

## *Annexe B. Astuces Eclipse*

**I**l y a dans Eclipse de nombreuses petites commandes utiles qui rendent la programmation Java un peu plus rapide. Je te propose une liste de commandes Eclipse utiles, mais je suis sûr que tu en trouveras d'autres quand tu commenceras à utiliser cet outil.

- ✓ Si tu vois un petit astérisque dans l'onglet classe, cela signifie que des modifications du code de la classe n'ont pas été enregistrées.
- ✓ Met en surbrillance le nom d'une classe ou d'une méthode utilisée dans ton code et appuie sur la touche *F3* de ton clavier pour aller à la ligne où a été déclarée cette classe ou cette méthode.
- ✓ Si une ligne est marquée avec un rond rouge, symbole d'erreur, déplace la souris au-dessus du rond pour voir le texte de l'erreur.
- ✓ Appuie sur *Ctrl-F11* pour exécuter à nouveau le dernier programme exécuté.
- ✓ Place le curseur après une accolade pour marquer l'accolade correspondante.
- ✓ Pour modifier la superclasse lorsque tu crées une classe, clique sur le bouton *Parcourir*, supprime `java.lang.Object` et entre la première lettre de la classe que tu souhaites utiliser. Tu peux alors choisir ta classe dans une liste.
- ✓ Pour copier une classe d'un paquetage à un autre, sélectionne-la et appuie sur *Ctrl-C*. Sélectionne ensuite sa destination et appuie sur *Ctrl-V*.
- ✓ Pour renommer une classe, une variable ou une méthode, clique dessus avec le bouton droit de la souris et sélectionne *Propager*

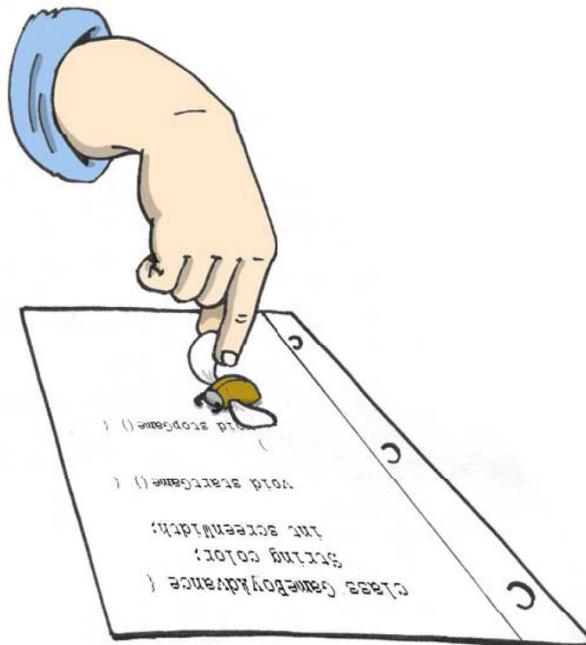
les modifications puis *Renommer* dans le menu contextuel. Toutes les occurrences de ce nom seront renommées.

- ✓ Si ton projet nécessite des fichiers jar externes, clique avec le bouton droit sur le nom du projet, sélectionne *Propriétés*, *Chemin de compilation Java*, choisis l'onglet *Bibliothèques* et clique sur le bouton *Ajouter des fichiers JAR externes*.

## Débogueur Eclipse

Selon la rumeur, il y a 40 ans, quand les ordinateurs étaient tellement gros qu'ils n'auraient même pas pu entrer dans ta chambre, l'un des programmes se mit soudain à donner des résultats erronés. Tous ces problèmes étaient le fait d'un petit insecte installé quelque part dans les câbles de l'ordinateur<sup>8</sup>. Quand on retira l'insecte, le programme se remit à fonctionner correctement. Depuis lors, *déboguer un programme* signifie trouver ce qui ne donne pas les résultats attendus.

Ne confonds pas les bogues et les erreurs de compilation. Disons par exemple que, au lieu de multiplier une variable par 2, tu la multiplies par 22. Cette erreur de frappe ne génère aucune erreur de compilation, mais le résultat sera faux. Les débogueurs te permettent de dérouler l'exécution d'un programme ligne à ligne et de voir ou modifier la valeur des variables à chaque étape de l'exécution du programme.



