
Eclipse atelier Java

Table des matières

1. Introduction.....	2
2. Télécharger eclipse.....	3
3. Installer eclipse.....	3
4. Premier lancement d'eclipse.....	3
5. Configurer eclipse pour faire du Java.....	5
6. Développer un programme Java.....	6
7. Emplacement des fichiers sources.....	10
7. Enregistrer les fichiers sources dans un autre dossier.....	10
9. Déboguer les programmes.....	10

Eclipse est un projet de la Fondation Eclipse visant à développer un environnement de production de logiciels libre qui soit extensible, universel et polyvalent, en s'appuyant principalement sur Java. Son objectif est de produire et fournir des outils pour la réalisation de logiciels, englobant les activités de programmation.



1. Introduction

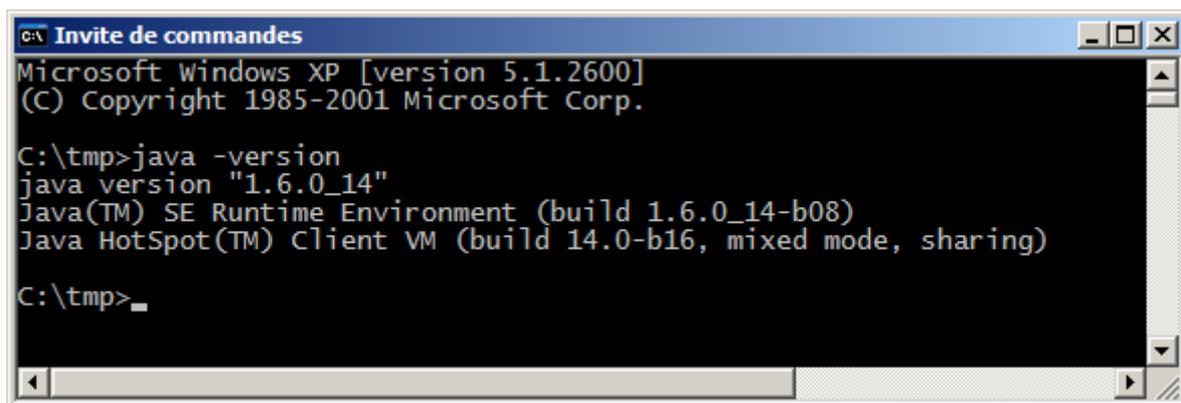
Eclipse ne contient ni le compilateur Java ni les autres outils basiques. Pour développer des programmes en Java il faut donc installer au préalable [un kit de développement](#). Nous conseillons celui de Oracle (anciennement Sun Microsystems, la maison mère de Java), qui est complet et gratuit.

Vous pouvez l'obtenir chez Oracle, section JDK Download. Le produit à télécharger s'appelle, lors de la publication de ce tutoriel, Java SE Development Kit 7u10.

Attention, ne confondez pas le JDK (Java Development Kit) avec le JRE (Java Runtime Environment), appelé aussi « plugin Java », qui ne contient que le nécessaire pour exécuter les programmes Java. Ne vous occupez pas de télécharger le JRE, à l'intérieur du JDK il y en a un exemplaire.

Le fichier téléchargé est un installateur auto-extractible : une fois chargé, il suffit de le lancer et de suivre les instructions qui s'affichent. Au besoin, des [informations sur l'installation du JDK](#) sont donnés sur le site de Oracle, aussi bien pour Windows que pour Linux.

Une fois l'installation terminée, vérifiez sa réussite en tapant « java -version » dans une console de commandes. Vous devez obtenir un message vous annonçant le numéro de version de la machine Java mise en place. Dans le cas de Windows cela ressemblera à ceci :



```
ca Invite de commandes
Microsoft Windows XP [version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\tmp>java -version
java version "1.6.0_14"
Java(TM) SE Runtime Environment (build 1.6.0_14-b08)
Java HotSpot(TM) Client VM (build 14.0-b16, mixed mode, sharing)

C:\tmp>
```

Pour développer des programmes en Java il vous faut disposer également de la documentation de l'API (Application Programmer Interface, c'est-à-dire le volumineux ensemble de paquetages, classes, méthodes et variables qui constituent la bibliothèque système). Vous pouvez la [consulter en ligne](#) ou bien la télécharger depuis le site de Oracle (pas loin du logiciel, voir plus haut).

NB : si vous souhaitez pouvoir employer le compilateur et les autres outils Java en dehors d'eclipse, c'est-à-dire en tapant des commandes dans une console Invite de commandes, alors vous devez procéder à la manipulation supplémentaire suivante : repérez le répertoire d'installation de Java, appelons-le RJAVA, et ajoutez le chemin RJAVA\bin dans la définition de la variable Path. Si vous avez laissé l'installateur de Java faire à sa guise, RJAVA doit être quelque chose comme C:\Program Files\Java\jdk1.7.0_10.

Vous pouvez examiner et modifier la valeur de la variable Path en cliquant avec le bouton droit sur l'icône du Poste de travail, puis Propriétés > Avancé > Variables d'environnement > Variables système ; sélectionner la ligne Path puis faire Modifier.

2. Télécharger eclipse

Eclipse est un logiciel libre que vous pouvez télécharger depuis le site www.eclipse.org, onglet Downloads. Le produit qui nous intéresse est Eclipse IDE for Java Developers (~150 MB).

Pour les systèmes « 64 bits », le fichier à télécharger se nomme :

- dans le cas de Windows : eclipse-java-juno-SR1-win32-x86_64.zip
- dans le cas de Linux : eclipse-java-juno-SR1-linux-gtk-x86_64.tar.gz
- dans le cas de Max OS X : eclipse-java-juno-SR1-macosx-cocoa-x86_64.tar.gz

Nous ne vous conseillons pas de télécharger une version française d'eclipse. Il peut exister des plugin de francisations de l'interface, mais outre le fait qu'elles sont assez imparfaites, elles servent surtout à vous empêcher d'utiliser la dernière version du logiciel.

