

EDI Eclipse - Prise en main rapide

EDI, Environnement de Développement Intégré

Anglais : IDE, Integrated Development Environment

Un EDI est un logiciel utilisé dans le cadre des développements informatiques, regroupant :

- Un éditeur de texte pour la saisie des programmes sources, disposant d'une coloration syntaxique et de la 'complétion' de code contextuelle ;
- Un compilateur qui traduit le code source en code exécutable par la JVM (*JVM, Java Virtual Machine*)
- Un débogueur (*anglais : debugger*) : outil d'aide au diagnostic en cas d'erreur d'exécution d'un programme
- Un système d'aide
- Un certain nombre d'outils dédié à la productivité du développement informatique (*assistants*)

Les EDI gèrent les développements d'applications sous forme de **projets** : un projet comporte plusieurs fichiers, parmi lesquels les fichiers sources, des fichiers de configuration utiles à l'EDI (*paramètres des options de configuration de l'interface graphique de l'EDI, etc.*)

EDI Eclipse

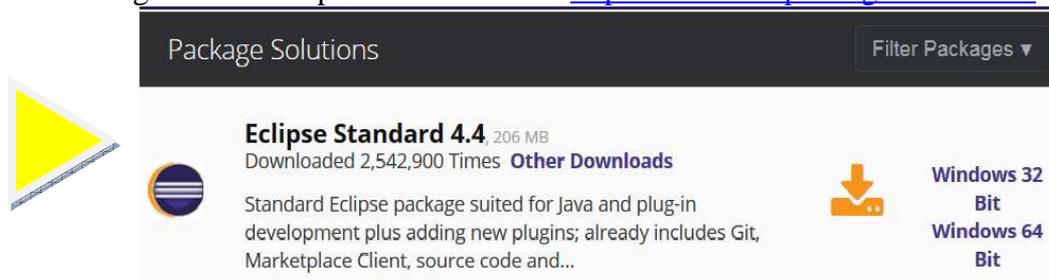
C'est un outil Open Source et gratuit.

Il est développé en Java, ce qui le rend portable sur les plateformes disposant de l'environnement d'exécution Java (*la JVM, Java Virtual Machine*).

2- Téléchargement et installation – version actuelle 4.4 (sept 2014)

Site officiel : <https://www.eclipse.org>

Le lien de téléchargement est disponible sur le site : <https://www.eclipse.org/downloads/>



The screenshot shows the Eclipse Package Solutions interface. At the top, there is a 'Package Solutions' header with a 'Filter Packages' dropdown. Below this, a yellow triangle icon is visible on the left. The main content area features a card for 'Eclipse Standard 4.4' with a size of 206 MB and a download count of 2,542,900. The card includes a description: 'Standard Eclipse package suited for Java and plug-in development plus adding new plugins; already includes Git, Marketplace Client, source code and...'. To the right of the card, there are download icons for 'Windows 32 Bit' and 'Windows 64 Bit'.

Choisir le site de téléchargement :

Download eclipse-standard-luna-R-win32-x86_64.zip from:



[France] Ialto (http)

Checksums: [MD5] [SHA1] [SHA-512]

Enregistrer le fichier ZIP (archive compressée) : taille. env. 200 Mo. puis extraire les fichiers de l'archive

EDI Eclipse - Prise en main rapide

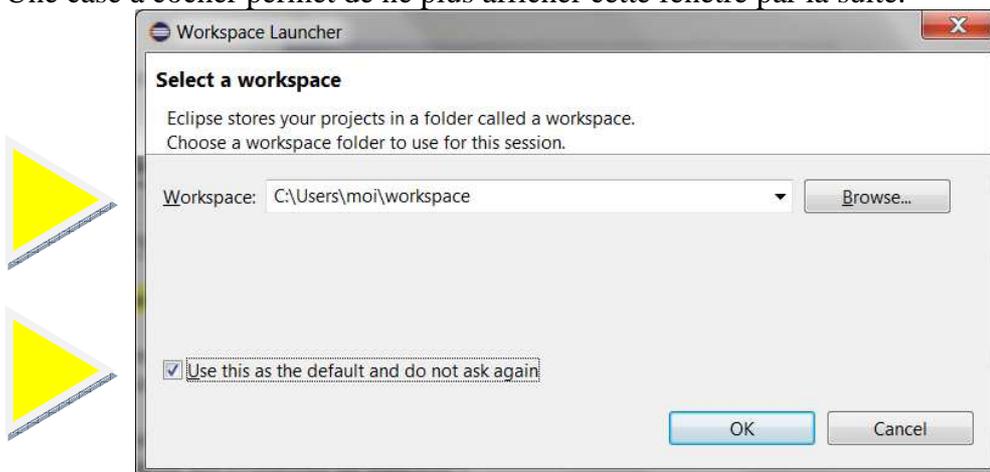
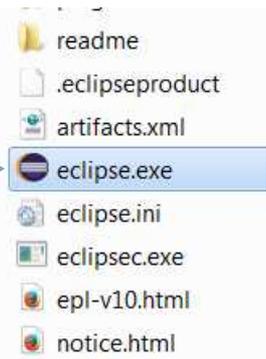
3- Lancement et création d'un premier projet

Démarrage

Dans le répertoire extrait, double-clic sur Eclipse :

Le chargement est relativement long : beaucoup de modules sont chargés au départ afin d'améliorer l'utilisation du produit en développement.

Au premier lancement, une fenêtre propose de définir le répertoire de travail d'Eclipse. Une case à cocher permet de ne plus afficher cette fenêtre par la suite.



La fenêtre d'accueil propose divers choix, parmi lesquels l'espace de travail (workbench) :



La fenêtre de l'application peut comporter des différences dans la présentation :



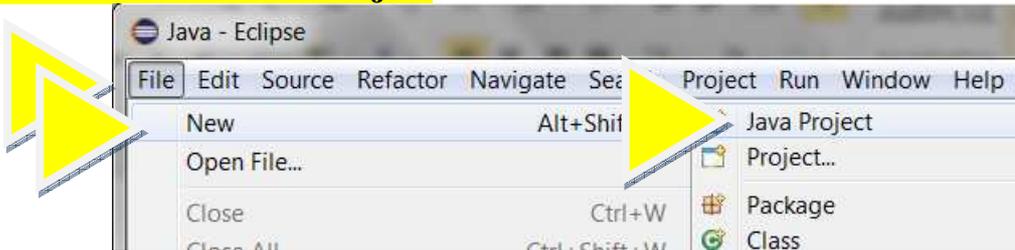
Par défaut, la fenêtre est prête pour les développements Java : le terme « perspective » définit un ensemble d'outils (*ensemble de « vues », fenêtres présentant des informations particulières en fonction du contexte*):

- perspective « Java » pour saisir ou modifier du code
- perspective « Debug » pour aider à la découverte des anomalies de fonctionnement d'un programme