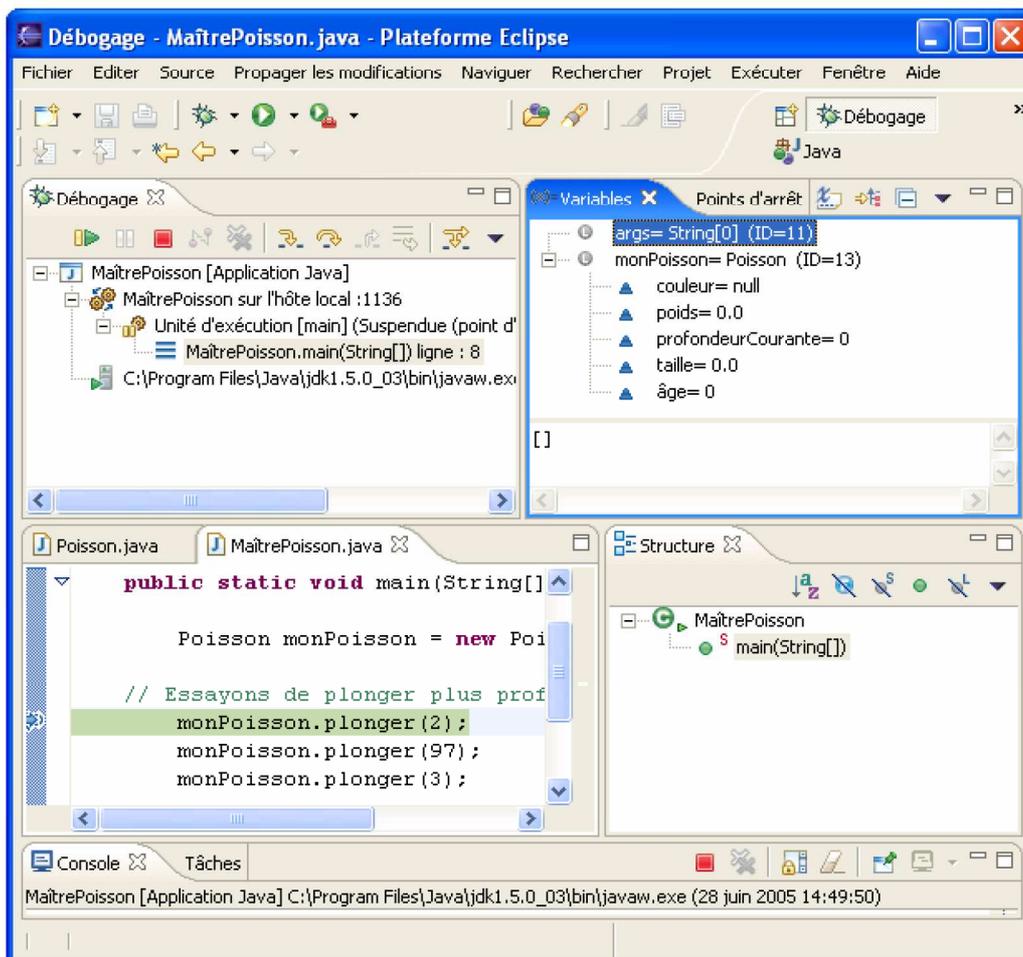
Je vais t'expliquer comment utiliser le débogueur d'Eclipse en prenant l'exemple du programme *MaîtrePoisson* du Chapitre 4.

<sup>8</sup> NDT : l'un des mots anglais pour insecte est *bug*.

Un *point d'arrêt (breakpoint)* est une ligne du code où tu voudrais que ton programme s'arrête afin de voir ou changer la valeur courante des variables et d'autres informations d'exécution. Pour définir un point d'arrêt, double-clique simplement sur la gauche de la ligne où tu veux que le programme s'arrête. Essayons sur la ligne `monPoisson.plonger(2)`. Tu peux voir sur cette ligne une puce ronde qui indique un point d'arrêt. Maintenant, sélectionne les menus *Exécuter, Déboguer...* Sélectionne l'application `MaîtrePoisson` et appuie sur le bouton *Déboguer*.

`MaîtrePoisson` démarre *en mode débogage (in the debug mode)* et s'arrête en attendant de nouvelles instructions de ta part dès qu'il atteint la ligne `monPoisson.plonger(2)`.