3. Installer eclipse

Pour installer eclipse il suffit de décompresser l'archive zip ou tar.gz téléchargée. Cela crée un dossier, nommé eclipse, que nous vous conseillons de placer aussi haut que vous le pouvez dans la hiérarchie de fichiers de votre système. Dans la suite de cette note nous supposons que vous avez fait ainsi et que vous avez donc un dossier nommé C:\eclipse

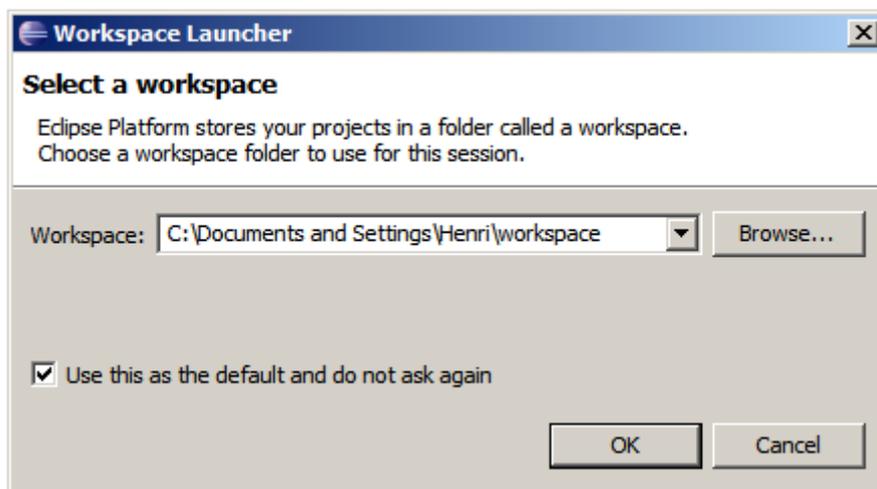
Pour faciliter le lancement d'eclipse créez un raccourci vers le fichier C:\eclipse\eclipse.exe et placez-le sur le bureau, dans le menu démarrer ou ailleurs, selon vos goûts.

NB : l'installation d'eclipse est donc bien plus légère que celle de beaucoup de logiciels ; en particulier, sous Windows elle ne produit pas des inscriptions dans la base de registres. Par conséquent, pour désinstaller complètement eclipse il suffira, le moment venu, de mettre à la corbeille le dossier C:\eclipse et le(s) espace(s) de travail (dossiers workspace, voir ci-dessous) créés ultérieurement.

4. Premier lancement d'eclipse

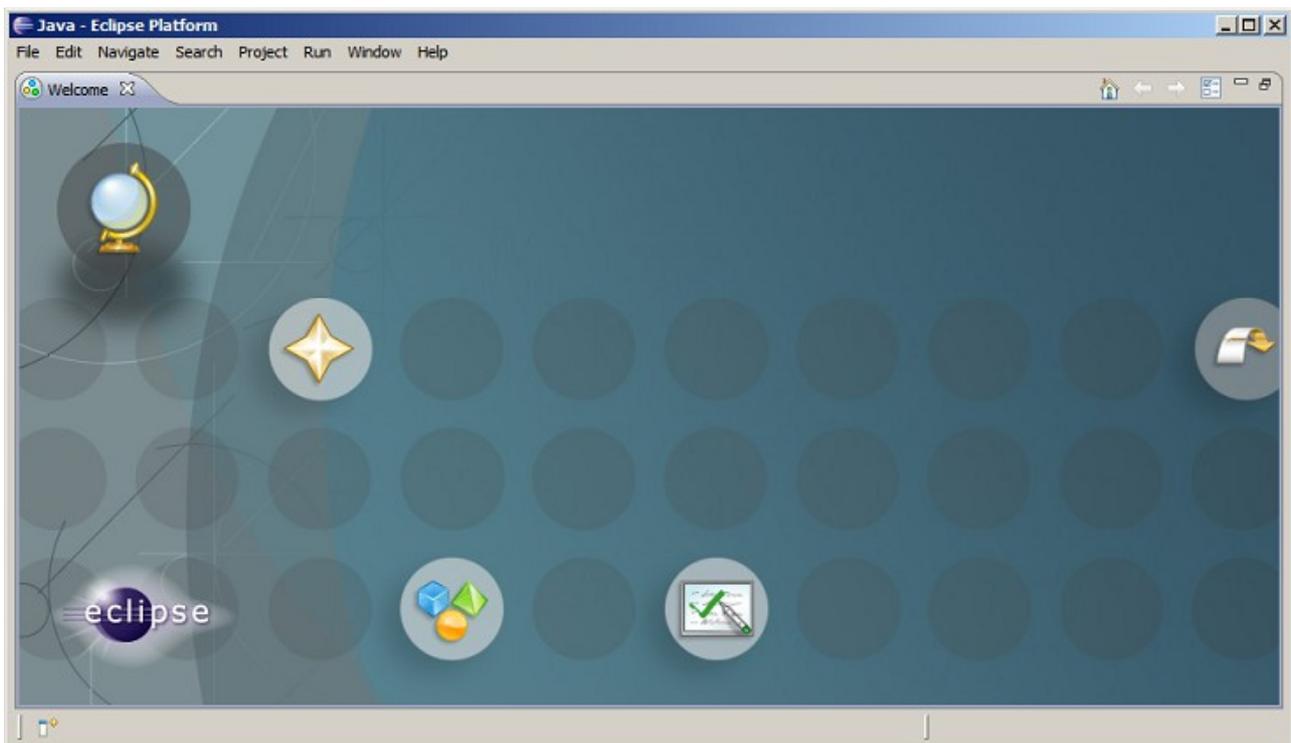
Lancez eclipse, par exemple en double-cliquant sur le raccourci que vous venez de créer. Au bout de quelques instants, on vous demandera de situer l'espace de travail dans lequel seront vos fichiers. Si vous travaillez sur un ordinateur partagé il est conseillé de mettre l'espace de travail dans votre dossier Documents. Si vous êtes le seul utilisateur de votre système, mettez l'espace de travail où bon vous semble.

Sauf indication contraire, les fichiers sources de vos programmes se trouveront dans l'espace de travail. Il est donc important de se souvenir de l'emplacement de ce dernier pour accéder aux sources (par exemple pour les sauvegarder, les transporter, etc.)

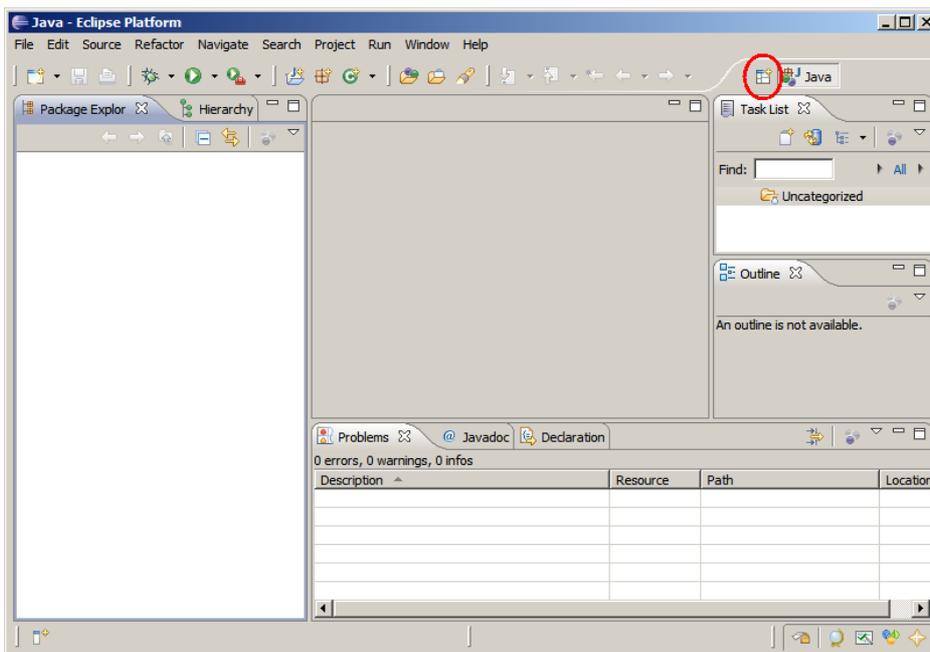


Si vous cochez la case « Use this as the default and do not ask again » eclipse ne vous posera plus cette question (mais il y a toujours un moyen pour changer ultérieurement l'espace de travail : File > Switch Workspace > Other...).

