

# Instructions simples et introduction à l'environnement Eclipse

## 1 Eclipse : quelques manipulations de base

Eclipse est un environnement de développement gratuit, écrit en Java, et dédié à Java, ainsi qu'à d'autres langages grâce à la notion de plugin que nous n'aborderons pas ici. Nous indiquons ici des manipulations de base. Si vous connaissez déjà Eclipse, lisez ce qui suit et assurez-vous que vous connaissez toutes les manipulations indiquées.

## 2 Lancer Eclipse

Lancez Eclipse Luna par le menu Application/Développement de votre bureau (clic droit souris). Vous êtes invité à préciser votre workspace. Le workspace correspond à l'espace de travail d'Eclipse, c'est en fait un répertoire où seront notamment stockés vos fichiers sources. Vous pouvez avoir plusieurs workspaces si vous le souhaitez.

### 2.1 Création d'un projet

2 solutions :

1. Dans le package explorer (partie gauche de la fenêtre) : clic droit → new → project
2. Dans le menu : File → New → Project

Un wizard s'ouvre.

- Choisissez Java Project (Next)
- Nommez le projet (*e.g.* TP1). Eclipse crée dans le workspace un répertoire de ce nom.
- Garder la case Use Default Location cochée
- Choisir Create Separate folders for sources and class files, afin de séparer vos fichiers sources (d'extension .java) des fichiers de bytecode (d'extension .class). Eclipse va créer dans votre projet un répertoire src et un répertoire bin
- Dans la partie JRE, vérifiez que la JRE est bien 1.7 ou 1.8.
- Si vous faites Next, vous arrivez à des configurations fines qu'il n'est pas nécessaire de modifier pour l'instant. (faites Finish)

Dans l'explorateur de gauche, on voit le nouveau projet avec un répertoire src, et la référence à la librairie système JRE (cliquer sur le triangle pour voir s'ouvrir votre projet et observer ce répertoire).

### 2.2 Création d'un package

Dans l'explorateur de gauche, sur l'icône src dans votre projet, faire un clic droit → new → package. Dans le wizard, donnez un nom à ce package. le nom commence usuellement par une minuscule, (par exemple : laPoste).

### 2.3 Création d'une classe

Dans l'explorateur de gauche, sur l'icône du packaging, faire un clic droit → new → class. S'ouvre un wizard "New Java Class".

- ne pas changer le source folder qui doit être correct, ni le nom du package, qui doit l'être aussi (vérifier)
- nommer la classe (ObjetPostal par exemple)
- ne pas toucher aux parties "superclass", "interfaces" pour le moment
- cocher la case `public static void main` si vous souhaitez mettre un `main` dans la classe.

Cela crée dans l'Explorer la classe à l'intérieur du package. Dans l'outline (à droite), s'ouvre un explorateur sur la classe faisant apparaître ses propriétés. Au centre, se trouve le code source de la classe. Vous pouvez contrôler ce qui a été généré (notez qu'il y a un embryon d'annotation pour la génération de la documentation). Notez un commentaire TODO qui signale qu'il faut compléter le code de la méthode main. Les TODO sont générés

automatiquement ou ajoutés par le programmeur. L'ensemble des tâches à faire peut être visualisé (menu Window puis Show view puis Tasks).

**Nota.** Si vous perdez des fenêtres et vous trouvez perdu, allez dans le menu "Window" et choisissez reset perspective, vous retrouverez le package explorer, la fenêtre de code centrale et la fenêtre d'outline à droite.

## 2.4 Quelques manipulations du code Java

Vous pouvez maintenant écrire le code de la classe. Vous remarquerez qu'il est directement indenté. Si vous copiez du code non indenté ou si vous voulez rafraîchir l'indentation : sélectionner la partie concernée, clic droit → source → correct indentation (ou ctrl +I).

**Eclipse corrige quelques erreurs de compilation** Créer une variable `clavier` de type `Scanner` :

- une petite croix apparaît en début de ligne pour signaler un problème (qui est explicité en passant la souris sur la croix),
- une petite lumière signale une solution,
- cliquer sur la lumière (clic gauche) et choisir la solution appropriée (ici : `import java.util.Scanner`).

**Eclipse crée les éléments (classes, méthodes, ...) nécessaires** Créer dans la classe `ObjetPostal` un attribut `destination` de type `Adresse`. La classe `Adresse` n'existant pas, Eclipse signale une erreur, et propose de la créer. Lancez effectivement la création de la classe `Adresse`. La classe `Adresse` est bien sûr vide. Eclipse peut également vous proposer de créer une méthode si vous faites appel à une méthode non encore développée. Dans ce cas, il génère juste l'entête de la méthode avec une note `TODO`, et pas son code, bien évidemment. Placez dans cette classe une méthode publique (par exemple : `getCodePostal`).

**Complétion sémantique** Définir la méthode `getCodePostalDestination` (`public float getCodePostalDestination()...`). Notez que la compilation se fait au fil de la saisie, et vous signale les erreurs en les soulignant en rouge. Quand on saisit un délimiteur ouvrant (ex. `{` ou `"`), il ajoute automatiquement le délimiteur fermant. Dans le corps de votre méthode, tapez `destination`. ("`destination`" suivi de `'`). La liste des méthodes définies pour l'objet `destination` est proposée ; il suffit de choisir celle qui nous intéresse (ici : `getCodePostal`) en double-cliquant dessus.

**Génération des accesseurs et des constructeurs** Créer un attribut dans la classe `ObjetPostal` `private float poids`. Sur le code, faire un clic droit → source → generate getters and setters. Les attributs sont proposés dans un wizard, où on choisit les attributs et les accesseurs désirés.

On peut générer des constructeurs à partir des attributs. Sur le code, faire un clic droit → source → generate constructors using fields. Sélectionner dans le wizard qui apparaît les champs que l'on veut prendre en paramètre. Pour l'instant, cocher la case *omit call to default constructor super()*.

**Renommage** Il existe une fonction de renommage des attributs et méthodes, classes, etc. Par exemple, on peut renommer "`destination`". Pour cela, dans l'outline, on fait un clic droit sur l'élément ("`destination`") → refactor → rename. Cela renomme toutes les occurrences de l'élément dans le code.

## 2.5 Exécution d'un programme

Finissez de développer votre premier programme. Pour exécuter le programme (qui contient un `main`), choisir menu run → run as → java application (ou utiliser l'icône avec la flèche blanche dans un disque vert).

Si la fenêtre "Console" n'apparaît pas en bas de votre environnement Eclipse avec le résultat de l'exécution, choisissez dans les menus Window -> Show View -> Console.