Premiers pas avec Eclipse :

INTRODUCTION

Eclipse est un IDE, *Integrated Development Environment* (EDI environnement de développement intégré en français), c'est-à-dire un logiciel qui simplifie la programmation en proposant un certain nombre de raccourcis et d'aide à la programmation. Il est développé par IBM, est gratuit et disponible pour la plupart des systèmes d'exploitation.

Au fur et à mesure que vous programmez, eclipse compile automatiquement le code que vous écrivez, en soulignant en rouge ou jaune les problème qu'il décèle. Il souligne en rouge les parties du programme qui ne compilent pas, et en jaune les parties qui compilent mais peuvent éventuellement poser problème (on dit qu'eclipse lève un avertissement, ou *warning* en anglais). Pendant l'écriture du code, cela peut sembler un peu déroutant au début, puisque tant que la ligne de code n'est pas terminée (en gros jusqu'au point-virgule), eclipse indique une erreur dans le code.

Il est déconseillé de continuer d'écrire le programme quand il contient des erreurs, car eclipse est dans ce cas moins performant pour vous aider à écrire le programme.

INSTALLATION

Si vous souhaitez l'installer chez vous, eclipse est disponible sur le site http://www.eclipse.org pour linux, BSD, Windows ou MacOS. Normalement, la page http://www.eclipse.org/downloads choisit automatiquement la bonne version.

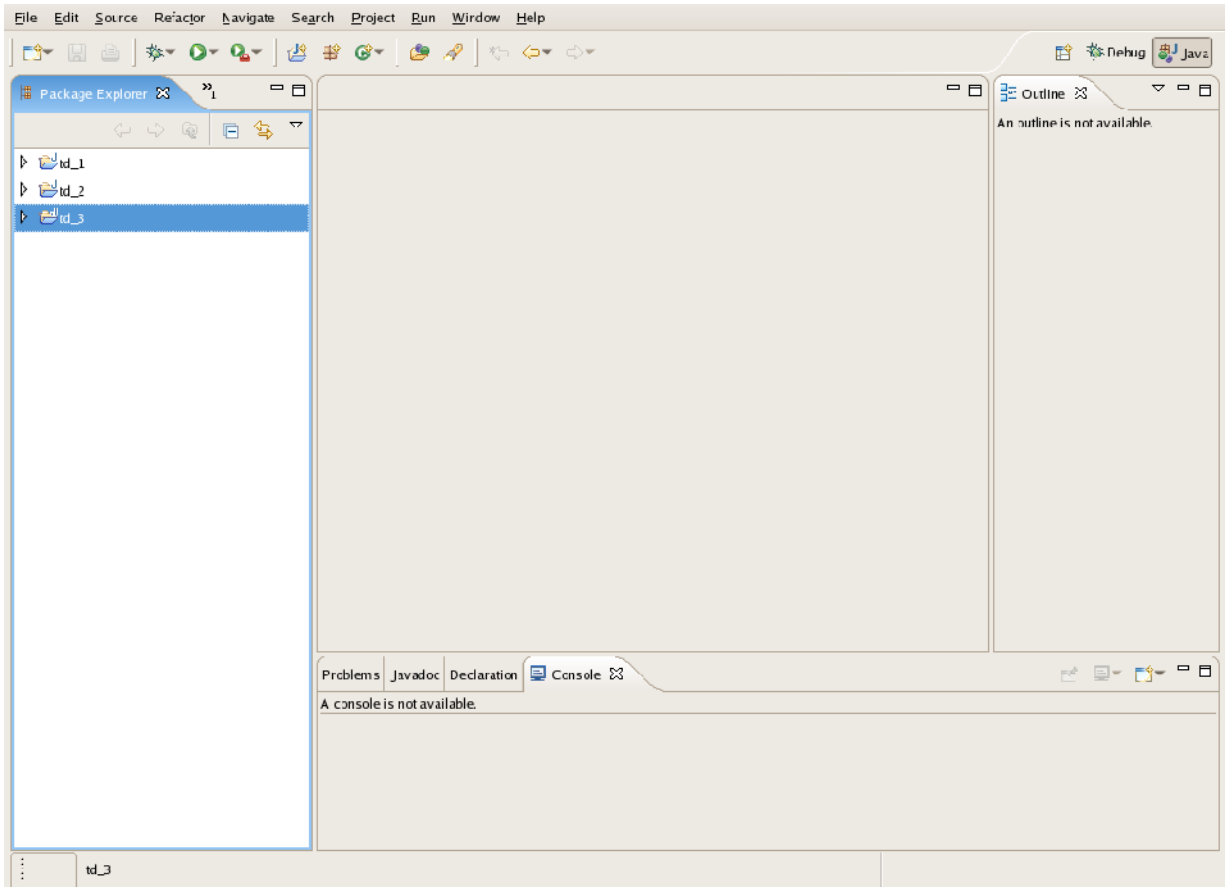
Parmi ses concurrents, on trouve NetBeans (http://www.netbeans.org), gratuit et développé par Sun, IDEA de JetBrain (http://www.jetbrains.com) qui est payant.

Premiers contacts :

Le workspace est le répertoire dans lequel sont stockés :

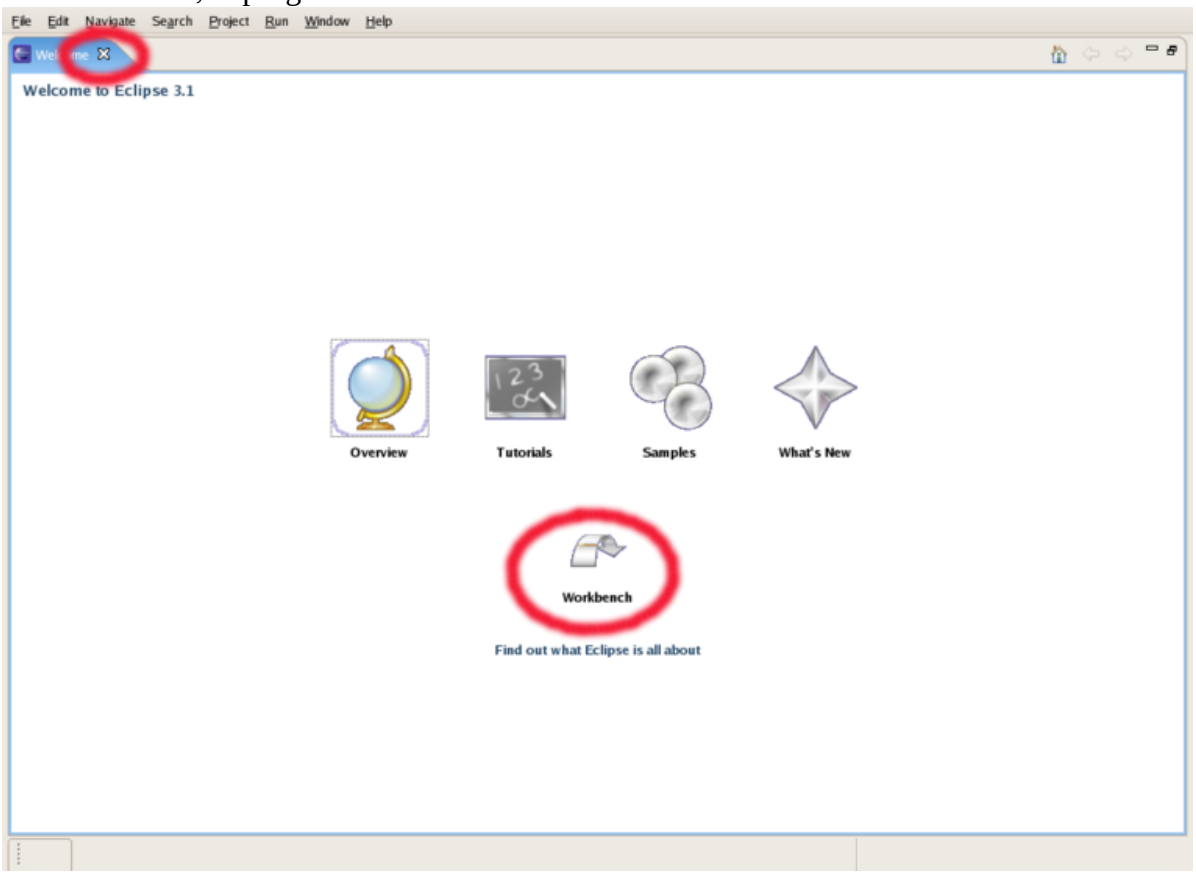
- les projets, c'est-à-dire les différents programmes que vous allez réaliser
- la configuration d'eclipse

Afin d'éviter de reconfigurer eclipse plusieurs fois, on évite en général de multiplier les workspaces.



Lors du lancement d'eclipse, un splash screen apparaît, suivi d'une fenêtre vous demandant dans quel *workspace* vous voulez travailler. L'emplacement par défaut est le répertoire `workspace` dans le répertoire principal, et peut être changé, sans créer le répertoire à l'avance.