Au bout de quelques instants (la première fois ce n'est pas très rapide) vous obtenez un écran qui présente le produit, comme ceci :



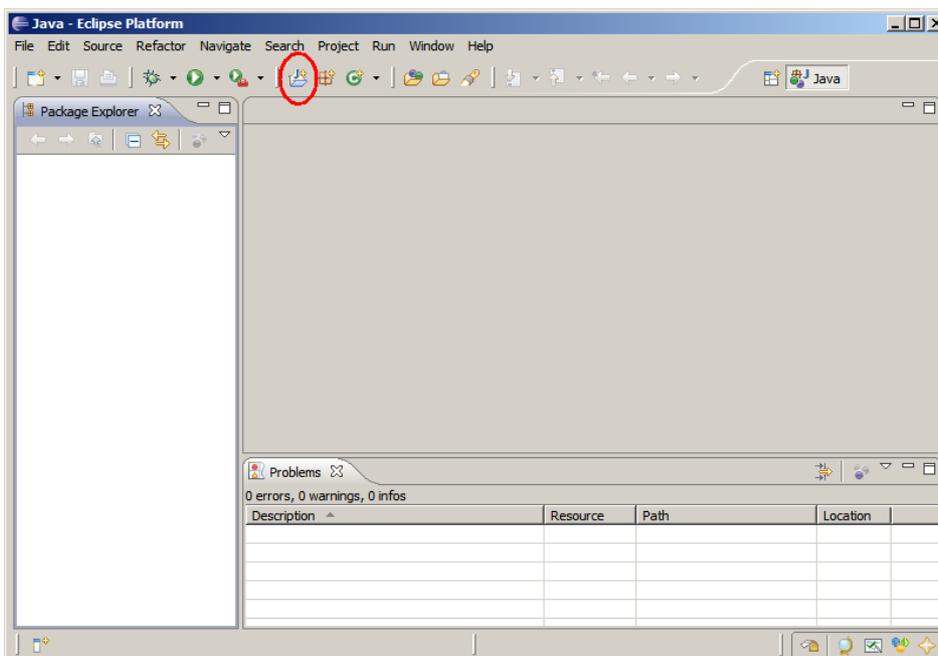
Vous pouvez feuilleter cette présentation, elle est faite pour cela. Quand vous en aurez assez, cliquez sur le lien Workbench (la flèche représentée à droite de l'écran). Le contenu de la fenêtre devient tout de suite beaucoup plus sérieux :



5. Configurer eclipse pour faire du Java

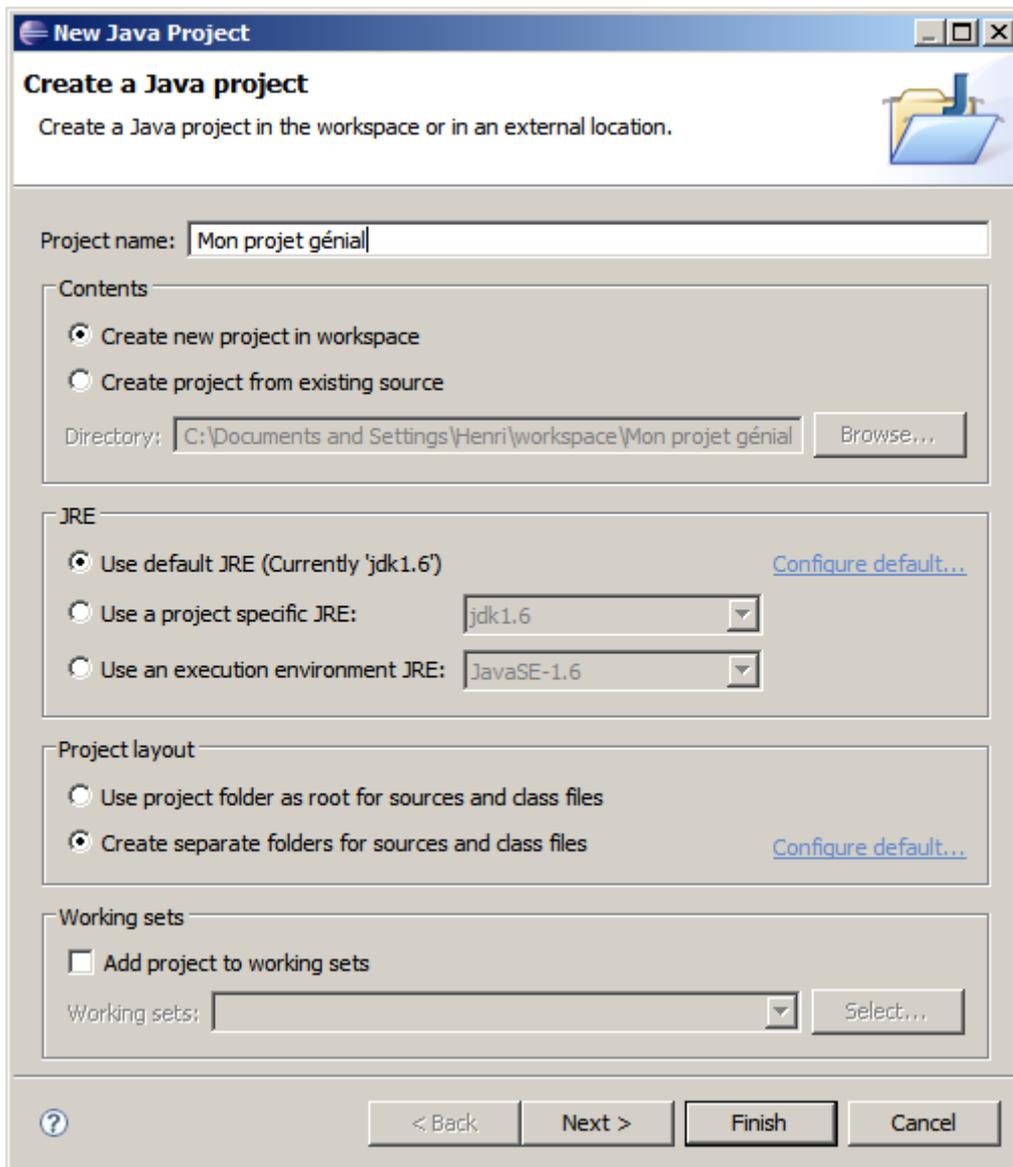
Eclipse est un environnement qui permet une grande variété d'activités de développement (pour vous en convaincre, faites un tour chez [eclipse plugin central](http://eclipse.plugin.central) ou sur le site francophone eclipsetotale.com). En standard, eclipse est prêt pour le développement en Java, encore faut-il veiller à ce que la perspective (c'est-à-dire l'arrangement des vues montrées à l'écran) soit celle qui convient le mieux à Java. Si ce n'est pas le cas, agissez sur la petite icône en haut à droite – cerclée de rouge –, étiquetée Open perspective et choisissez Java.