EDI Eclipse - Prise en main rapide

Créer un nouveau projet

>> menu **File > New > Java Project**



Une nouvelle fenêtre permet de définir

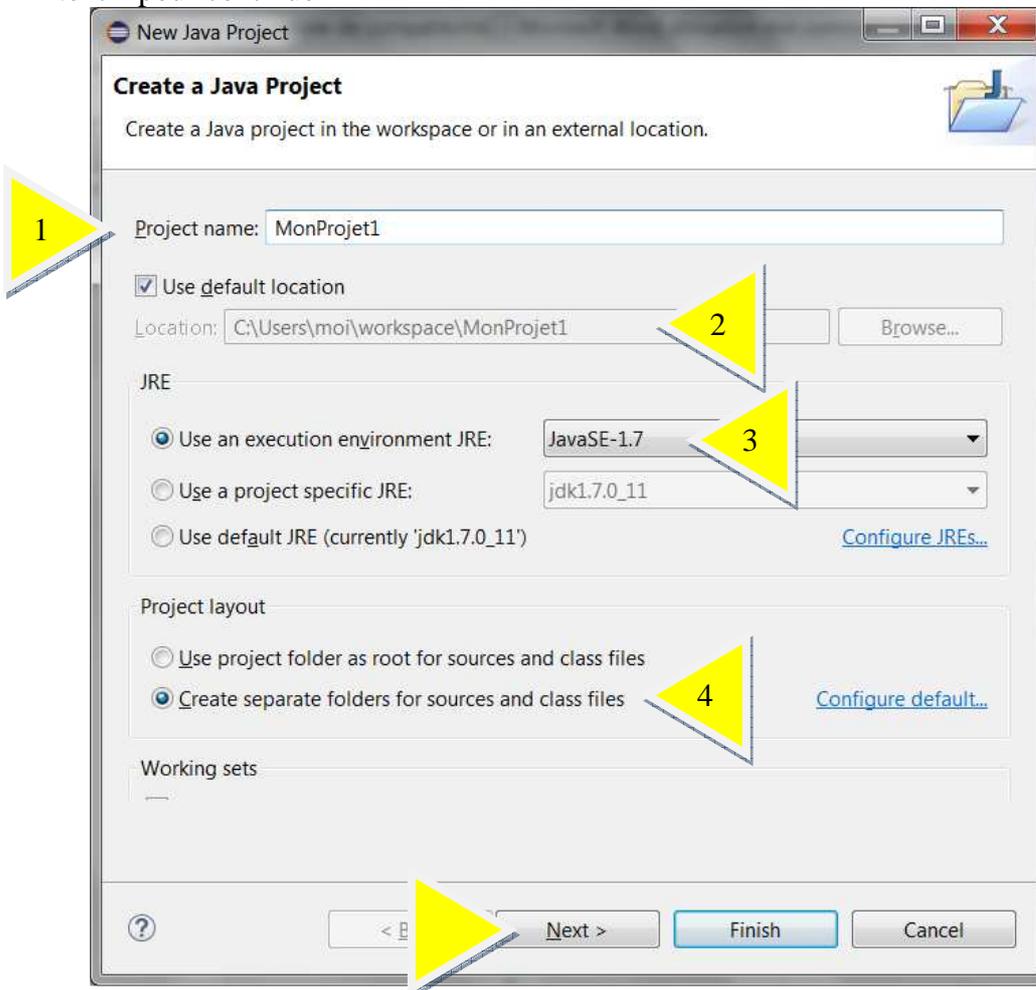
1. le nom du projet

2. le répertoire de stockage des fichiers du projet (codes sources Java, entre autres)

3. la version de Java utilisée (Eclipse a dû le découvrir par défaut)

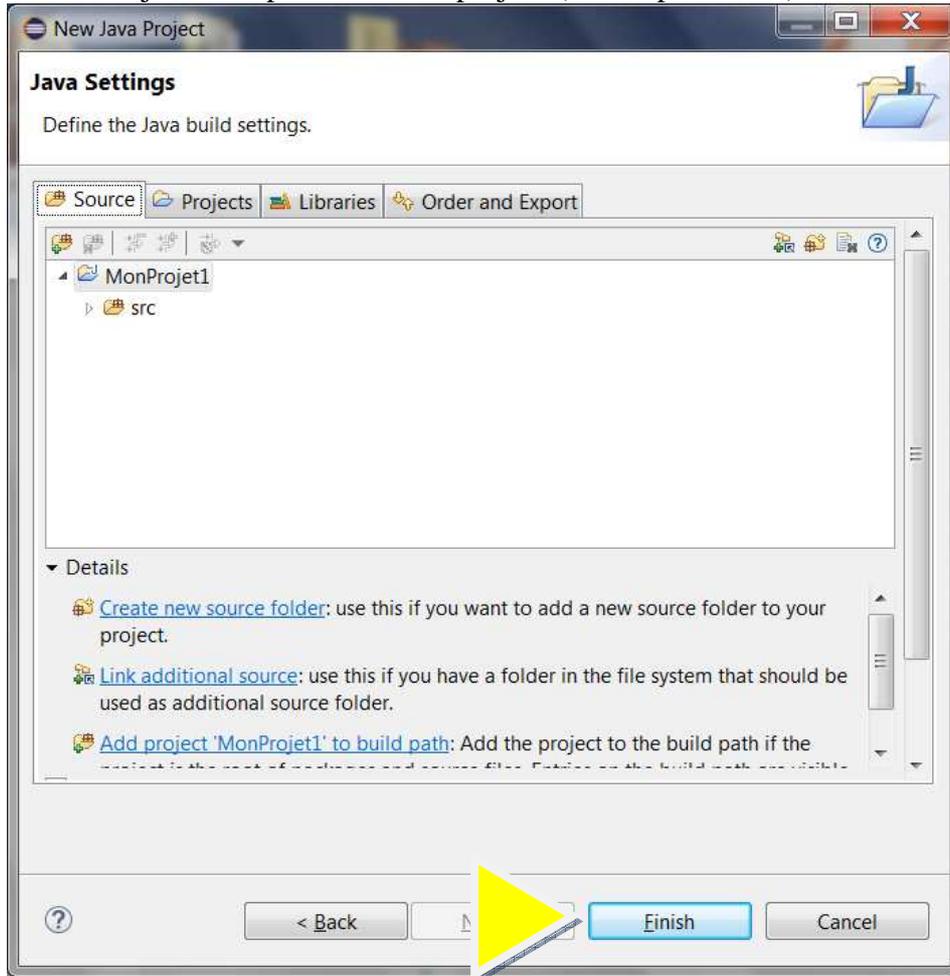
4. le fait de créer un répertoire différent pour les codes sources et les codes compilés