Une fenêtre comme celle-ci est affichée :

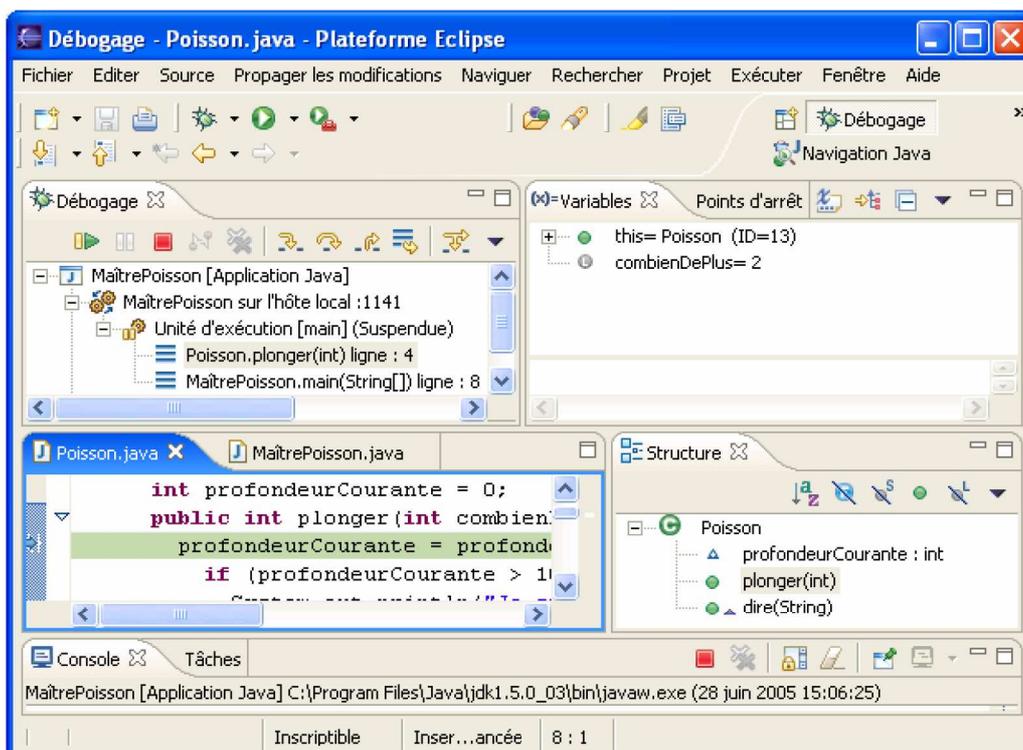


En bas à gauche de la perspective de débogage, la ligne correspondant au point d'arrêt est en surbrillance. La flèche bleue pointe sur la ligne qui est sur le point d'être exécutée. Sur le côté droit (dans la vue *Variables*), clique sur le petit signe + près de la variable `monPoisson`. Comme cette variable pointe sur l'objet `Poisson`, on peut examiner

toutes les variables membres de cette classe et leurs valeurs courantes ; par exemple, `profondeurCourante = 0`.

Les flèches de la partie haut gauche de la fenêtre permettent de poursuivre l'exécution du programme dans différents modes. La première flèche jaune signifie *Avance d'un pas avec entrée*, ou *entre dans (step into)* la méthode. Si tu appuies sur cette flèche (ou sur *F5*), tu te retrouveras à l'intérieur de la méthode `plonger()`. La fenêtre change et te présente la valeur de l'argument `combienDePlus = 2` comme dans la capture d'écran ci-dessous. Clique sur le petit signe + à côté du mot `this` pour voir quelles sont les valeurs courantes des variables membres de cet objet.

Pour modifier la valeur de la variable, clique dessus avec le bouton droit et entre la nouvelle valeur. Cela peut t'aider quand tu n'es pas sûr de comprendre pourquoi le programme ne fonctionne pas correctement et que tu veux jouer au jeu du *et si*.



Pour poursuivre l'exécution ligne à ligne, clique sur la flèche suivante, *Avance d'un pas sans entrée* (ou appuie sur le bouton *F6*).

Si tu veux reprendre l'exécution normale du programme, clique sur le bouton *Reprise* (petit triangle vert à gauche) ou appuie sur le bouton *F8*.

Pour supprimer le point d'arrêt, double-clique sur la petite puce ronde, qui disparaît. J'aime utiliser le débogueur même si mon programme ne

contient pas de bogue : ça m'aide à mieux comprendre ce qu'il se passe exactement dans le programme qui s'exécute.

Où placer un point d'arrêt ? Si tu as une idée de la méthode qui te pose problème, mets-le juste avant la ligne suspecte. Si tu n'en es pas sûr, place-le simplement sur la première ligne de la méthode `main()` et avance petit à petit dans le programme.