Fermez les vues Task List et Outline (à droite) ; pour afficher la structure des classes, la vue Package explorer (à gauche) suffit. Vous obtenez un cadre de travail tout à fait commode pour développer en Java :

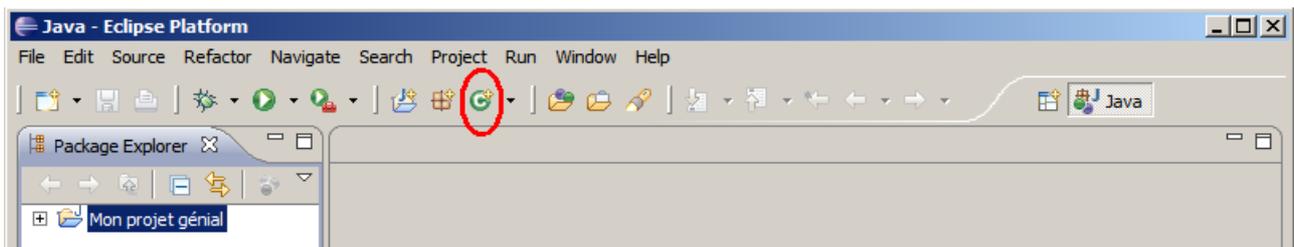


6. Développer un programme Java

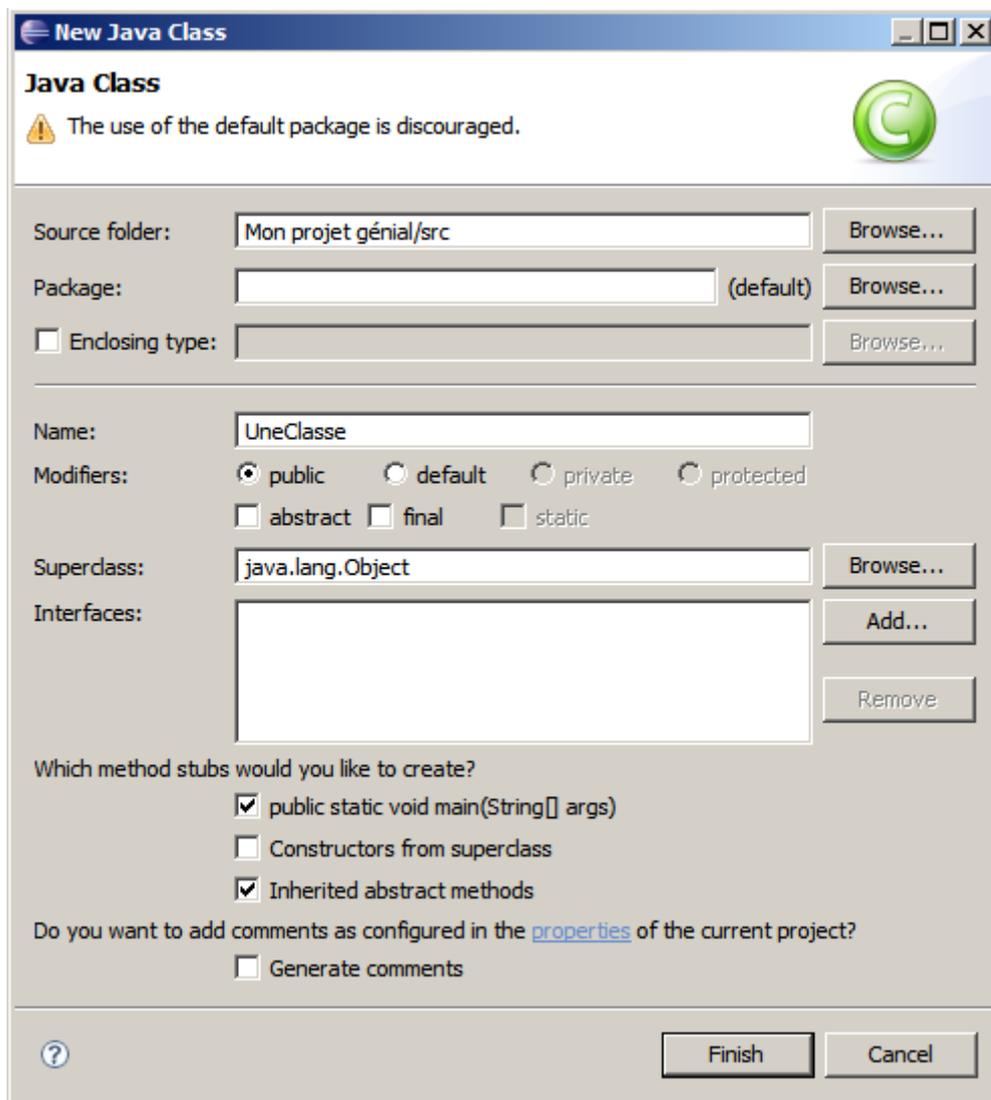
Pour créer un projet, cliquez sur le premier des boutons d'assistants Java (cerclé de rouge sur la figure 5) étiqueté « New Java Project ». Vous obtenez le panneau New Java Project où, au minimum, vous devez donner un nom pour votre projet :



La méthode rapide consiste à donner un nom de projet et cliquer sur le bouton Finish. Notez que les autres « questions » posées dans ce panneau sont intéressantes. La troisième, notamment, permet de conserver séparément les fichiers sources (précieux) et les fichiers classes (qu'en cas de perte on peut toujours refaire).