Cliquer sur « Next » pour continuer

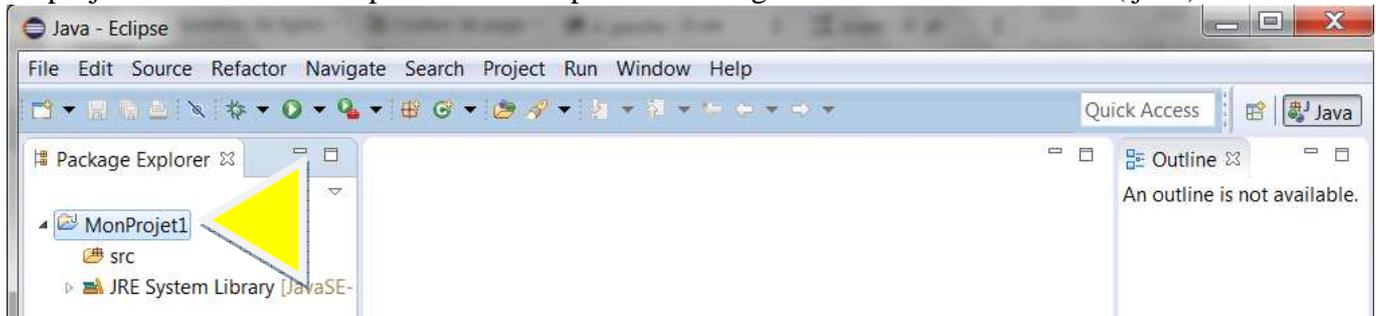


EDI Eclipse - Prise en main rapide

La possibilité est offerte d'ajuster les paramètres du projet : (laisser par défaut) : Finish

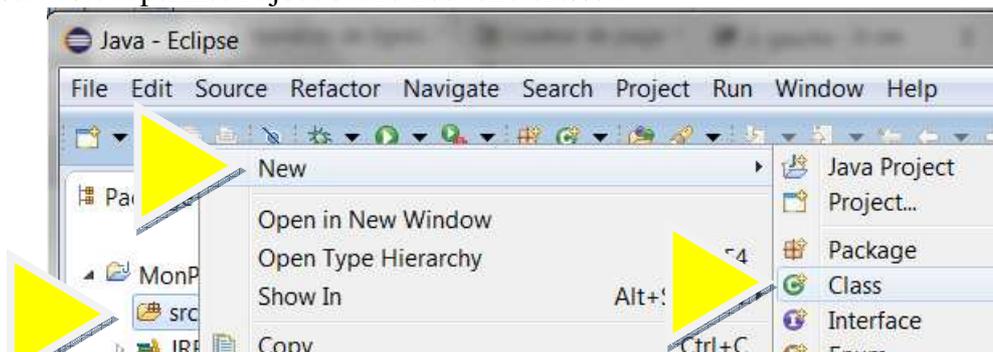


Le projet est créé avec un répertoire « src » pour le stockage des fichiers sources Java (.java) :



Ajouter une classe au projet

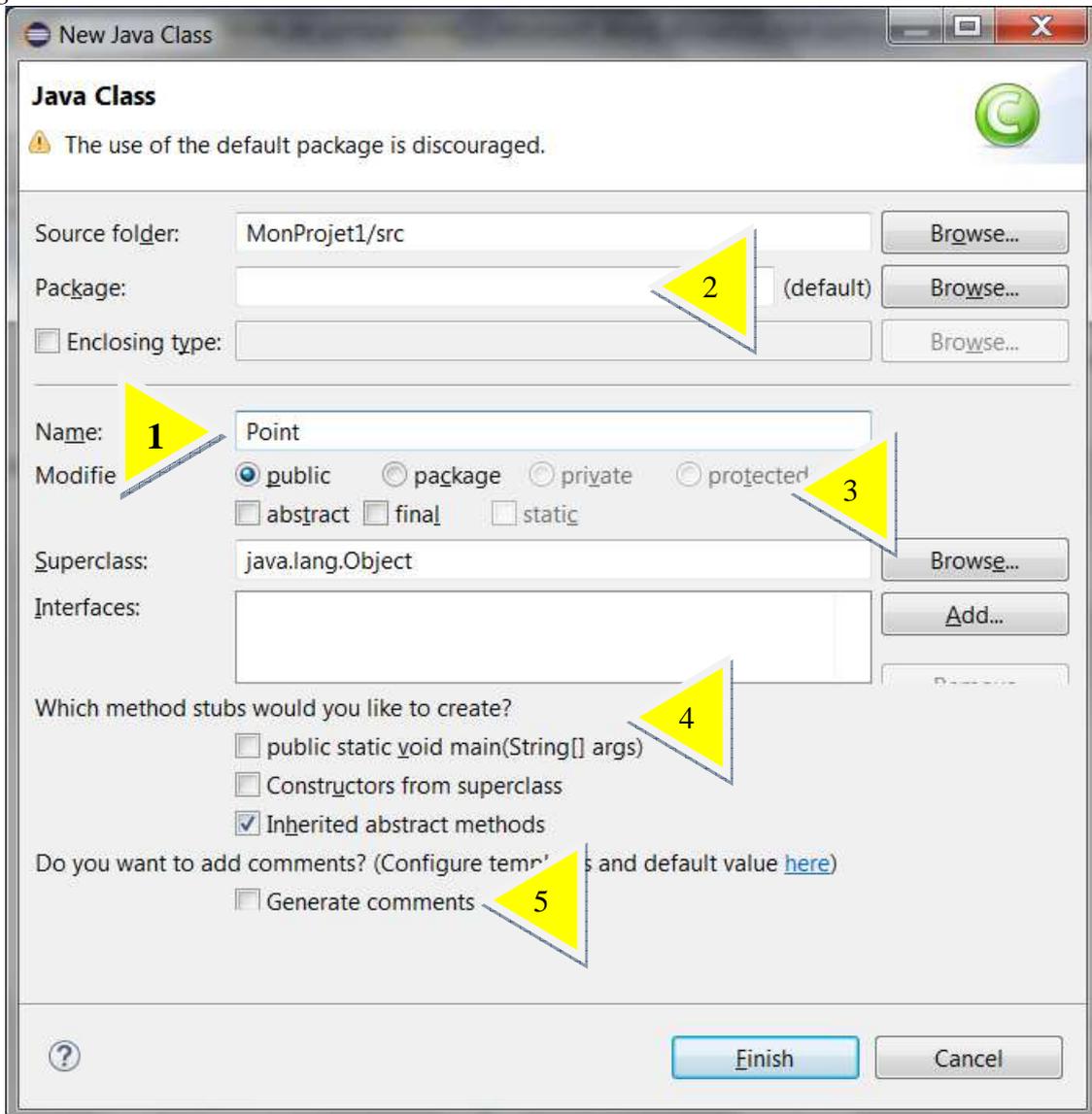
Un clic-droit sur « src » permet l'ajout d'une nouvelle classe :