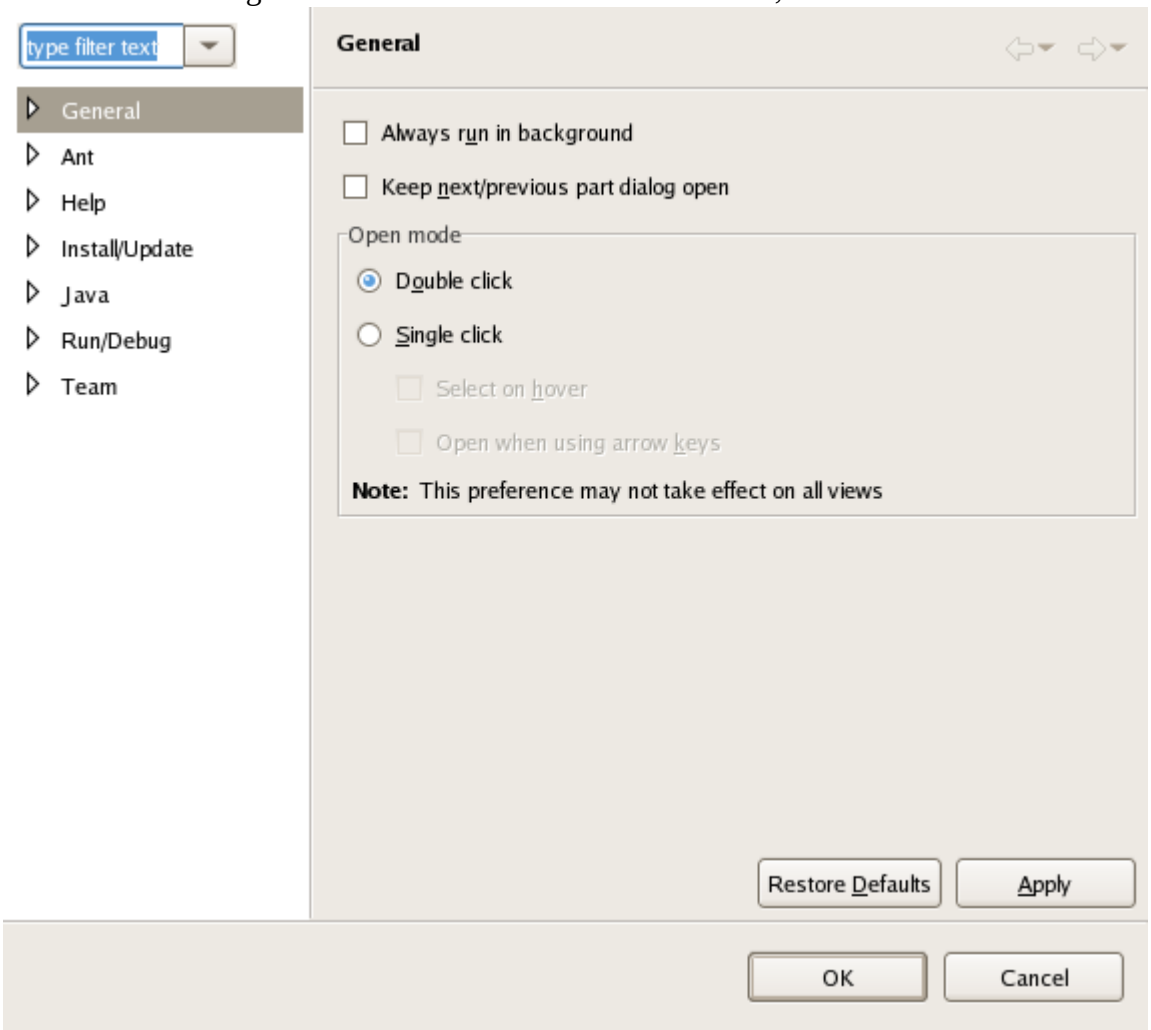
Une fois choisi, le programme s'initialise sur la fenêtre suivante :



Pour commencer à travailler, il suffit de fermer l'onglet ou de clicker sur Workbench.

Comme la plupart des logiciels aboutis, eclipse contient un nombre considérable d'options. La configuration par défaut est en général acceptable, mais il y a certains points qui méritent d'être changés, et d'autre qui doivent l'être (comme par exemple le fait que l'on utilise java 1.5 et non java 1.4).

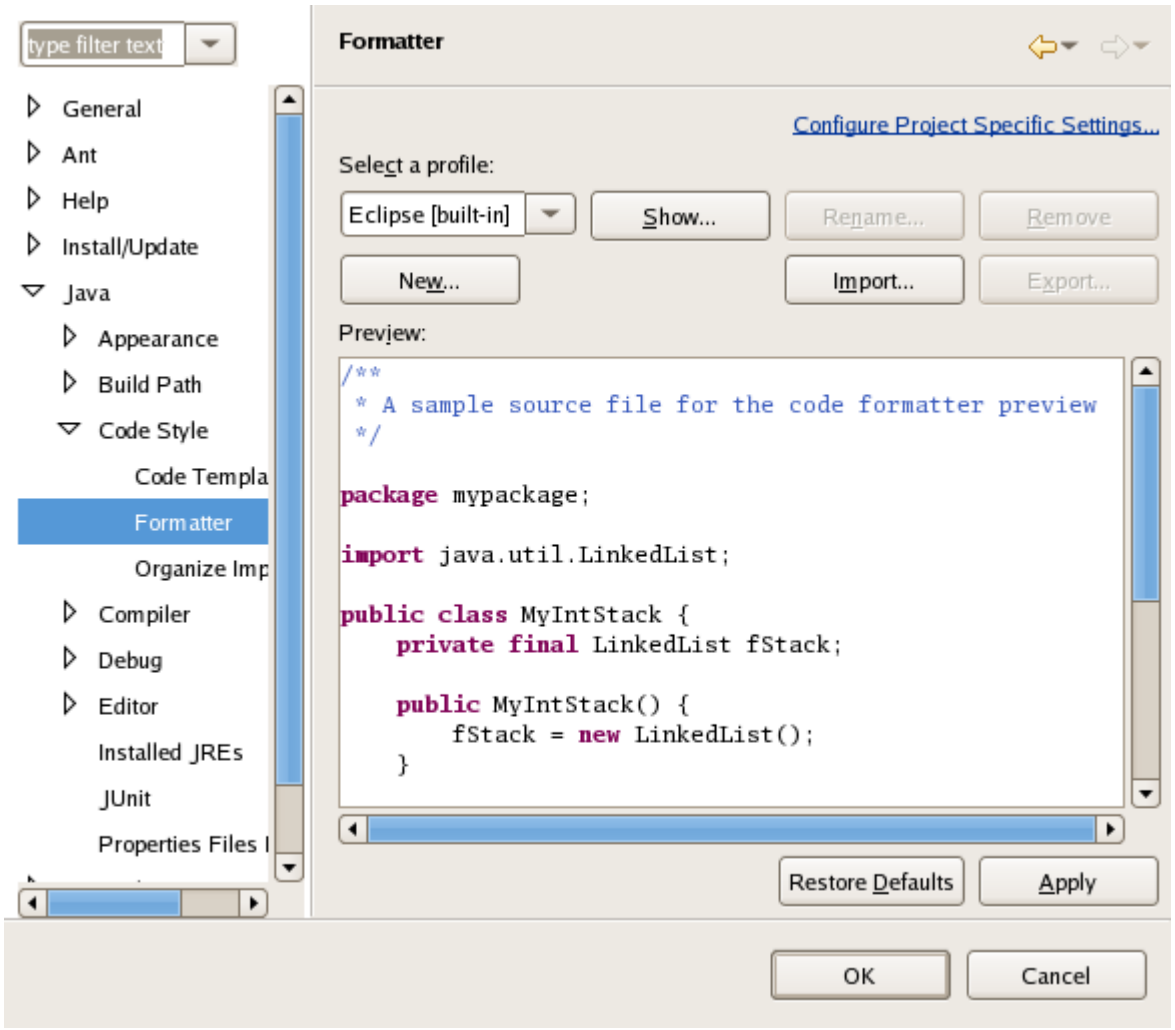
La fenêtre de configuration s'obtient dans le menu Windows, article Preferences...



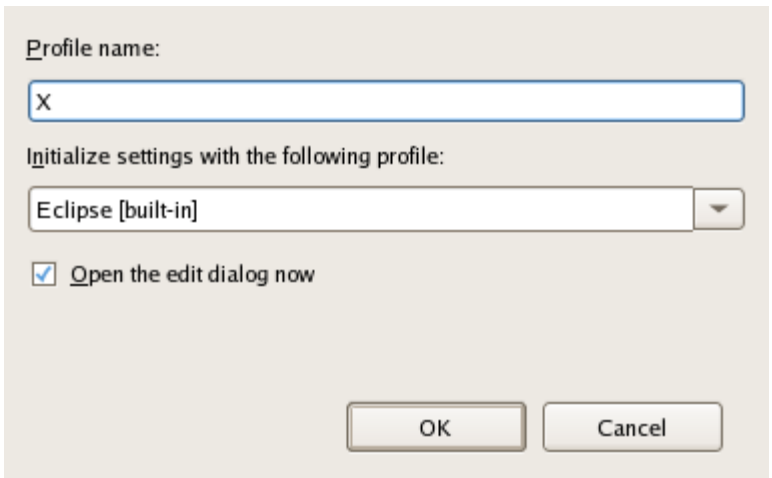
Les différentes pages d'options sont choisies dans l'arbre situé sur la gauche de la fenêtre, la page apparaissant sur la droite. Les options relatives à java sont dans le sous-arbre java, celles relatives à l'apparence d'eclipse (font, couleurs) sont dans le sous-arbre General.