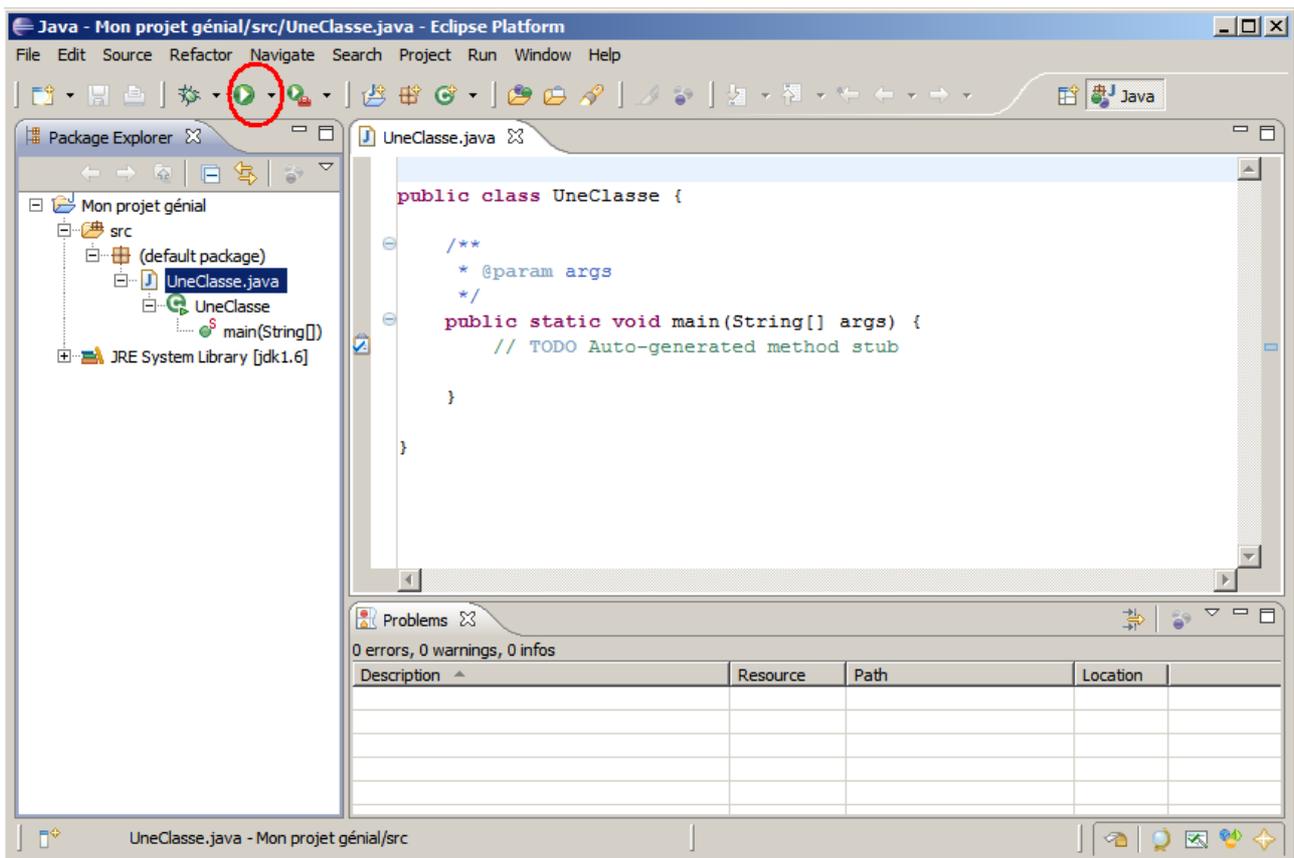


Dans un projet sérieux nous commencerions par créer des packages (deuxième bouton des assistants Java, « New Java Package »). Mais, puisque nous débutons, allons à l'essentiel et ajoutons directement une ou plusieurs classes au projet : c'est le troisième des boutons d'assistants Java, « New Java Class » (cerclé de rouge), qui fait cela. La méthode rapide consiste à donner le nom de la classe et cocher la case étiquetée `public static void main(String[] args)` :



Notez qu'Eclipse critique notre démarche, nous indiquant que l'emploi du package par défaut (sans nom) est découragé.

Eclipse crée alors un fichier source contenant une classe rudimentaire, correcte mais creuse, que vous n'avez plus qu'à compléter pour en faire le programme voulu :

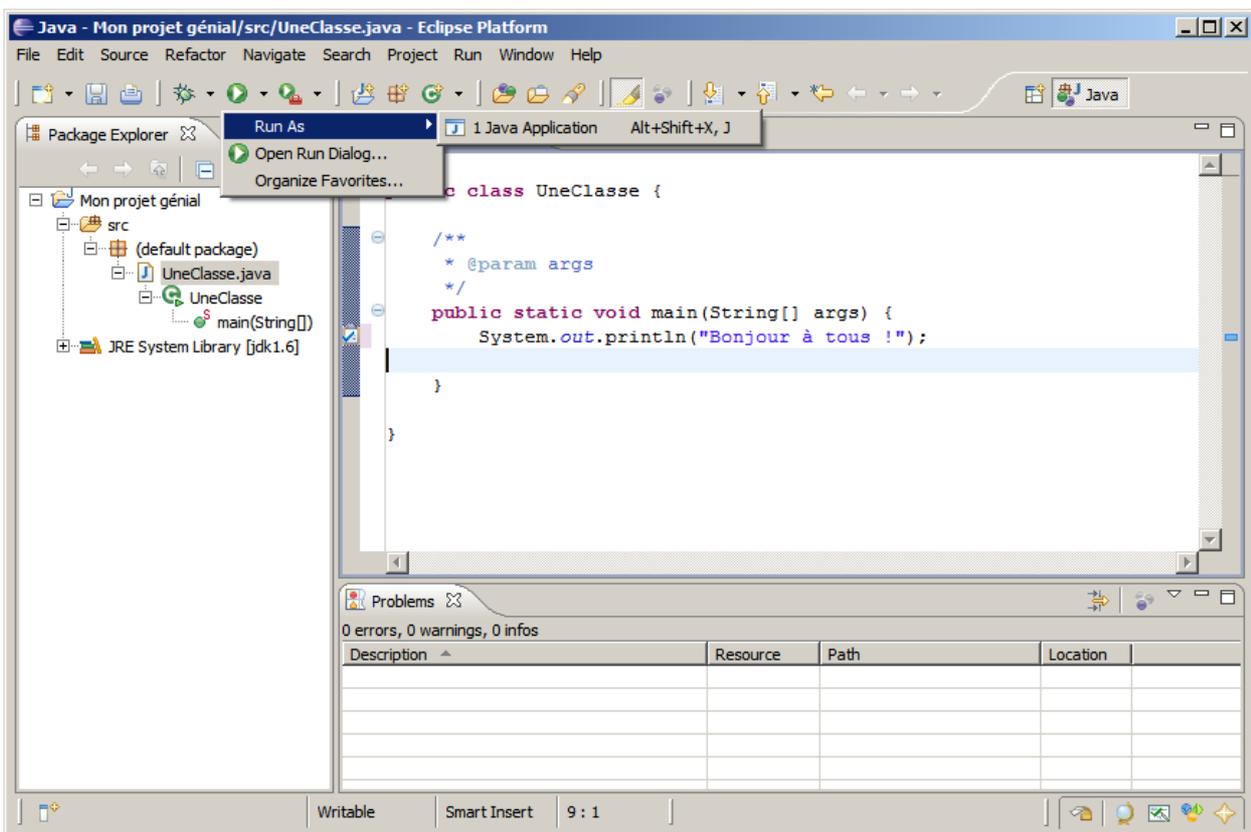


Lorsqu'un commentaire contient l'expression TODO, eclipse affiche une marque bleue dans la marge qui permet de se rendre rapidement à cet endroit. C'est très pratique pour retrouver dans les gros fichiers ces commentaires qui signalent des morceaux en chantier.