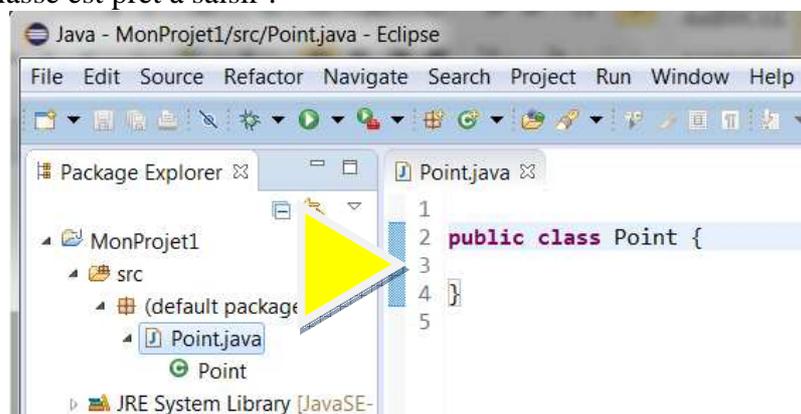
Une nouvelle fenêtre permet de définir

EDI Eclipse - Prise en main rapide

1. le nom de la classe
2. le nom du package (un message « l'utilisation du package par défaut est à éviter » : c'est pourtant ce que nous ferons dans un premier temps...
3. des modificateurs de classe
4. des squelettes de méthodes ou autre pour préparer la classe
5. la génération des commentaires de documentation



Le code source de la classe est prêt à saisir :

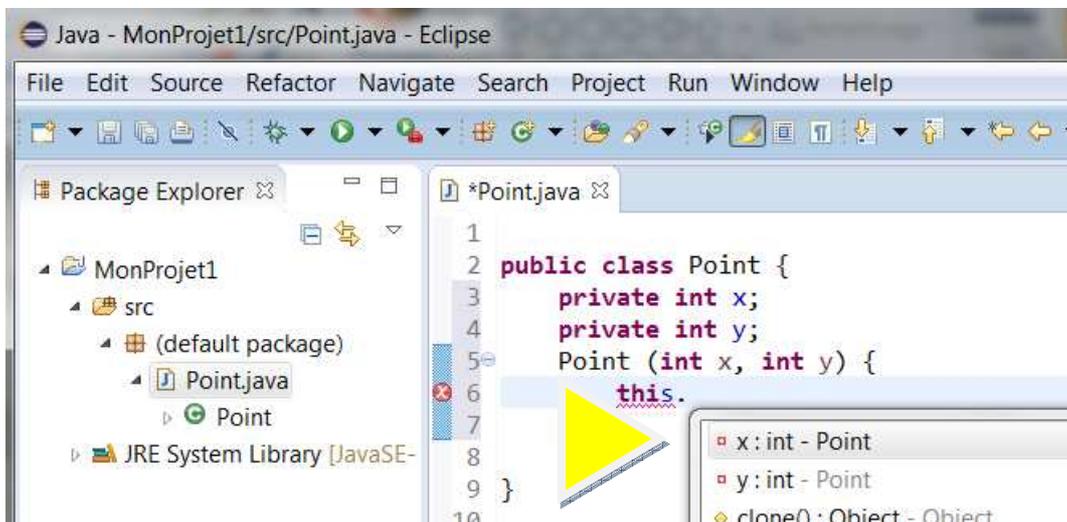


EDI Eclipse - Prise en main rapide

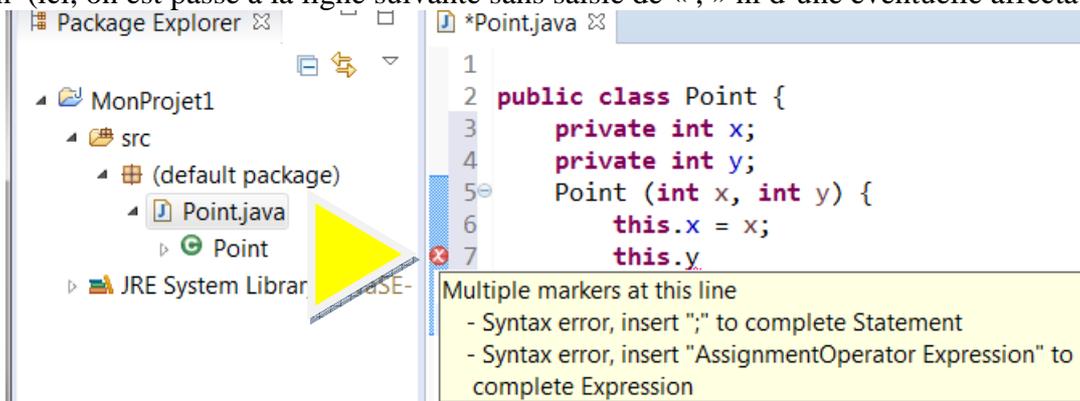
Modifier le code source – Complétion de code et anticipation des erreurs

Au fur et à mesure de la saisie, l'éditeur propose des options contextuelles valides pour simplifier la saisie : c'est ce qu'on appelle la « complétion de code » : ici, après « this. », on peut saisir un de ses membres (*ou un des membres hérités de la classe Object*).

La combinaison de touches CTRL+espace permet à tout moment d'afficher la liste des propositions)



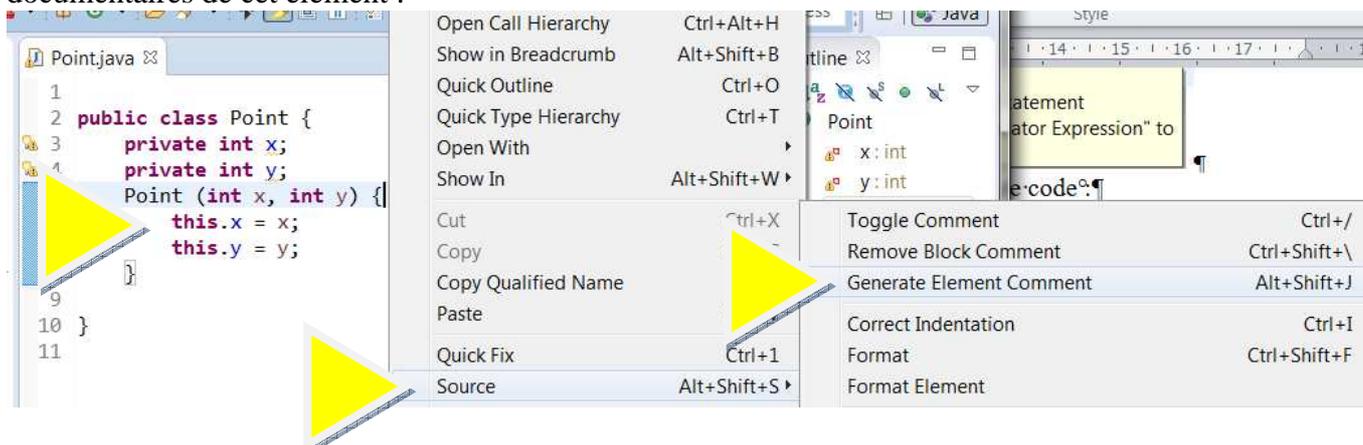
En cas d'erreur supposée de syntaxe, un indicateur d'erreur dans la marge permet d'aider au diagnostic et à la correction (ici, on est passé à la ligne suivante sans saisie de « ; » ni d'une éventuelle affectation) :