Le formatage du code et l'indentation fondamental pour une relecture et un débogage aisé des programmes. Pour que le formatage soit conforme à ce que l'on attend, il convient de modifier la manière dont eclipse formate les programmes par défaut.

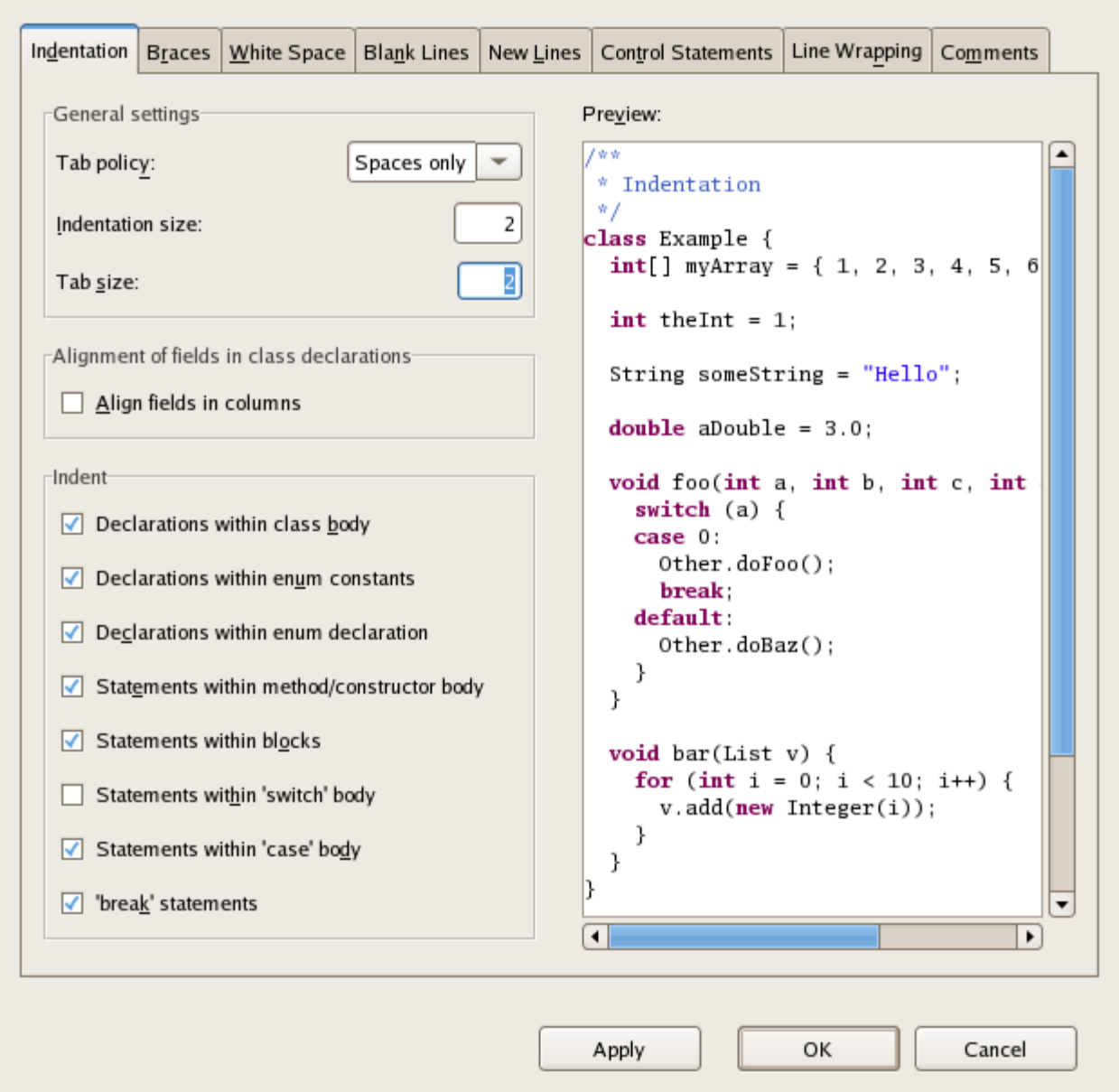
Pour cela, choisir, dans la fenêtre des préférences, la page Java -> Code Style -> Formatter. On obtient la fenêtre suivante :



Pour définir une nouvelle configuration de formatage, cliquer sur New..., et donner un nom à cette configuration dans la boîte qui s'ouvre, par exemple X :



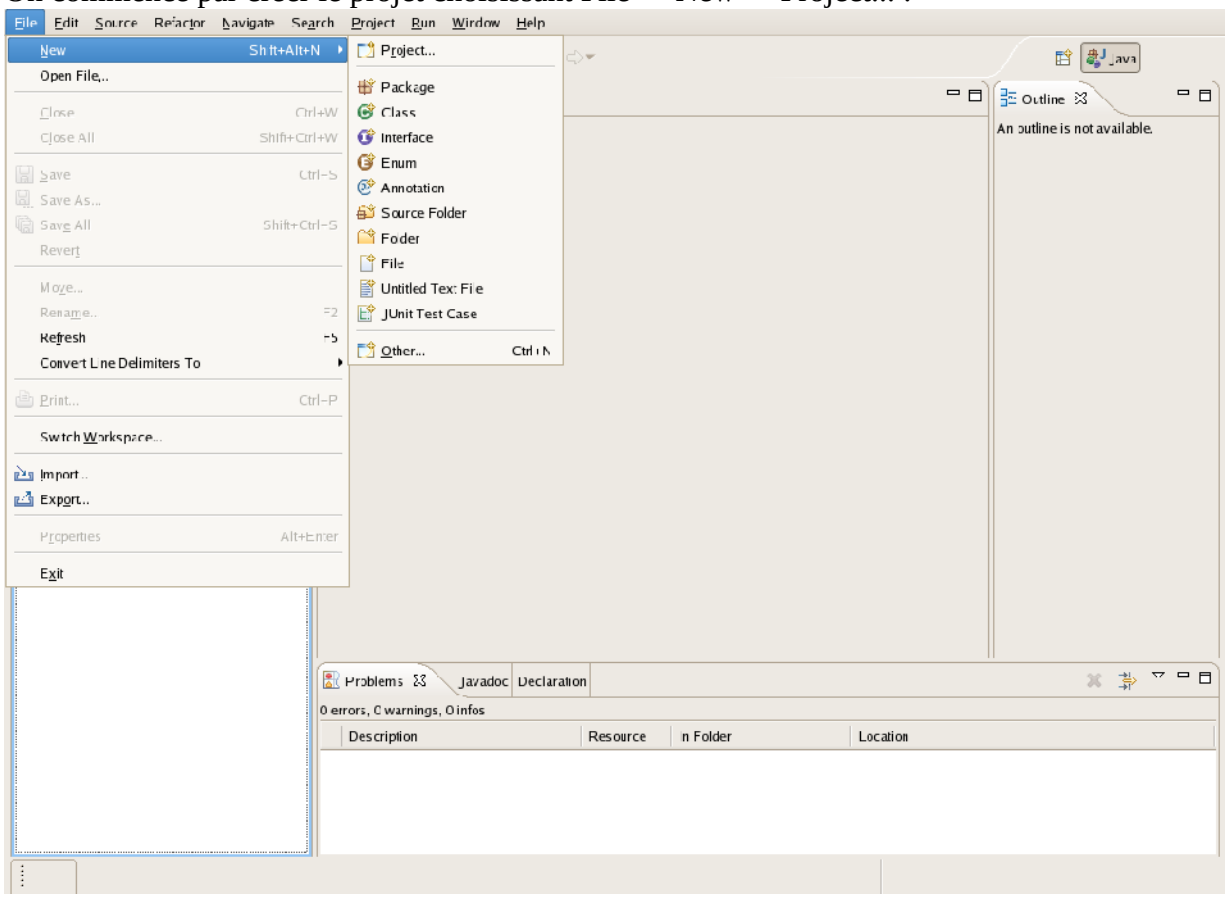
Un fois cliqué sur OK, on peut effectuer les modifications suivantes : dans le cadre General Settings, modifier Tab Policy en "Spaces only" puis indiquer 2 pour "indentation size" et "tab size".