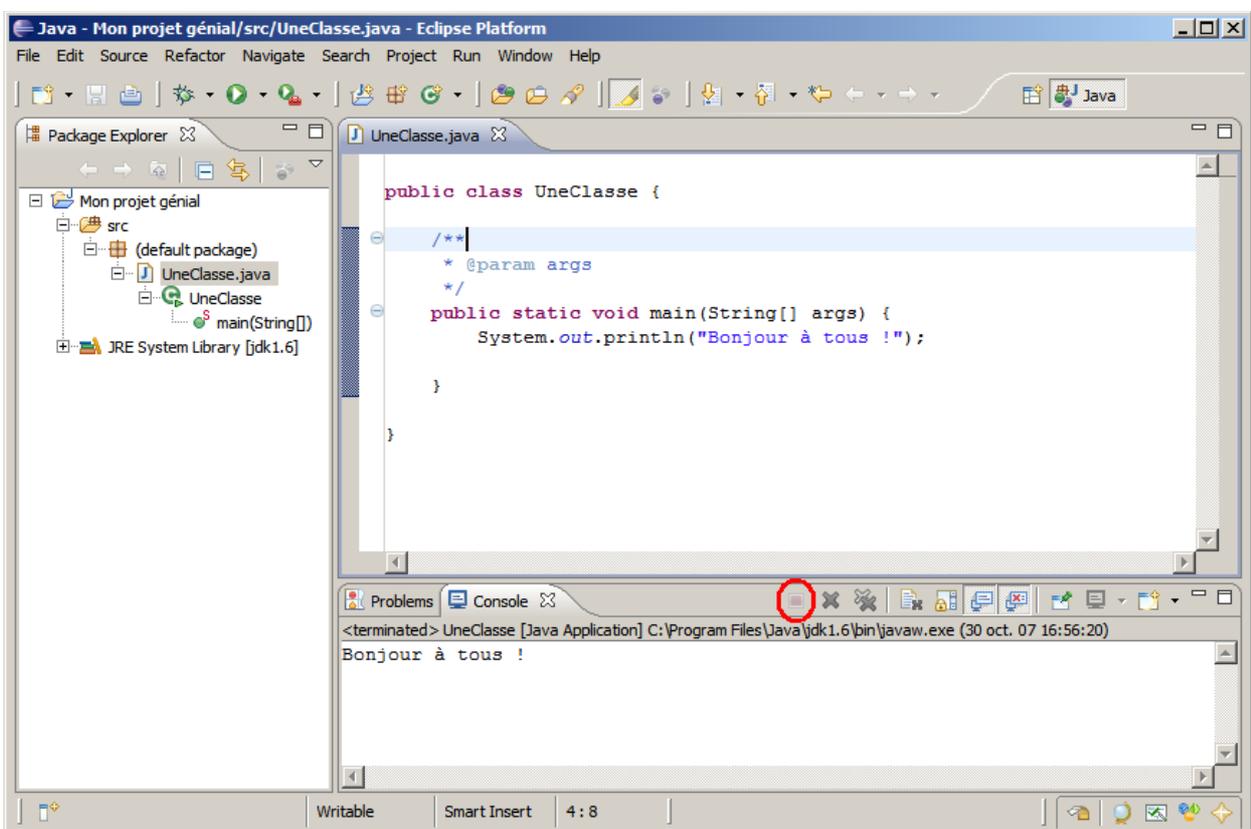
Pour essayer votre programme vous allez taper le classique `System.out.println("Bonjour à tous!");` à l'intérieur de la fonction main. Au fur et à mesure que vous tapez, remarquez comment :

- la vue Package Explorer montre les packages (répertoires) qui composent votre projet, les classes que ces paquetages contiennent, les membres de ces classes, etc. Bien entendu, double cliquer sur une de ces entités vous positionne dessus dans le texte source.
- si vous marquez une pause lorsque vous tapez un point, eclipse vous montre la liste de ce que vous pouvez taper ensuite,
- si vous laissez traîner le curseur sur un identificateur, eclipse affiche la documentation correspondante,
- si vous faites une faute eclipse vous la signale immédiatement et, dans le cas d'erreurs sémantiques, vous suggère des corrections,
- le simple fait de sauver le programme en provoque la compilation

Pour exécuter le programme assurez-vous que la vue éditeur contient une classe exécutable (c'est-à-dire une classe publique avec une méthode `public static void main(String[] args);`) et alors activez la commande Run as > Java Application du menu attaché au bouton cerclé de rouge :



L'application s'exécute et, si des sorties sont à afficher, une vue Console apparaît au-dessous de la vue éditeur :



Notez que dans la vue Console il y a un bouton – cerclé de rouge – qui permet d’arrêter une application qui bouclerait indéfiniment. Ce bouton est rouge quand l’application est vivante, gris (estompé) lorsque l’application est morte.

7. Emplacement des fichiers sources

Après avoir développé une application dans eclipse, vous souhaitez récupérer vos fichiers sources pour les amener sur un autre système, les compiler dans un autre environnement ou tout simplement les ranger dans vos archives.

Lors de la création du projet si vous avez laissé l’option par défaut « Create new project in workspace » alors les sources, rangés dans des dossiers correspondant aux packages, sont dans le dossier workspace, lui-même placé à l’endroit que vous avez indiqué au lancement d’eclipse.

7. Enregistrer les fichiers sources dans un autre dossier

Deux cas possibles : ces fichiers forment déjà un projet eclipse (par exemple créé sur un autre système), ou bien il ne s’agit que d’un ensemble de fichiers sources en vrac.

1. Vous avez déjà un projet eclipse :

Copiez le dossier du projet eclipse où vous voulez (par exemple dans le dossier workspace, mais ce n’est pas une obligation), puis faites la commande File > Import... Ensuite, choisissez General puis Existing Projects into Workspace. Le projet que vous venez d’importer apparaît dans la fenêtre Package explorer, c’est terminé.