A tout moment, des aides contextuelles permettent également la génération de portions de code :

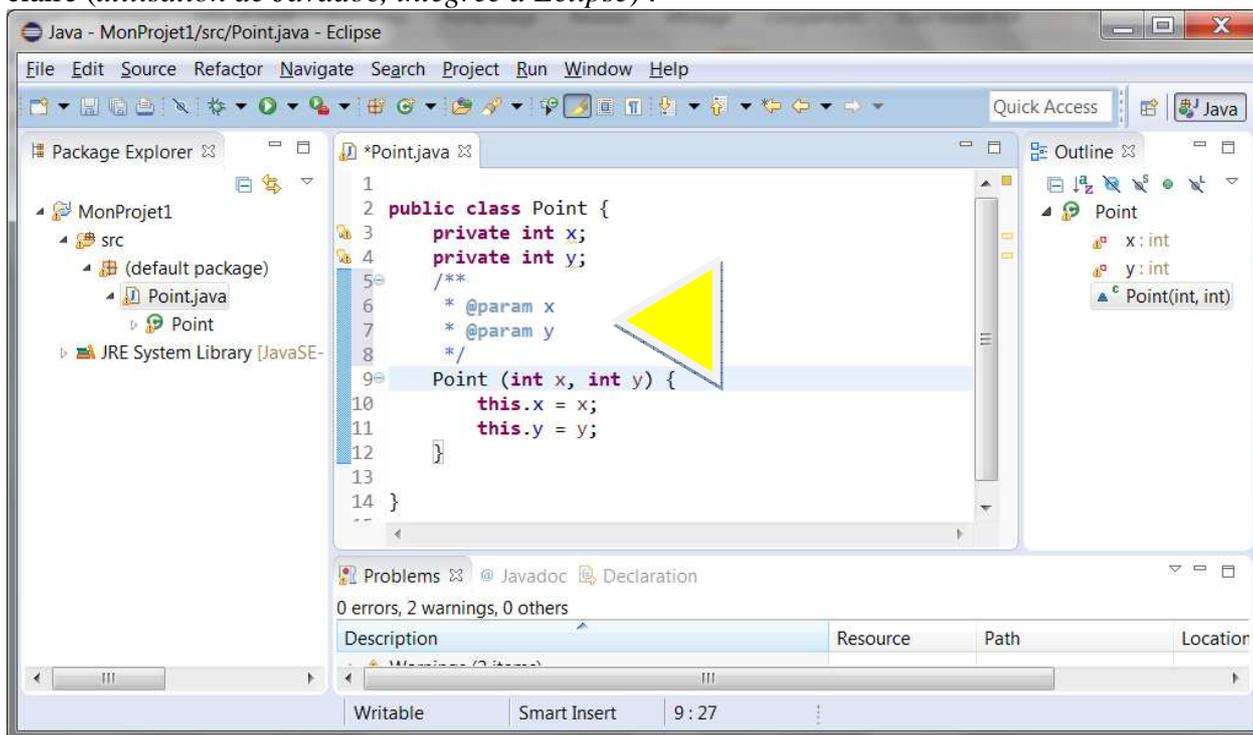
Génération de commentaires Javadoc :

par exemple ici, un clic-droit dans le bloc du constructeur donne accès à la génération des commentaires documentaires de cet élément :



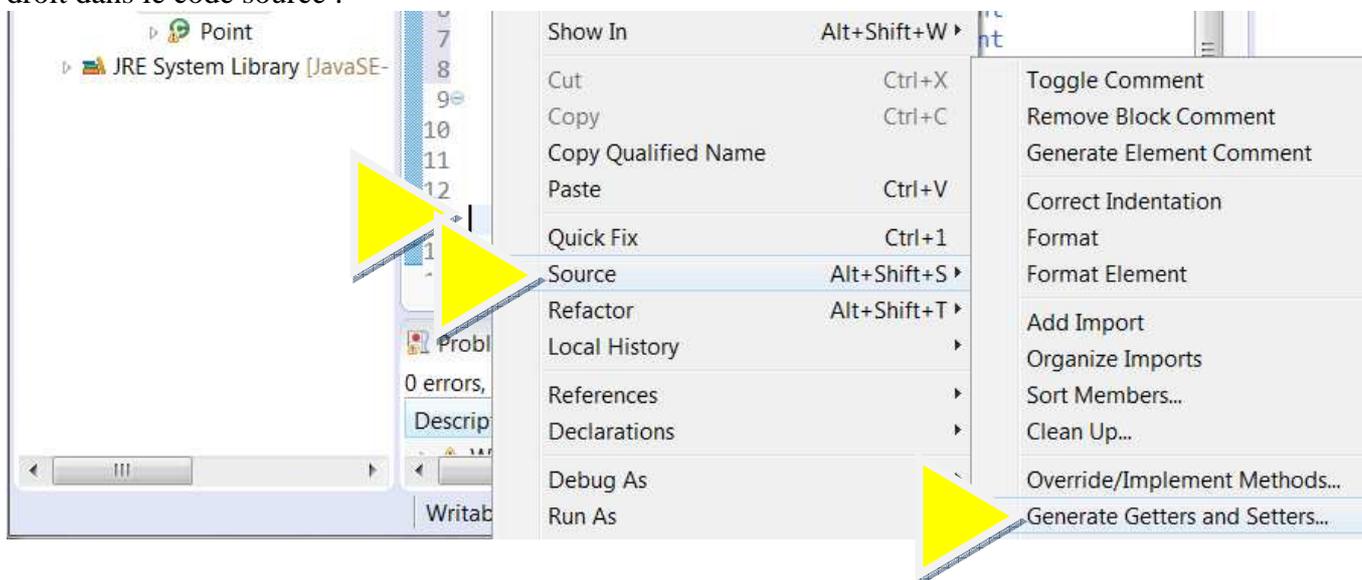
EDI Eclipse - Prise en main rapide

ce qui génère les commentaires suivants, qu'il est possible d'enrichir afin d'obtenir une documentation claire (*utilisation de Javadoc, intégrée à Eclipse*) :



