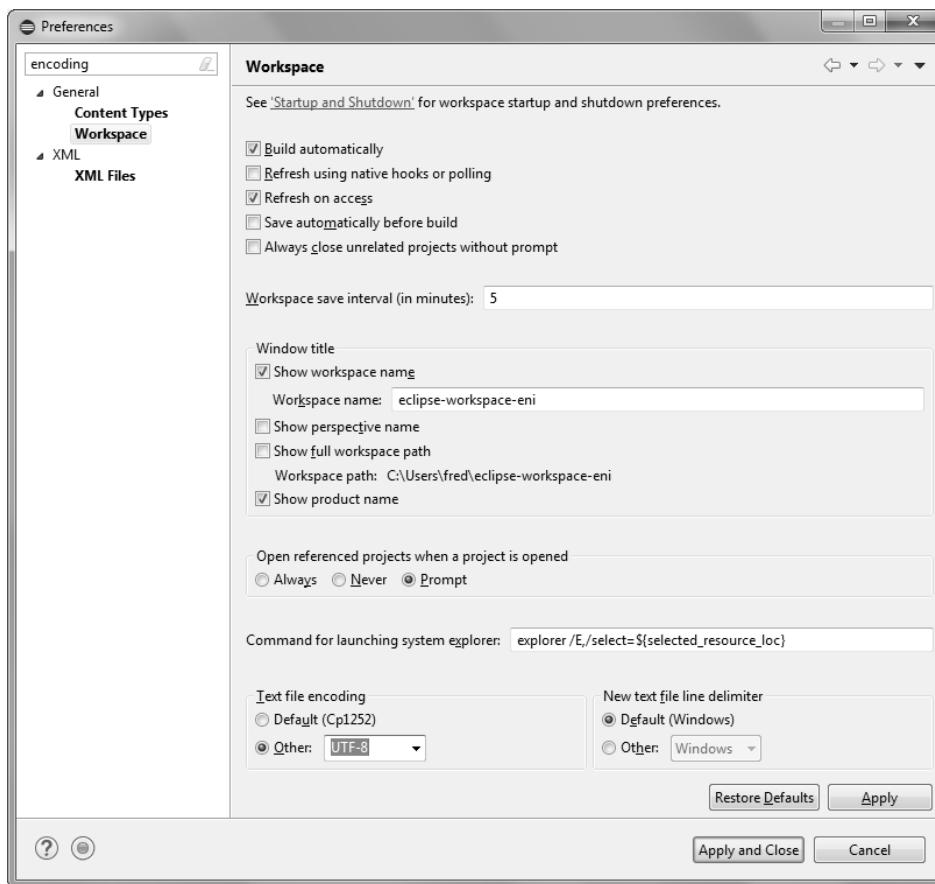
### ■ Remarque

*Si vous ne rangez pas vos classes dans un package, Eclipse les placera dans le package par défaut. L'utilisation du package par défaut est déconseillée.*

C'est également l'occasion de proposer une structure pour les fichiers du projet, afin d'améliorer la lisibilité et la testabilité des applications.

En préambule à ce travail, et afin de faciliter l'ouverture des fichiers provenant de systèmes d'exploitation différents, il convient de préciser l'encodage des fichiers.

■ Dans le menu, choisissez **Window - Preferences**. Naviguez ensuite dans **General - Workspace** et sélectionnez la valeur **Other - UTF-8** dans la section **Text file encoding**.

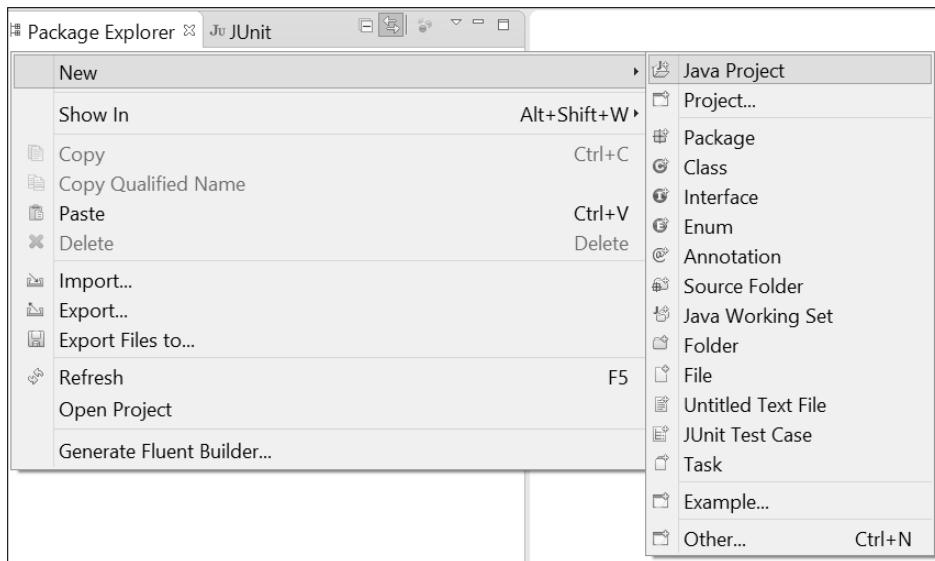


Ce réglage permettra de lire correctement les fichiers avec des caractères accentués dans Eclipse, que ce soit sur Windows, Linux ou Mac.