2. Vous n’avez qu’un ensemble de fichiers sources Java :

Prenez un projet qui existe déjà, ou bien créez un nouveau projet. Ensuite :

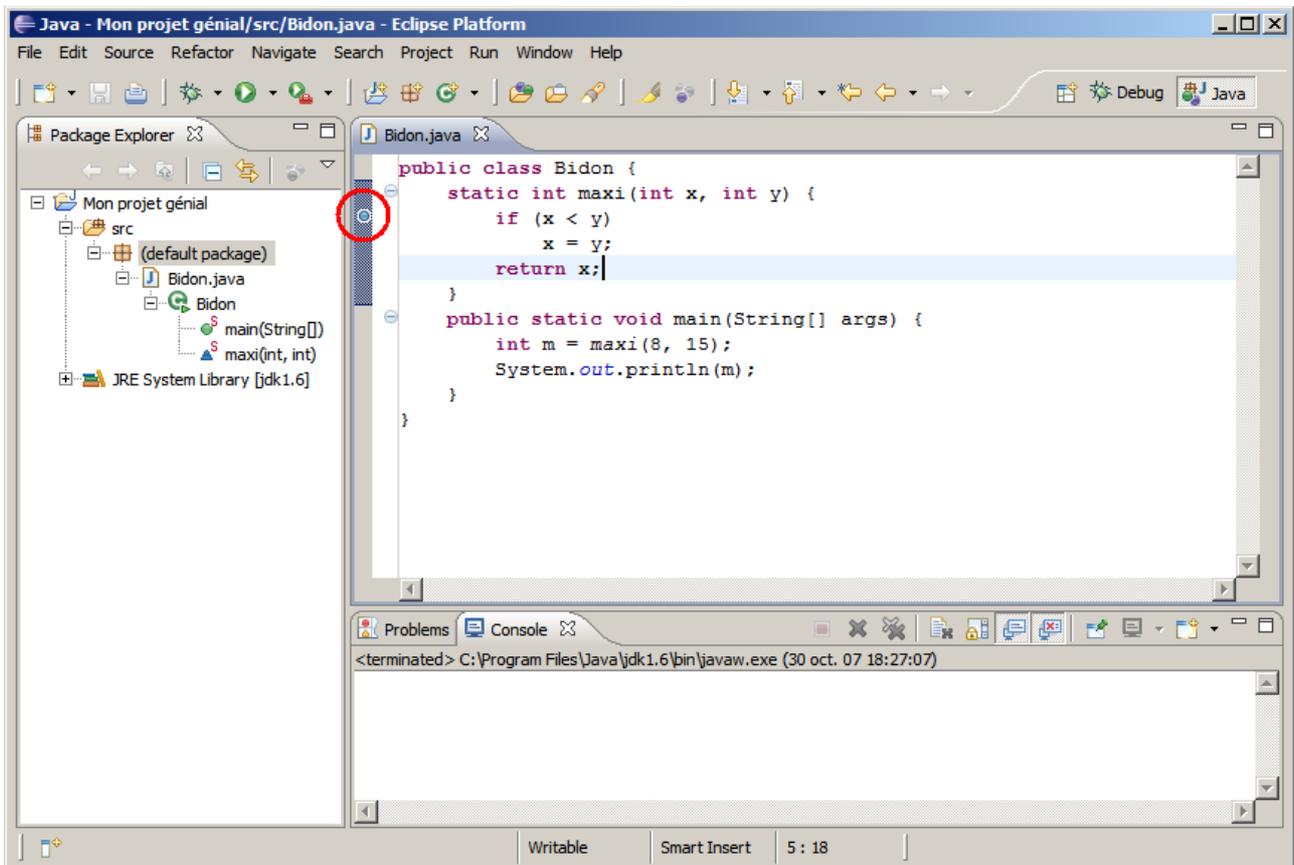
- soit, à l’aide de la commande File > Import... > General > File system, vous naviguez à la recherche du(des) fichier(s) en question et vous les importez dans ce projet,
- soit, plus simplement :
 - ➔ vous copiez les fichiers dans le dossier où sont les sources d’un des projets connus dans eclipse
 - ➔ vous sélectionnez ce projet dans la vue Package Explorer
 - ➔ vous exécutez la commande File > Refresh

9. Déboguer les programmes

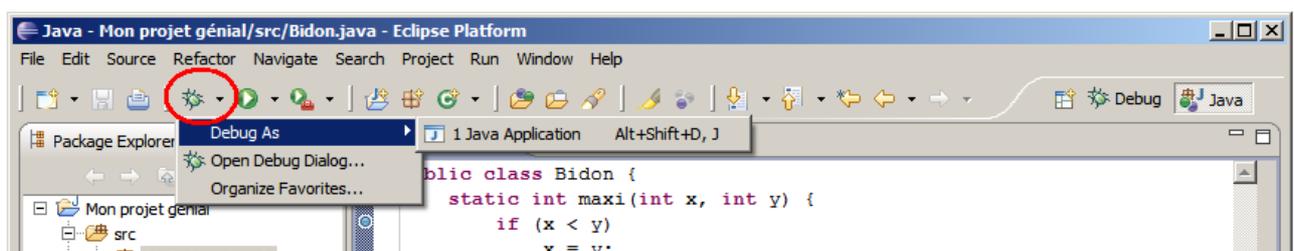
Un programme « bogué » est un programme qui ne donne pas les résultats qu’il devrait. « Déboguer » un programme c’est chercher les erreurs de programmation à l’origine de tels dysfonctionnements. Pour aider le programmeur dans cette recherche, eclipse offre un mode debug permettant, entre autres choses :

- la pose de marques, appelées points d’arrêt, sur des lignes du programme source, de telle manière que l’exécution s’arrêtera lorsque ces instructions seront atteintes,
- lors de tels arrêts, l’examen des valeurs qu’ont alors les variables locales et les membres des objets,
- à partir de là, l’exécution du programme pas à pas (c’est-à-dire ligne à ligne)