Génération de code Java

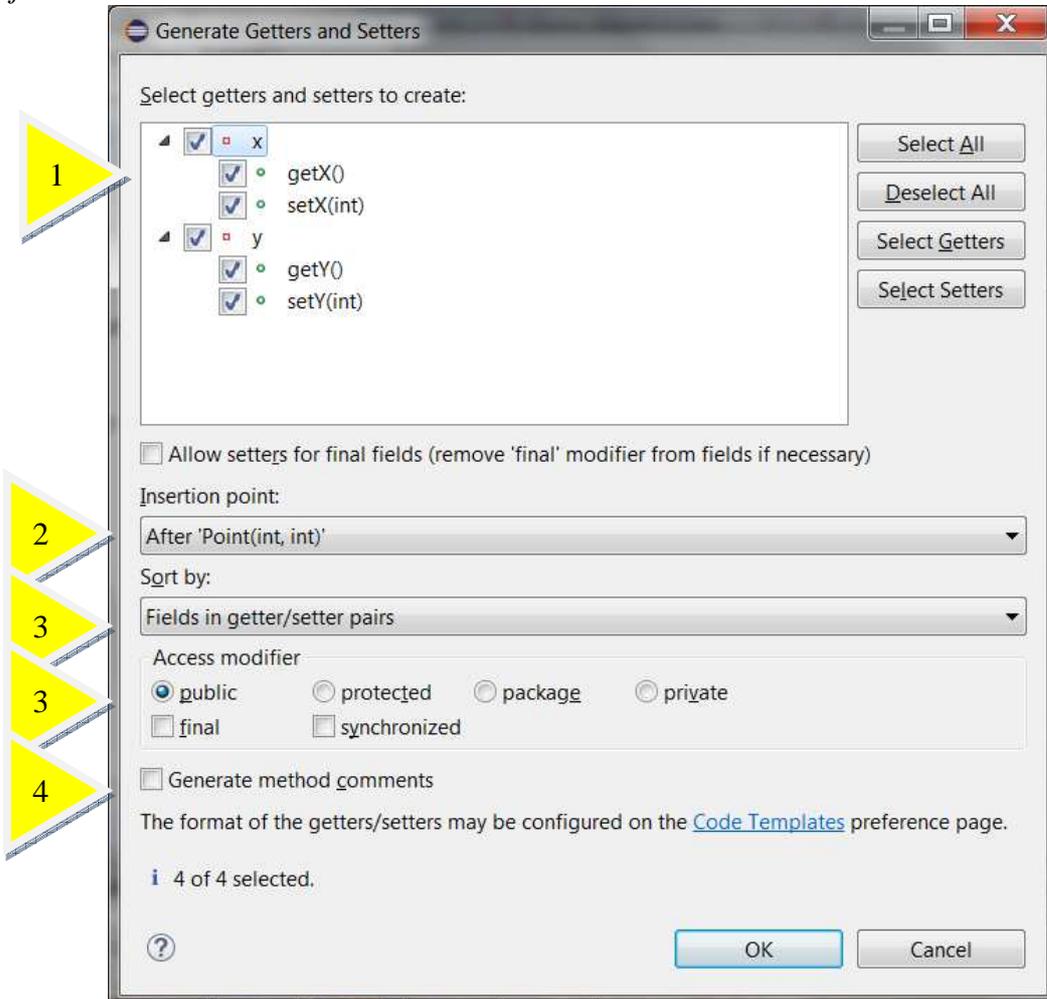
Pour les membres attributs, il est ainsi possible de générer automatiquement les accesseurs à partir d'un clic-droit dans le code source :



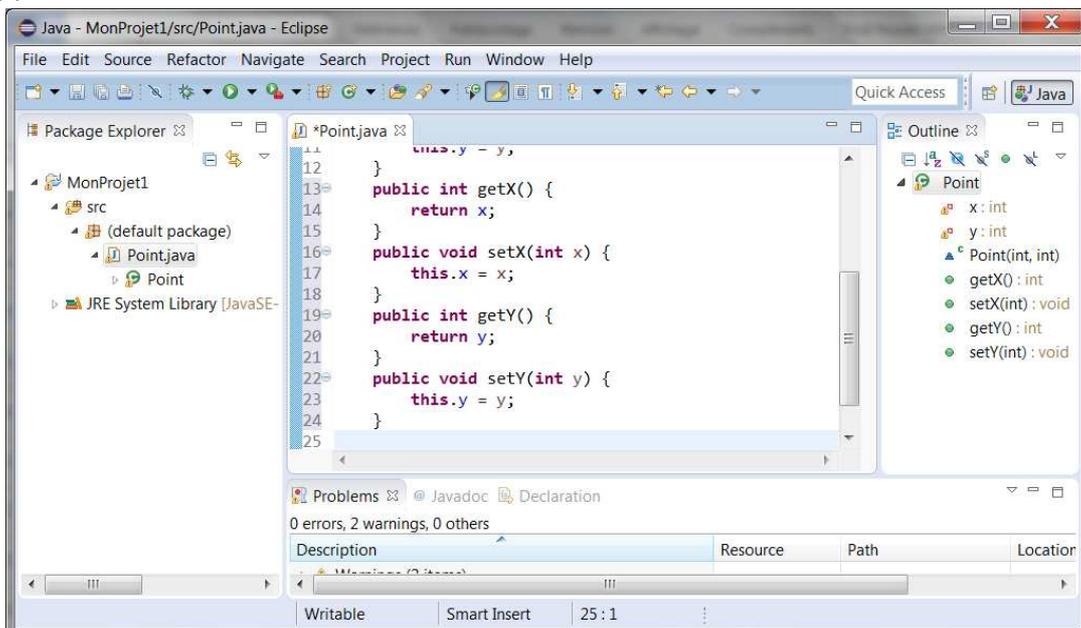
EDI Eclipse - Prise en main rapide

Il est possible de préciser :

1. les accesseurs à insérer
2. la position d'insertion dans le code et l'ordre d'apparition des accesseurs
3. le modificateur d'accès des accesseurs
4. l'ajout des commentaires associés



et le résultat !



EDI Eclipse - Prise en main rapide

Compléter le code avec les méthodes « métier » (spécifiques)

La classe sera complétée

- avec des méthodes d'instance

```
26 public double getDistance(Point p) {
27     return (Math.sqrt(Math.pow(x-p.getX(),2)
28         + Math.pow(y-p.y, 2)));
29 }
30
31 @Override
32 public String toString() {
33     return "Point [x=" + x + ", y=" + y + "]";
34 }
```

- et une méthode « main » afin de pouvoir la tester

```
36 public static void main(String [] args) {
37     // déclarations
38     Point p1, p2;
39     double distance;
40     // initialisation
41     p1 = new Point(0,0);
42     p2 = new Point(1,1);
43     // traitement
44     distance = p1.getDistance(p2);
45     // résultat
46     System.out.println("la distance entre "+p1
47         +" et "+p2+" est "+distance);
48 }
```