#### ■ Remarque

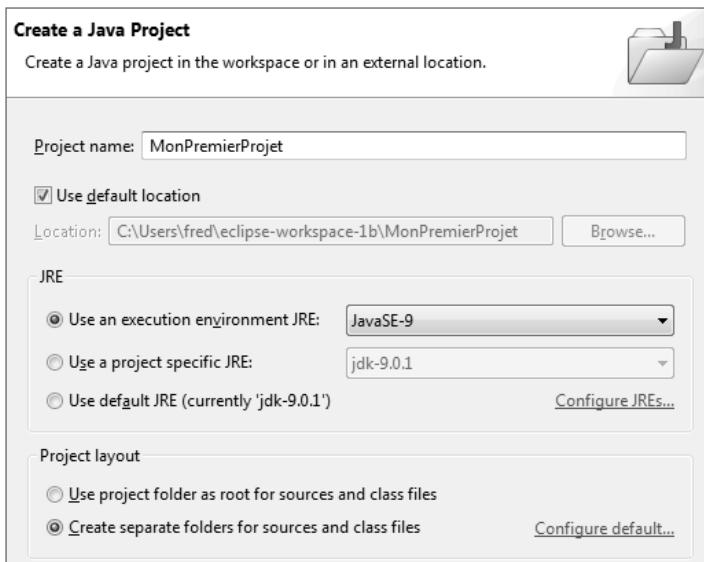
Les accents ainsi que les caractères spéciaux sont permis dans les noms des variables, mais ne sont pas recommandés. Vous comprendrez le jour où votre projet sera ouvert sur un ordinateur Linux, par exemple, ou quand il s'agira de travailler sur un projet avec les noms de variables en chinois, japonais, hébreu ou farsi.

- Dans le menu, choisissez **File - New - Java Project**, ou faites un clic droit dans la vue puis **New - Java Project**.



La boîte de dialogue de création de projet s'ouvre. Elle permet de nommer le nouveau projet et de choisir l'emplacement sur le disque dur où seront stockés les fichiers du projet (par défaut, les projets sont stockés dans le dossier d'espace de travail qui a été choisi lors de l'installation d'Eclipse).

- Saisissez le nom du projet, puis cliquez sur le lien **Configure default** en face du bouton radio **Create separate folders for sources and class files**.



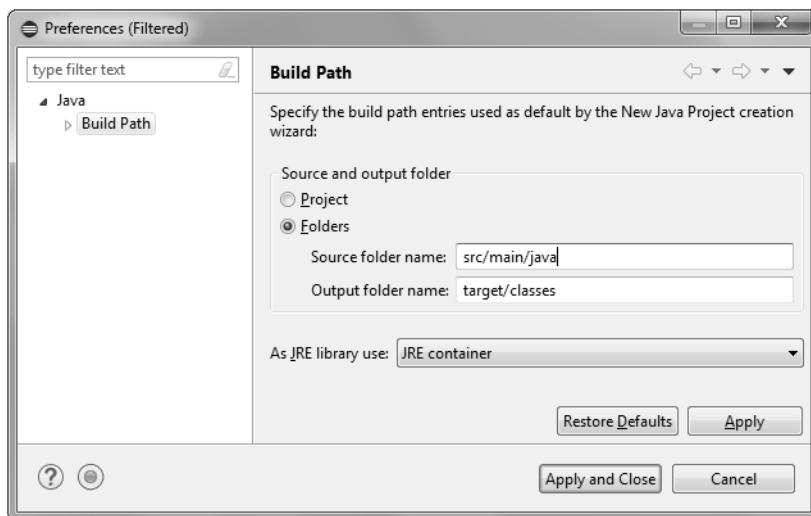
Cette manipulation permet de définir les dossiers dans lesquels sont rangés les fichiers sources et les fichiers binaires compilés. D'une manière générale, ces deux types de fichiers doivent être séparés. Cela permet par exemple de copier ou d'effacer facilement les binaires.

Les fichiers sources dans un projet Java ont souvent plusieurs sous-types : il existe des fichiers sources pour les classes de l'application, des fichiers sources pour les classes de tests, des fichiers de ressources telles que les images ou les configurations...

La manipulation suivante permet de définir des dossiers distincts pour chacun de ces types. Ces dossiers distincts sont inspirés des projets Maven (<http://maven.apache.org/>) et Gradle (<https://gradle.org/>), qui sont des outils puissants pour la gestion des projets informatiques, permettant de lancer en ligne de commande ou via une interface graphique la compilation des sources en binaires, le passage des tests unitaires, la transformation des classes en fichier jar, le paquetage dans un installateur et le déploiement sur un serveur, pour ne citer que quelques-unes des fonctionnalités disponibles.

Maven met en application l'adage « Une place pour chaque chose et chaque chose à sa place » et définit le répertoire **src/main/java** comme répertoire pour les fichiers sources, **src/main/resources** pour les fichiers de configuration et les images, **src/test/java** pour les sources de tests et **src/test/resources** pour les fichiers de configuration des tests. Les binaires sont stockés quant à eux dans **target/classes**.

- Changez les noms des dossiers comme dans l'écran suivant, puis validez.



Eclipse retourne sur la boîte de dialogue de création du projet.

- Cliquez sur **Finish** pour finaliser la création du projet.

Le projet et son contenu sont maintenant visibles dans la vue **Package Explorer**.