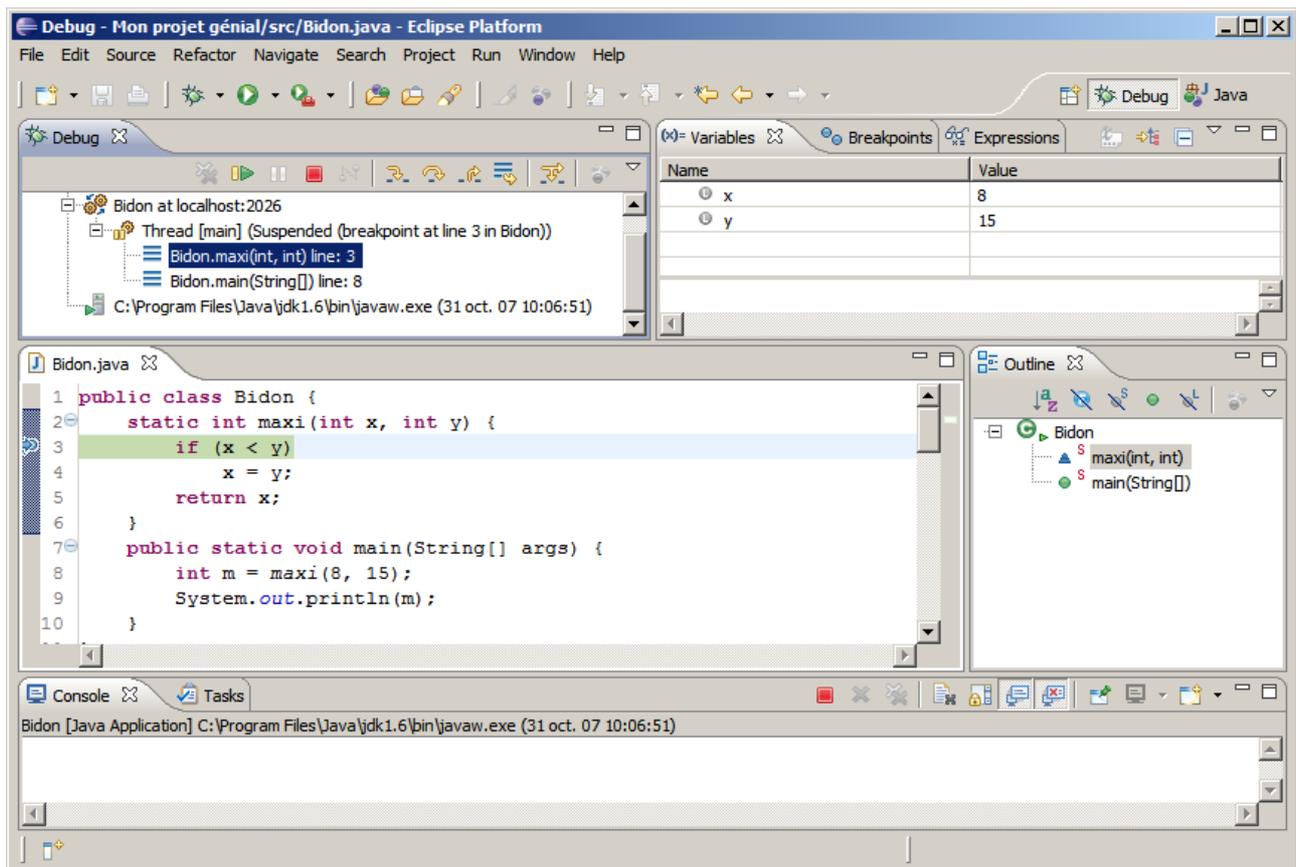
Pour déboguer simplement un programme il suffit de poser un point d'arrêt au début de l'endroit qu'on souhaite examiner en détail. Pour cela il faut double-cliquer dans la marge, à gauche de la ligne en question, ce qui fait apparaître un disque bleu (cerclé de rouge) qui représente le point d'arrêt.



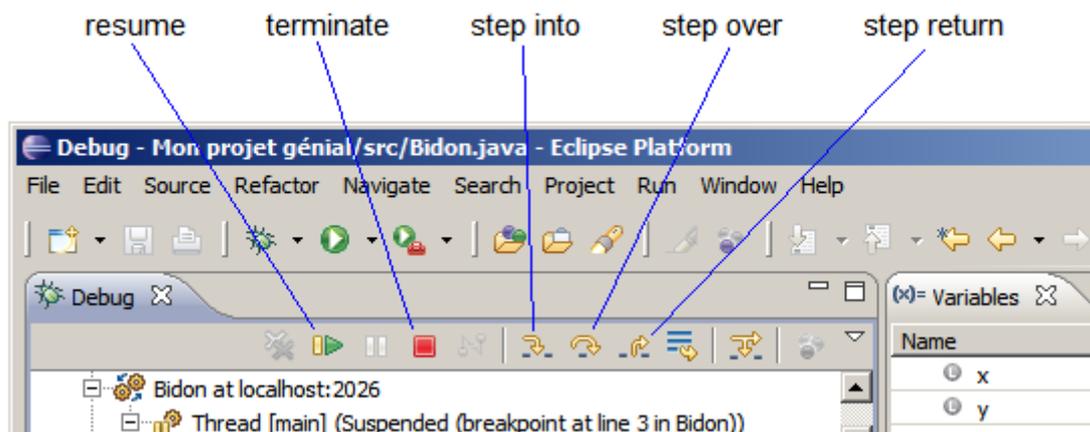
Il faut ensuite lancer le débogage, à l'aide du bouton à gauche de celui qui lance l'exécution, représentant une punaise (bug) :



L'exécution est alors lancée et se déroule normalement jusqu'à atteindre le point d'arrêt. Eclipse demande alors la permission de changer de perspective (ensemble et disposition des vues montrées) et adopte l'apparence :



La vue Debug, en haut à gauche de la fenêtre, montre la pile d'exécution, c'est-à-dire, pour chaque thread, l'empilement des méthodes qui se sont mutuellement appelées (méthodes commencées et non terminées). Dans la figure 14, par exemple, on attire notre attention sur la méthode Bidon.main, plus précisément la ligne 8 du fichier source, où a été appelée la méthode Bidon.maxi, dans laquelle l'exécution est arrêtée, à la ligne 3.



En haut de cette vue se trouvent des boutons très utiles. Parmi les principaux :

- step over : faire avancer l'exécution d'une ligne. Si cette dernière contient un appel de méthode, ne pas détailler l'activation de celle-ci, c'est-à-dire considérer l'appel comme une instruction indivisible,
- step into : avancer l'exécution d'une ligne. Si un appel de méthode est concerné, détailler

son activation, c'est-à-dire aller dans la méthode et s'arrêter sur sa première ligne,

- `step return` : relancer l'exécution normale, jusqu'à la fin de la méthode dans laquelle on est arrêté et le retour à la méthode qui a appelé celle-ci,
- `resume` : relancer l'exécution normale, jusqu'à la fin du programme ou le prochain point d'arrêt,
- `terminate` : terminer l'exécution.

En haut et à droite de la fenêtre principale se trouvent les vues Variables et Expressions. La première affiche les valeurs courantes des variables locales de la méthode en cours, la deuxième affiche les valeurs courantes des expressions sélectionnées avec la commande Watch (cliquer avec le bouton droit sur l'expression à surveiller).

Le débogueur d'eclipse possède bien d'autres commandes très puissantes, comme les points d'arrêt conditionnels et la possibilité de modifier les valeurs des variables du programme. Prenez un peu de temps pour les explorer, c'est payant.



Pour en savoir plus, cliquer ici