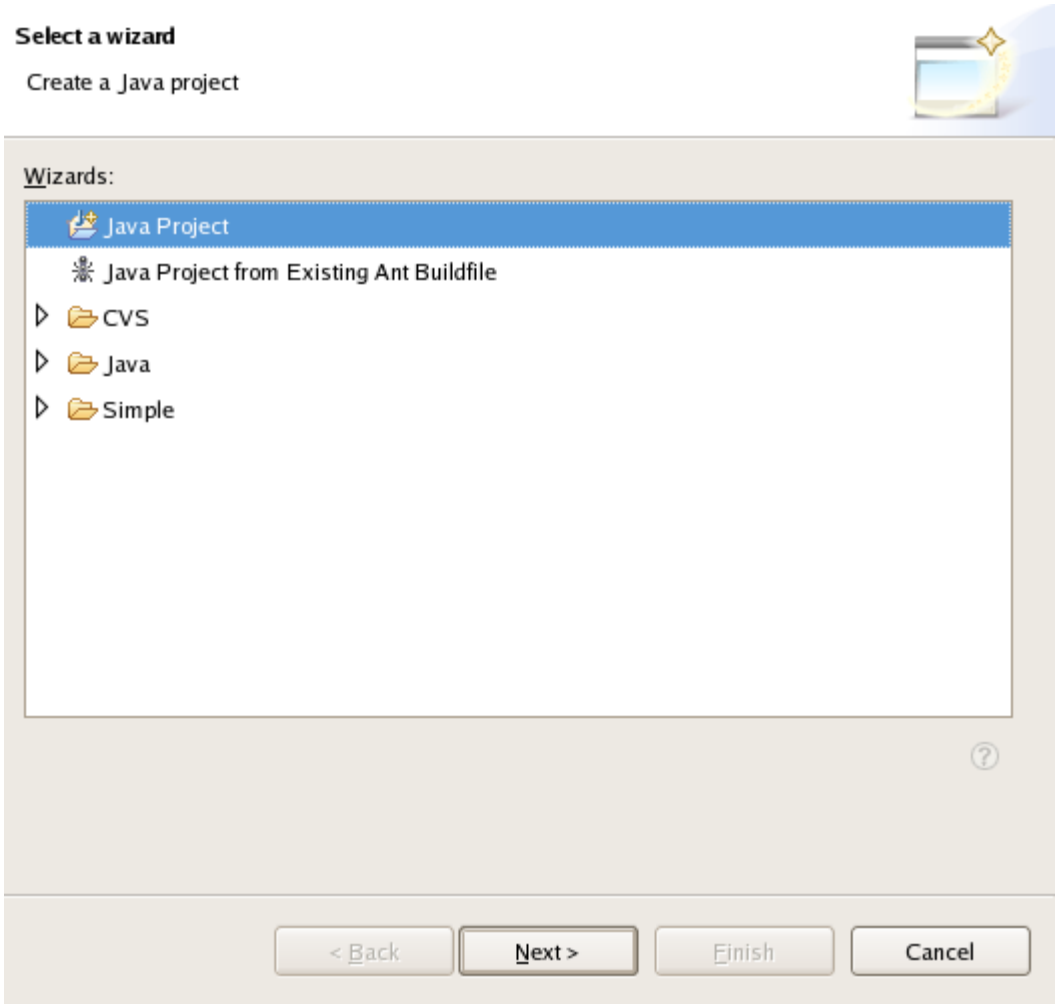
Une fois terminé, cliquer sur OK, puis à nouveau sur le bouton OK de la fenêtre de configuration.

Comme initiation, nous allons créer et exécuter le traditionnel programme Hello World, en Java, à l'aide d'eclipse.

On commence par créer le projet choisissant File -> New -> Project... :



On obtient alors un wizard (suite de boîtes de dialogues avec un bouton "Next>" pour passer d'une étape à la suivante). Tout d'abord, on indique que l'on veut créer un projet Java en choisissant "Java project", puis on clique sur "Next>" où l'on indique le nom du projet, par exemple hello. Bien que l'on puisse spécifier plus d'options en cliquant sur "Next>", on clique sur "Finish" pour choisir les options par défaut :


Select a wizard
Create a Java project
Wizards:
Java Project
Java Project from Existing Ant Buildfile
CVS
Java
Simple
< Back
Next >
Finish
Cancel

Create a Java project

Create a Java project in the workspace or in an external location.

Project name: hello

Contents

- (•) Create new project in workspace
- ( ) Create project from existing source

Directory: /tmp/tmp/hello [Browse...]

JDK Compliance

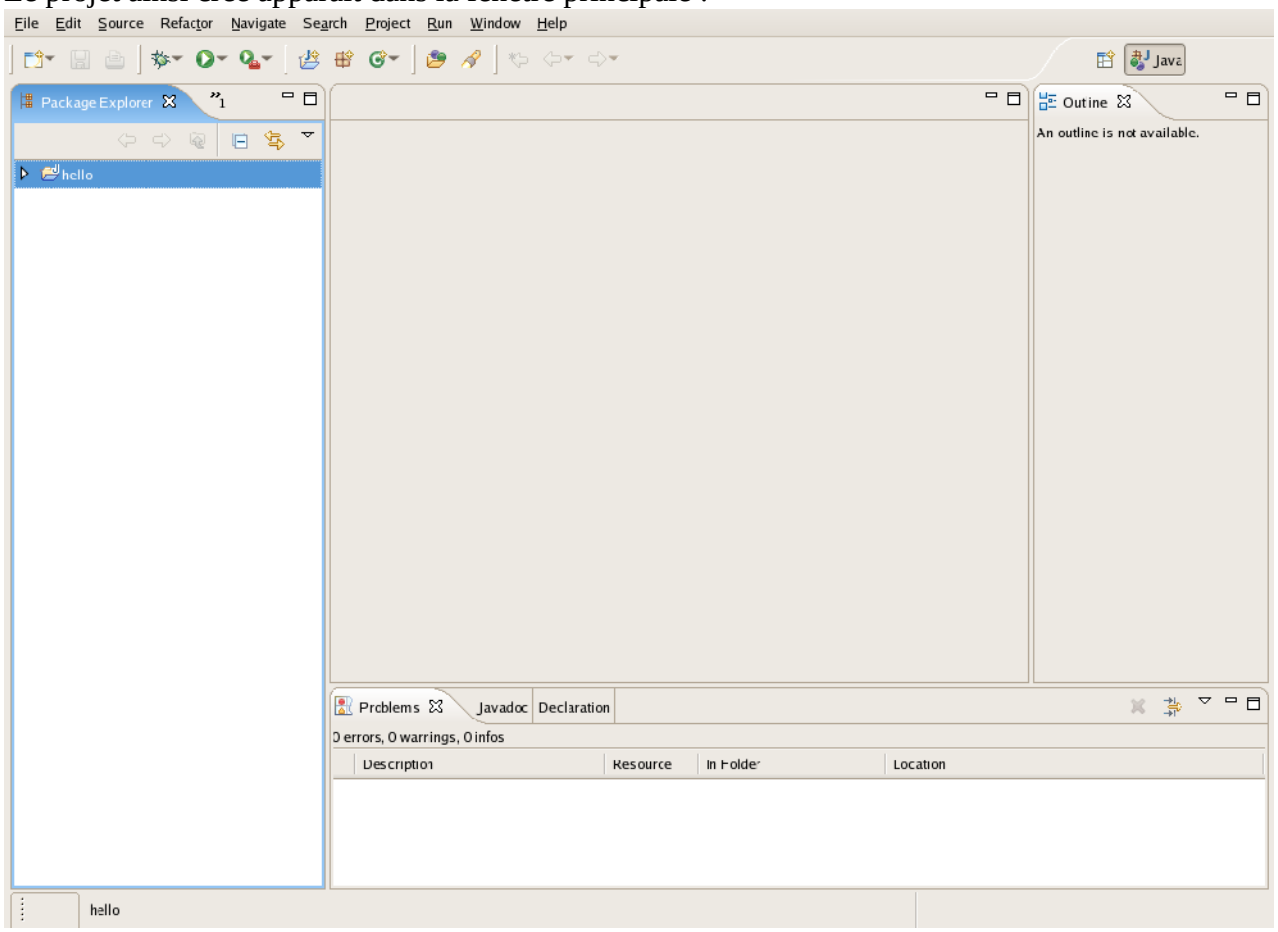
- (•) Use default compiler compliance (Currently 1.4) Configure default...
- ( ) Use a project specific compliance: 1.4