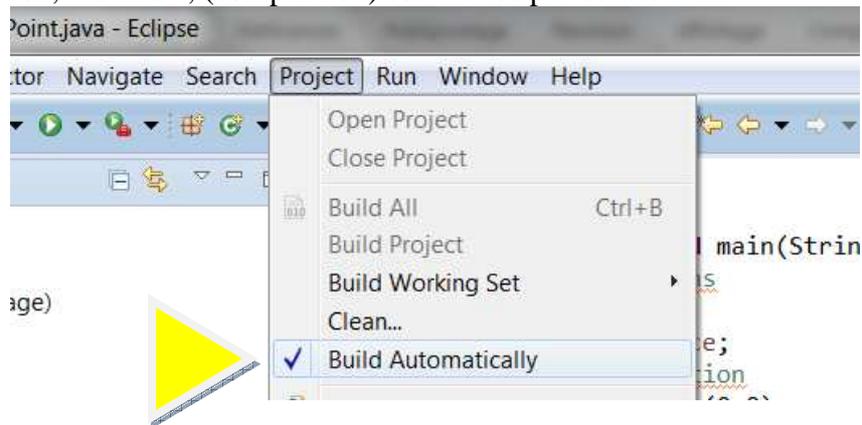
Une représentation hiérarchique donne des indications sur le type de chaque membre (static, constructeur, attribut ou méthode) et donne un accès immédiat (double-clic) au code source de chacun d'eux :

```
36 public static void main(String [] args) {
37     // déclarations
38     Point p1, p2;
39     double distance;
40     // initialisation
41     p1 = new Point(0,0);
42     p2 = new Point(1,1);
43     // traitement
44     distance = p1.getDistance(p2);
45     // résultat
46     System.out.println("la
47         +" et "+p2+
48     }
```

EDI Eclipse - Prise en main rapide

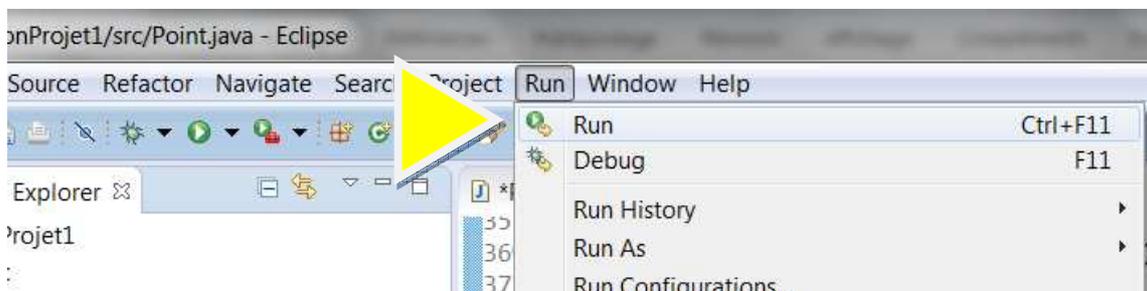
Compiler la classe – construire le projet

L'option de construction, « Build », (compilation) est activée par défaut :

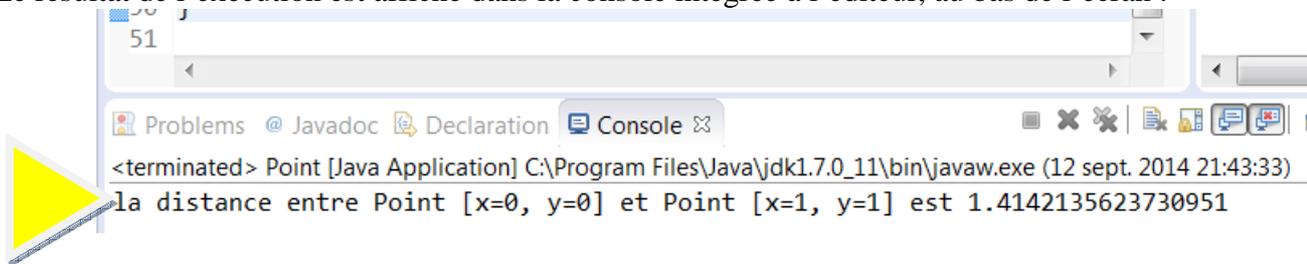


Lancer l'exécution

L'exécution peut être lancée : **>> menu Run > Run (ou CTRL-F11)**



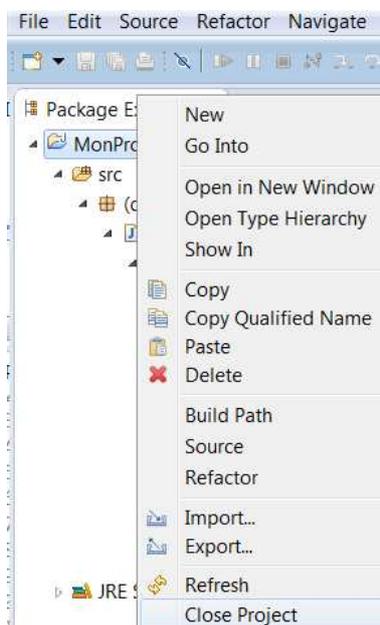
Le résultat de l'exécution est affiché dans la console intégrée à l'éditeur, au bas de l'écran :



Fermer le projet

Clic-droit sur le projet > Close project.

Le projet est fermé, on peut alors passer à un autre projet !



4 - Débogage d'applications

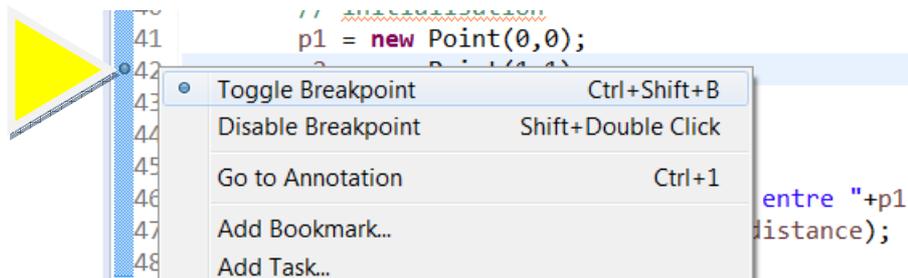
Le débogage est l'activité consistant à découvrir les anomalies d'exécution d'un programme qui ne donne pas le résultat attendu lors des tests avec un jeu d'essais.

Les EDI intègrent généralement des outils facilitant ces opérations :

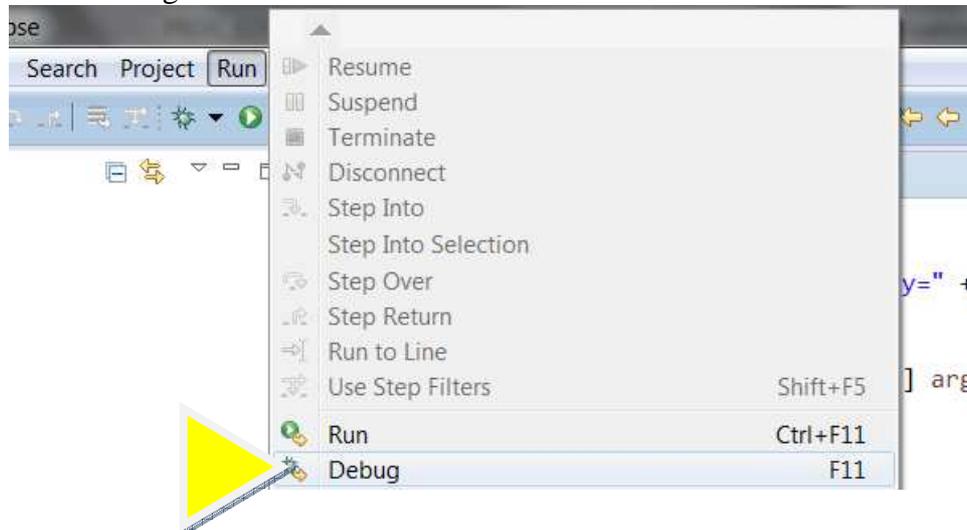
- points d'arrêt : le point d'arrêt permet d'interrompre l'exécution d'un programme, de vérifier certaines valeurs, puis de relancer l'exécution
- consultation des valeurs intermédiaires des variables :
- etc.