Project layout

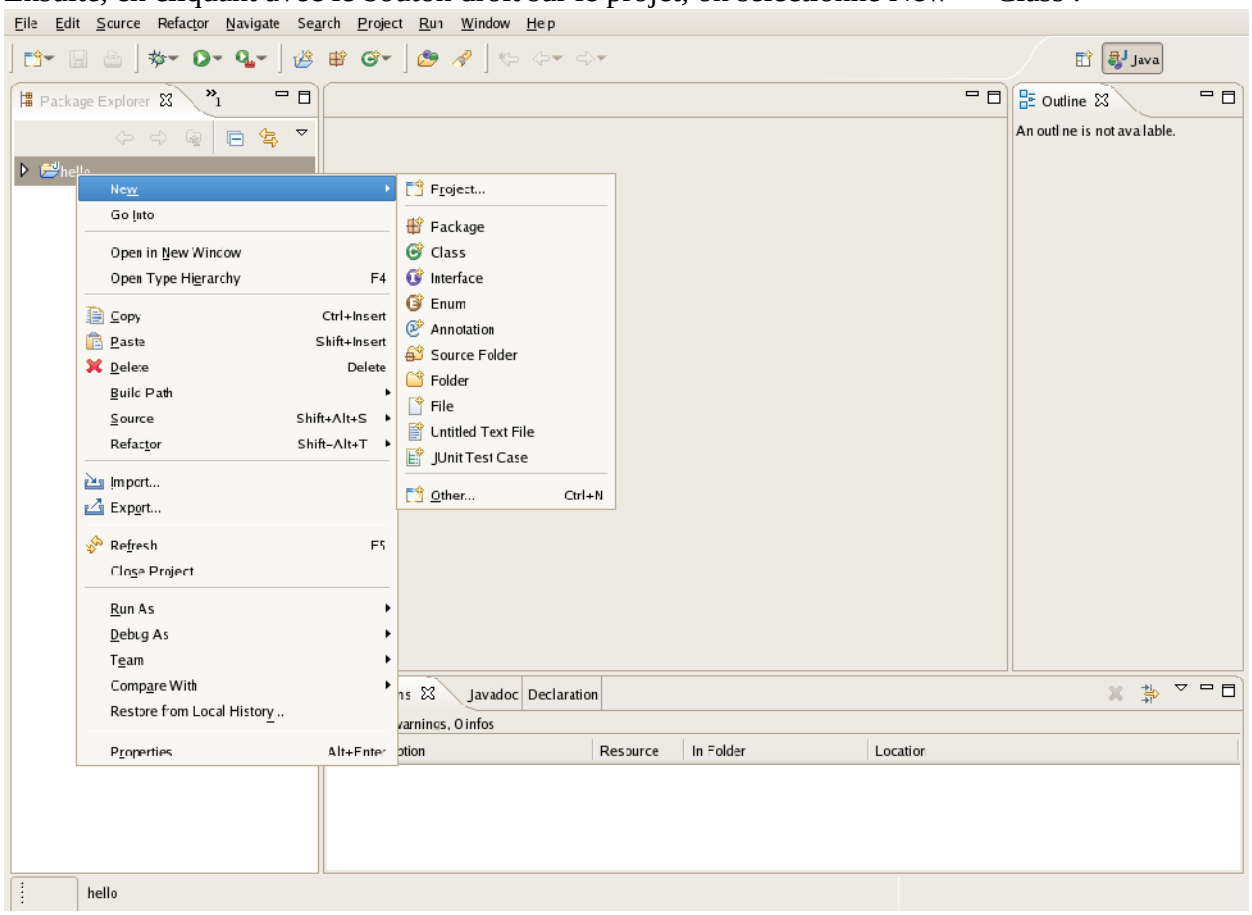
- (•) Use project folder as root for sources and class files
- ( ) Create separate source and output folders Configure default...

[< Back] [Next >] [Finish] [Cancel]

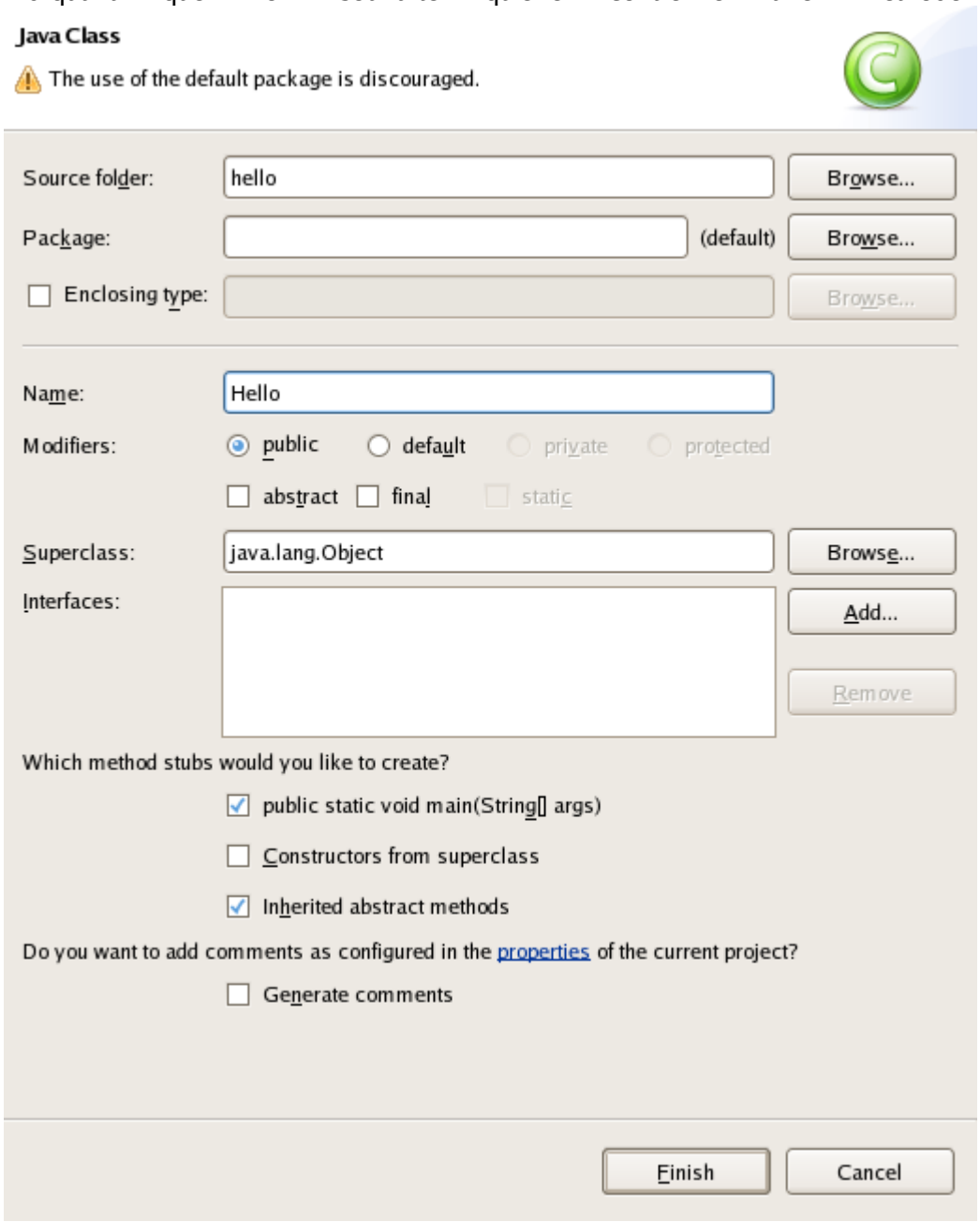
Le projet ainsi créé apparaît dans la fenêtre principale :



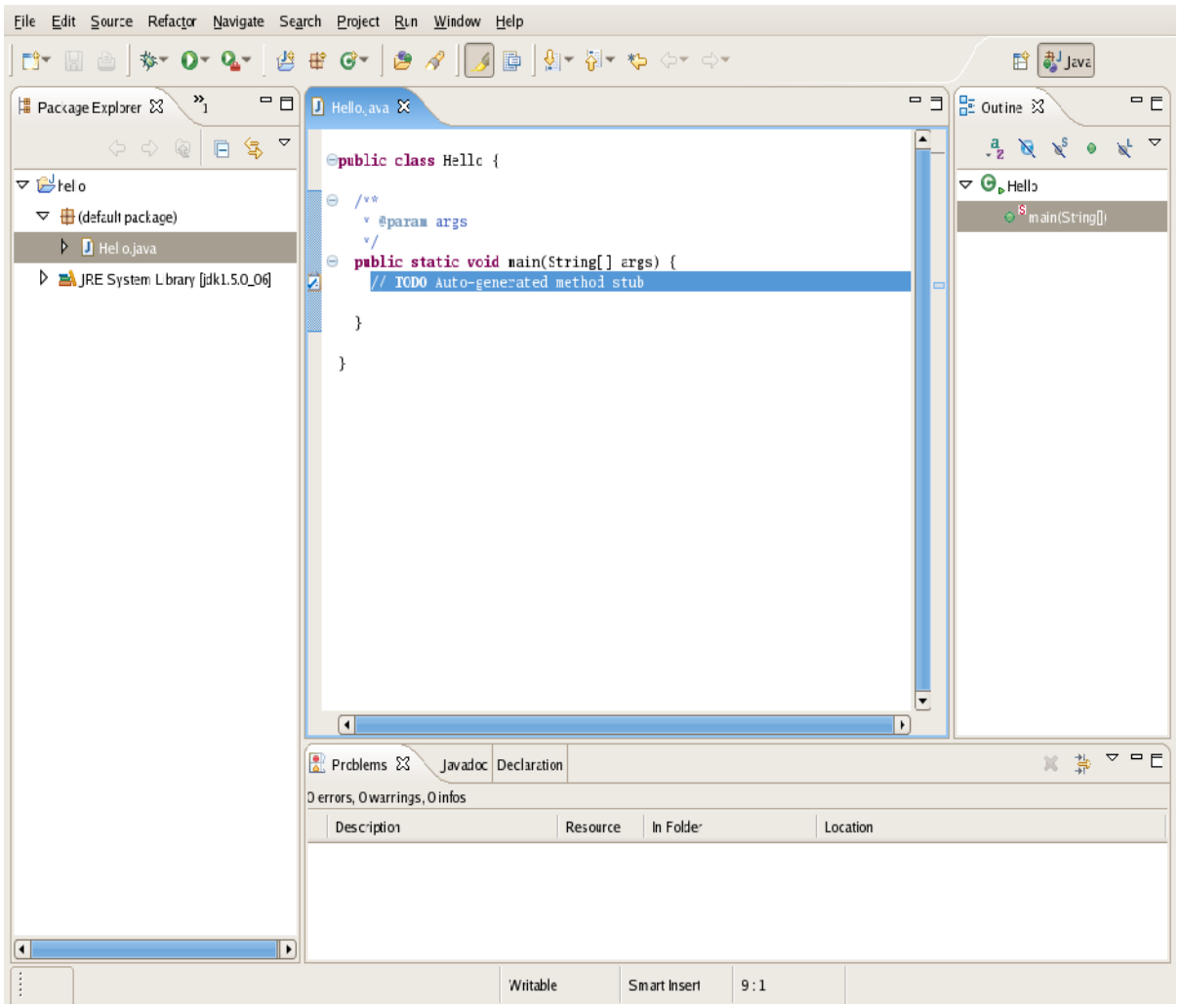
Ensuite, en cliquant avec le bouton droit sur le projet, on sélectionne New -> Class :



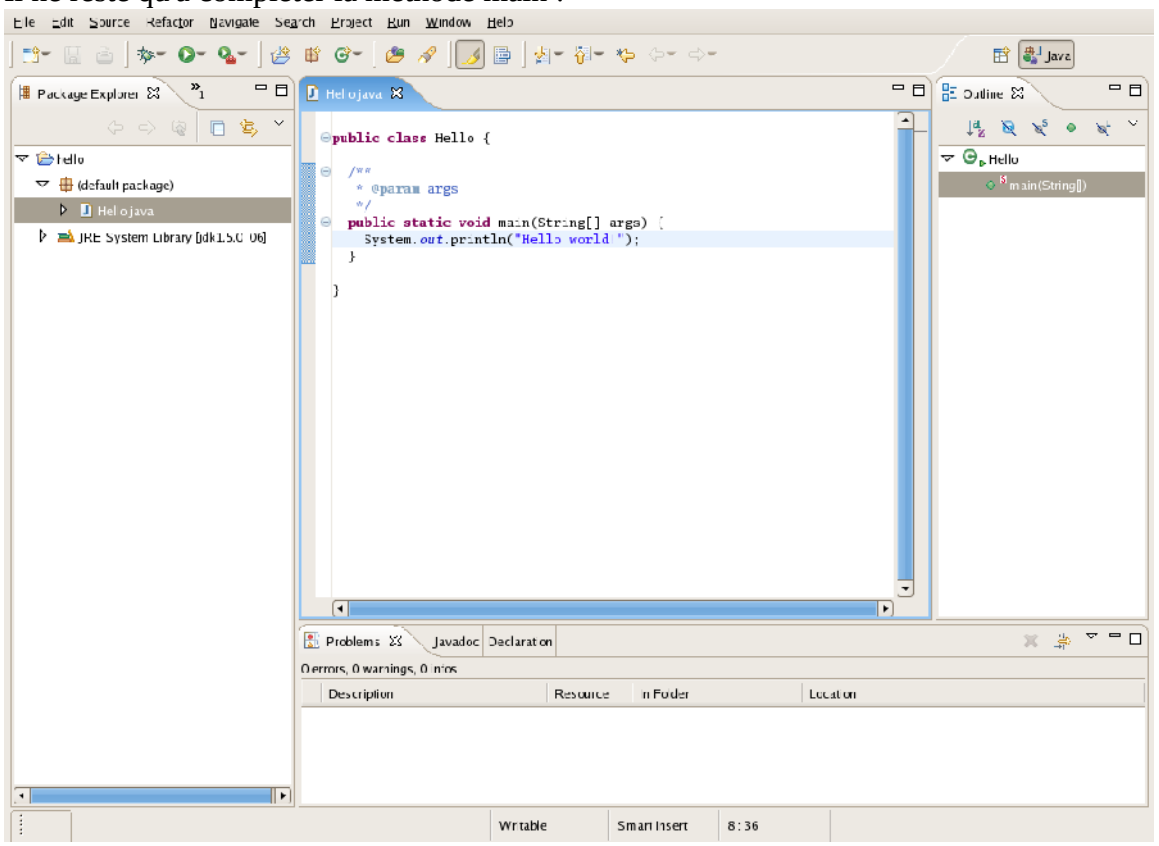
Dans la fenêtre qui s'ouvre, on indique le nom de la classe, Hello, et on coche la case indiquant que l'on souhaite qu'elle contienne une méthode `main` :



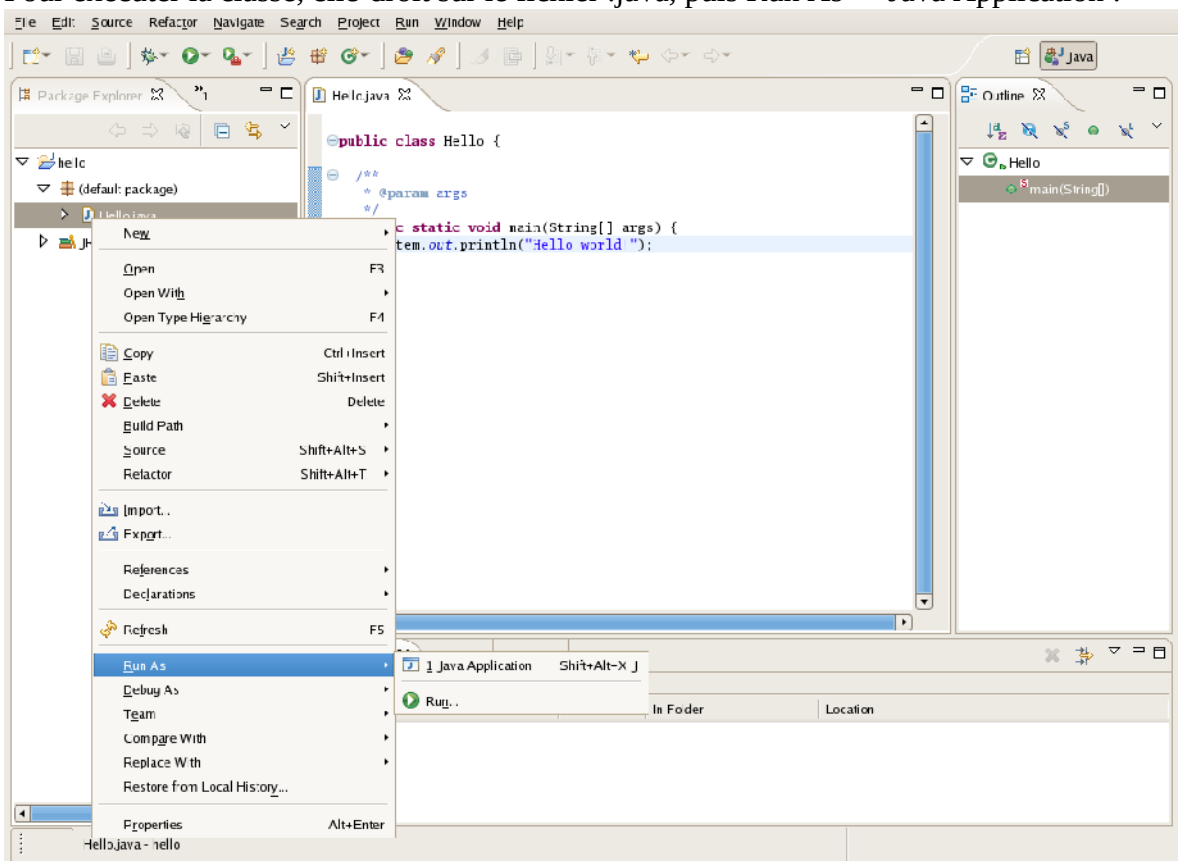
Eclipse crée le squelette de la classe (stub en anglais). Notez le commentaire javadoc préécrit (voir cette section pour plus d'informations), et le commentaire en TODO rappelé dans la marge par l'icône .