Poser des points d'arrêt

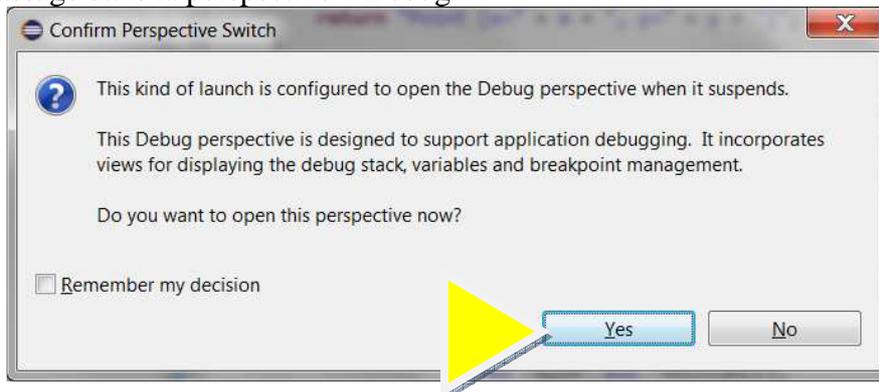
Un clic-droit sur un numéro de ligne permet le positionnement d'un point d'arrêt de l'exécution (ici ligne 42) :



L'exécution en mode « Debug »

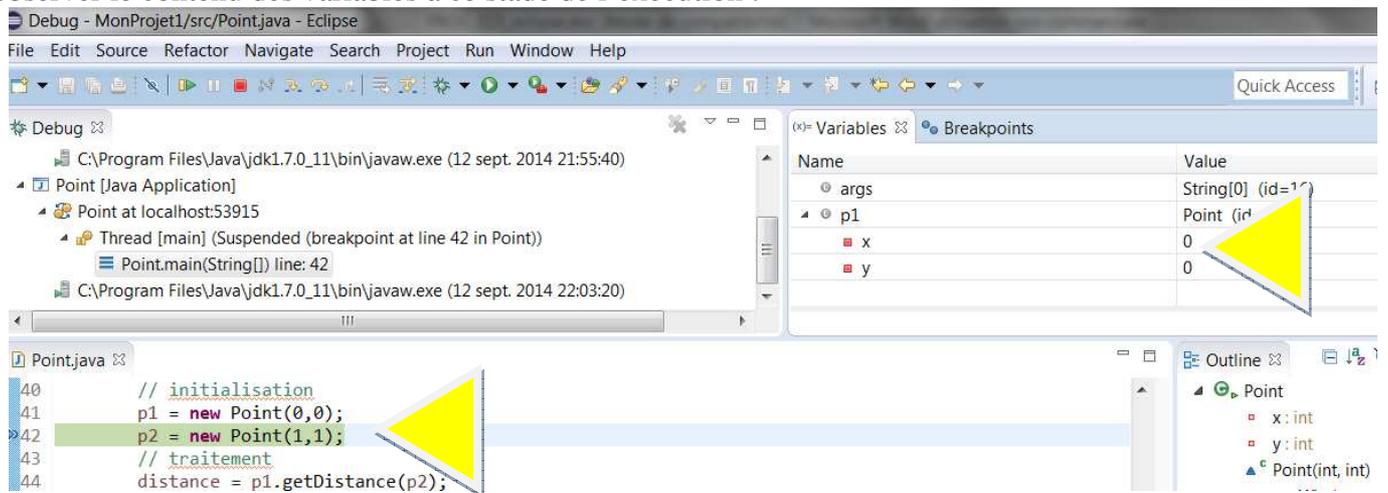


Avertissement de passage dans la perspective « Debug » :

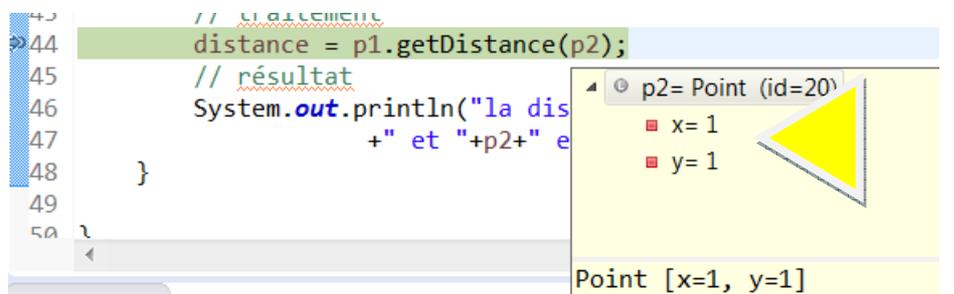


EDI Eclipse - Prise en main rapide

La perspective « Debug » est activée, l'exécution a lieu jusqu'au prochain point d'arrêt. On peut alors observer le contenu des variables à ce stade de l'exécution :

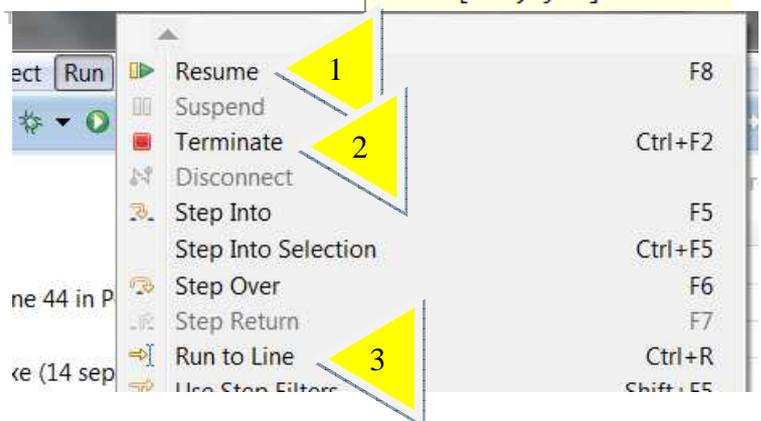


Le passage de la souris sur une variable (*type de base ou classe*) permet également d'afficher des informations sur la valeur courante de cette variable : ici passage de la souris au-dessus de p2 :



Après un point d'arrêt de l'exécution, on peut :

1. poursuivre l'exécution, jusqu'au prochain point d'arrêt
2. arrêter l'exécution
3. sélectionner une ligne puis relancer l'exécution jusqu'à cette ligne, etc.



Après arrêt de l'exécution, on peut revenir à l'édition du code Java afin d'effectuer des modifications puis de recompiler (ici automatique).

Pour revenir à la perspective « Java » :