File Edit Source Refactor Navigate Search Project Run Window Help
Java
Package Explorer
Hello.java
Outline
Hello
(default package)
Hello.java
JRE System Library [jdk1.5.0_06]
public class Hello {
/**
* @param args
*/
public static void main(String[] args) {
// TODO Auto-generated method stub
}
}
main(String[])
Problems
Javadoc
Declaration
0 errors, 0 warnings, 0 infos
Description
Resource
In Folder
Location
Writable
Smart Insert
9 : 1

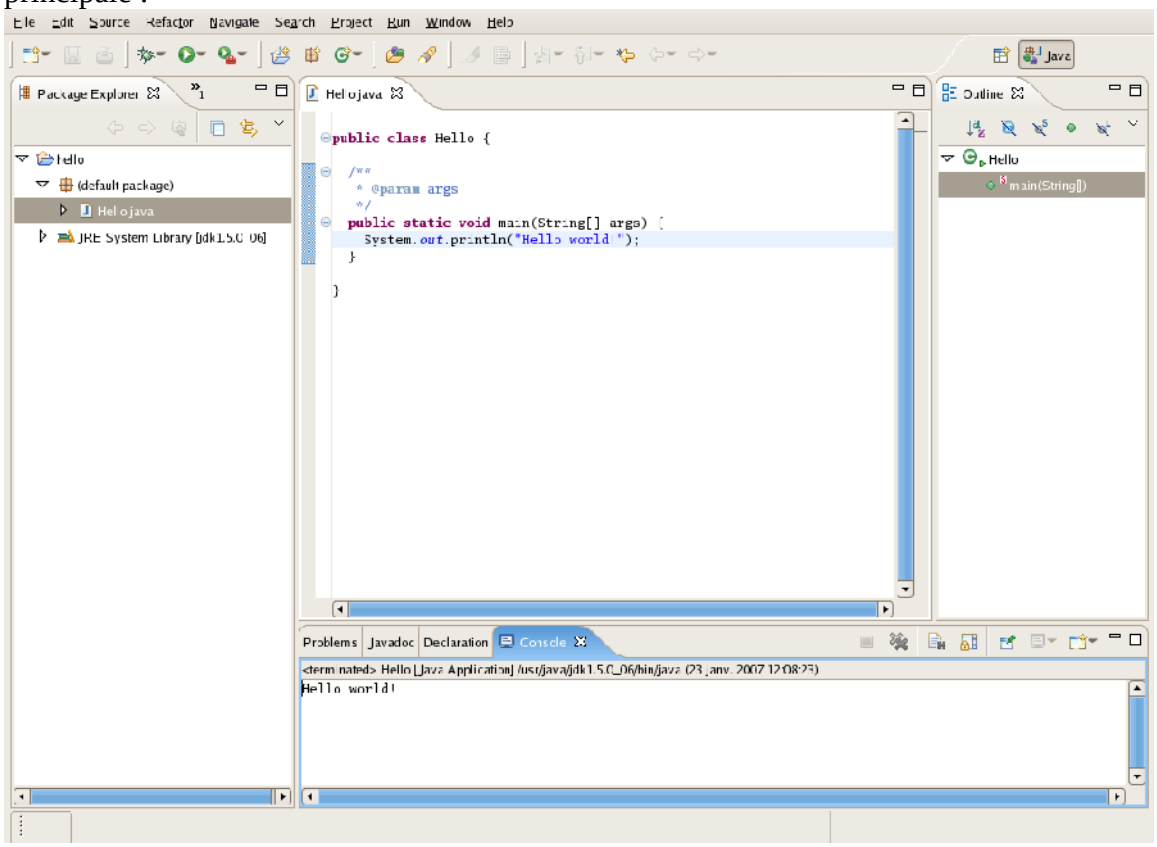
Il ne reste qu'à compléter la méthode main :



Pour exécuter la classe, clic-droit sur le fichier .java, puis Run As -> Java Application :

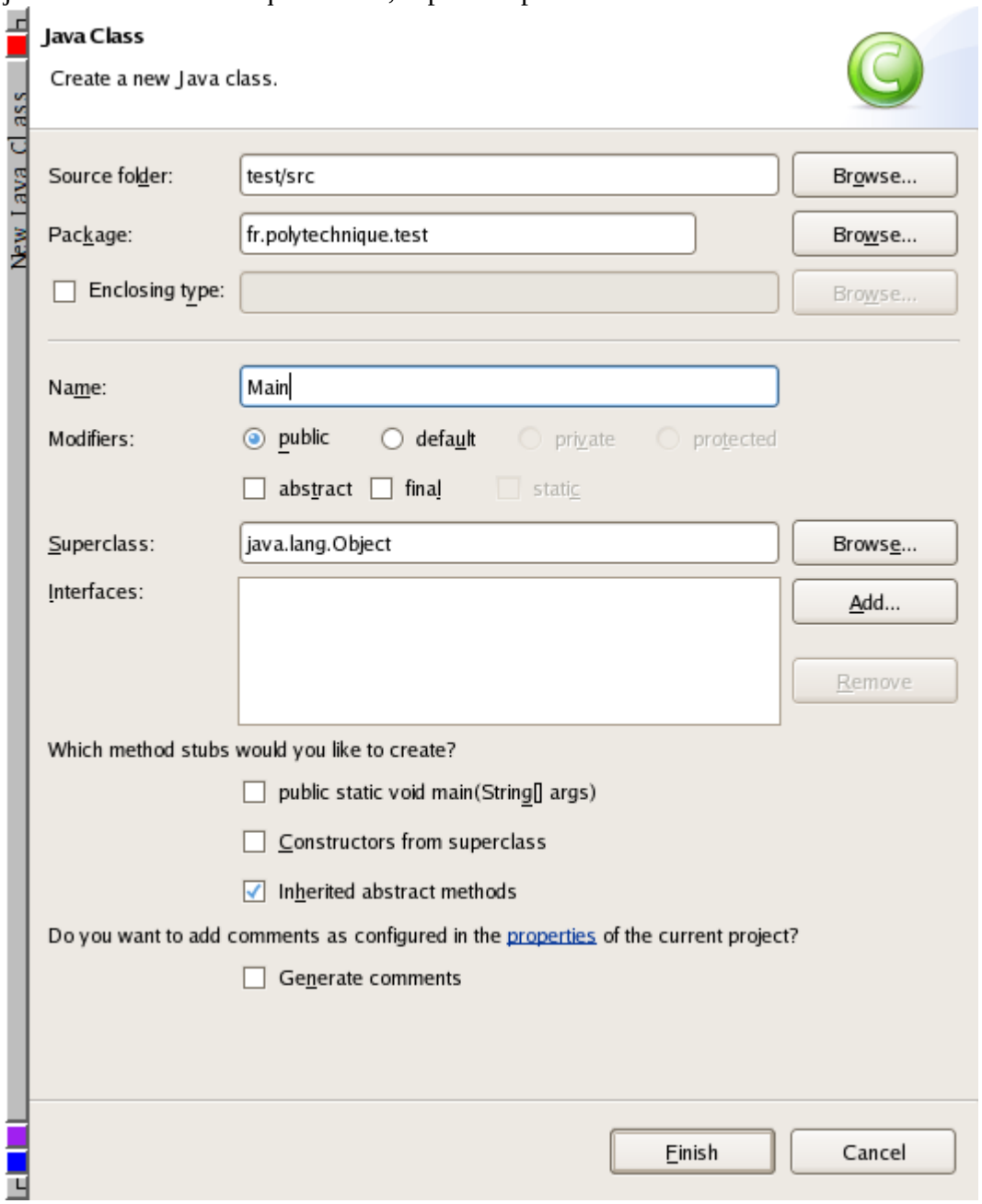


L'affichage de l'exécution s'effectue sous l'onglet "console" en bas de la fenêtre principale :

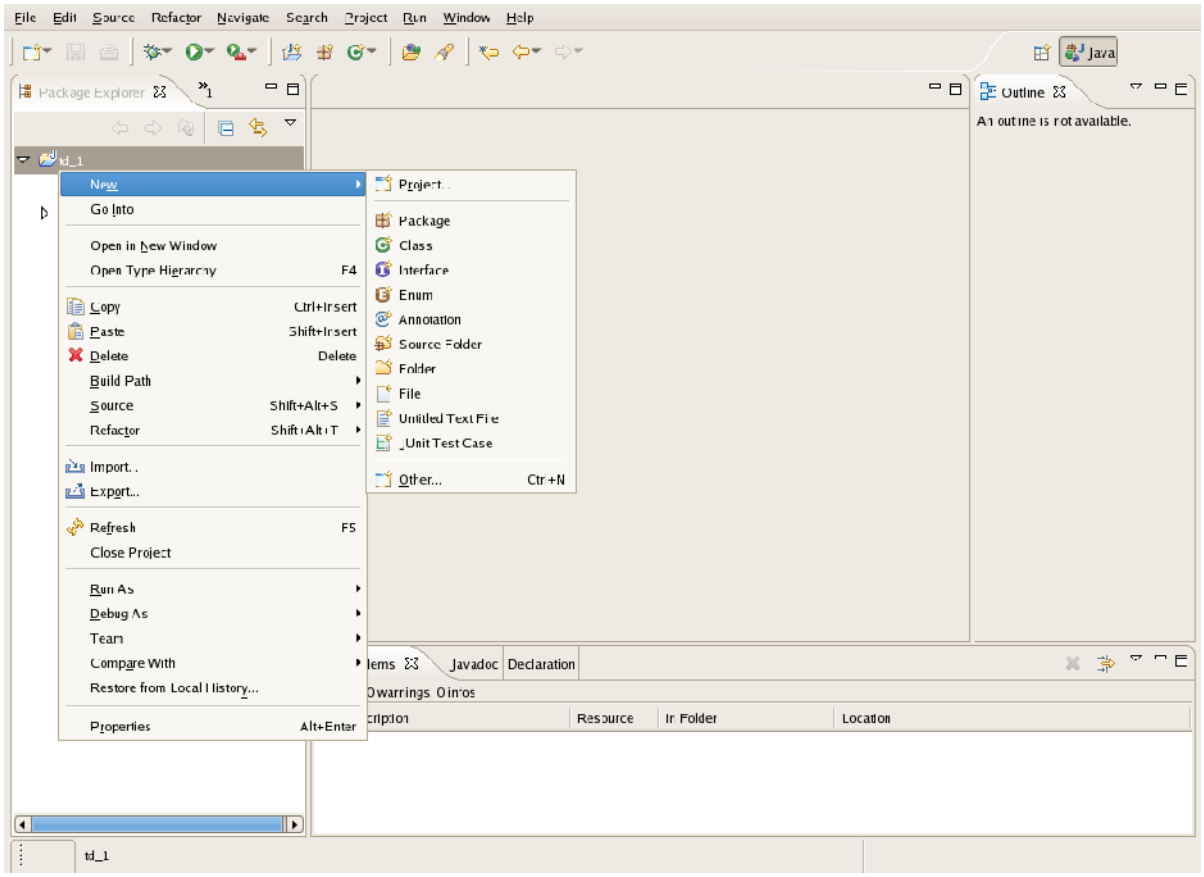


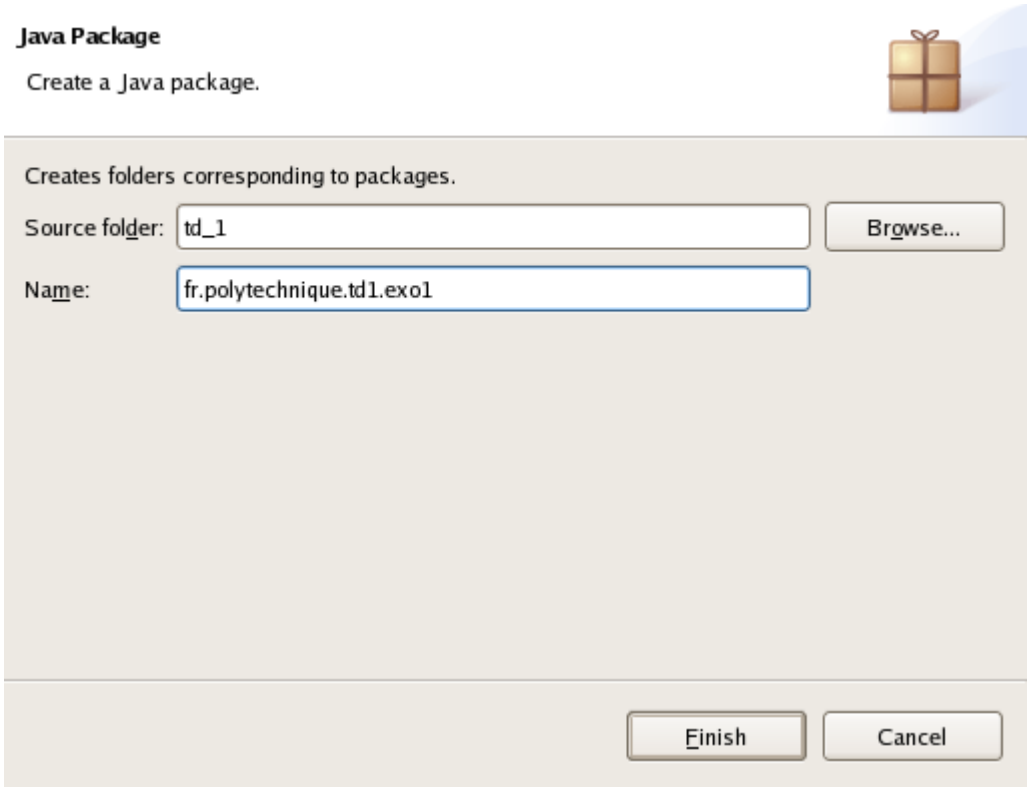
Pour créer une nouvelle classe, clic droit, soit sur le projet, soit sur le répertoire des sources, soit sur un paquetage. Un wizard s'ouvre proposant diverses options, dont le nom de la classe, le nom du paquetage de la classe (pré-rempli si le clic droit était sur un paquetage), avec par exemple si on veut un méthode main. Eclipse crée alors le fichier

java à l'endroit qu'il faut, pré-rempli avec les informations demandées :



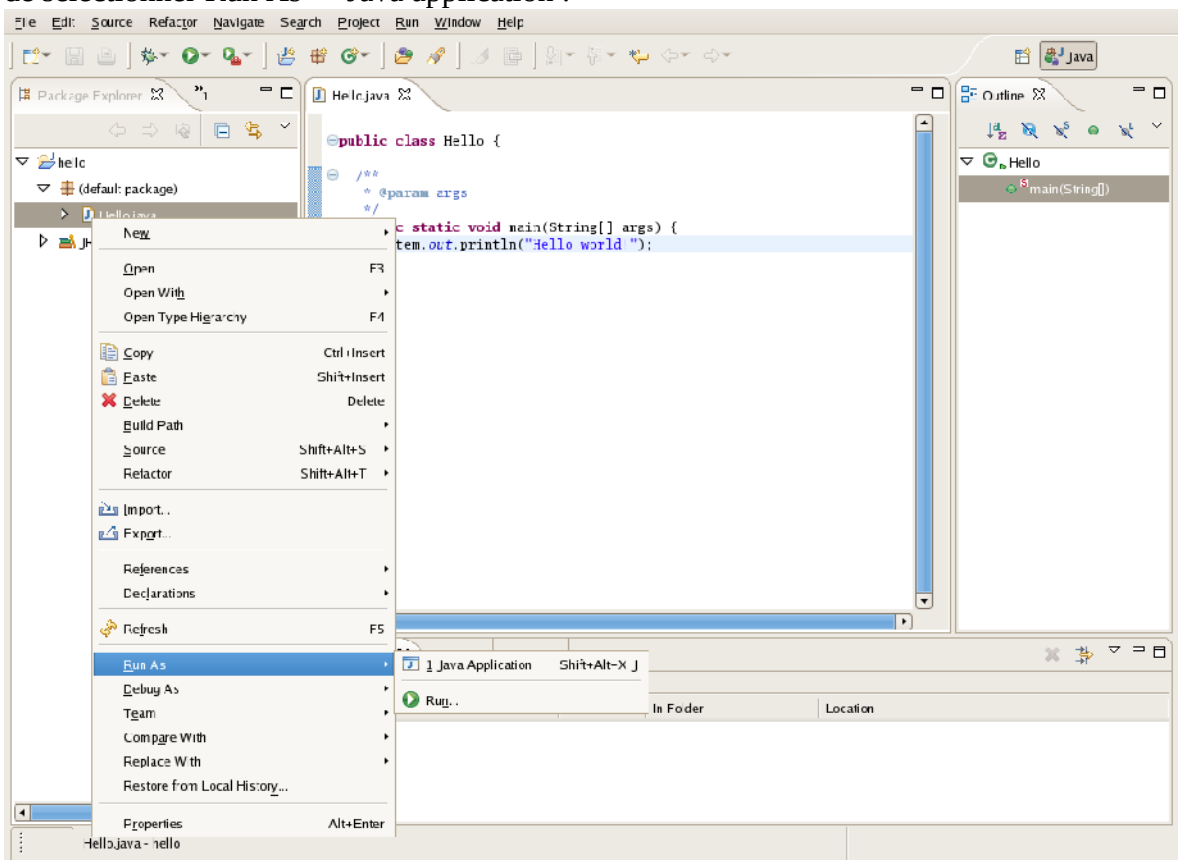
Avant d'ajouter des classes, on peut créer des paquetages pour les y mettre. Pour cela, clic droit sur le répertoire des sources, puis new puis paquage. Un wizard demandant le nom de paquetage s'ouvre. Le paquetage apparaît ensuite dans le répertoire des sources :


File Edit Source Refactor Navigate Search Project Run Window Help
Java
Package Explorer
Outline
An outline is not available.
td_1
New
Go Into
Open in New Window
Open Type Hierarchy F4
Copy Ctrl+Insert
Paste Shift+Insert
Delete Delete
Build Path
Source Shift+Alt+S
Refactor Shift+Alt+T
Import...
Export...
Refresh F5
Close Project
Run As
Debug As
Team
Compare With
Restore from Local History...
Properties Alt+Enter
Project...
Package
Class
Interface
Enum
Annotation
Source Folder
Folder
File
Untitled Text File
JUnit Test Case
Other... Ctrl+N
Javadoc
Declaration
Resource
Location
td_1

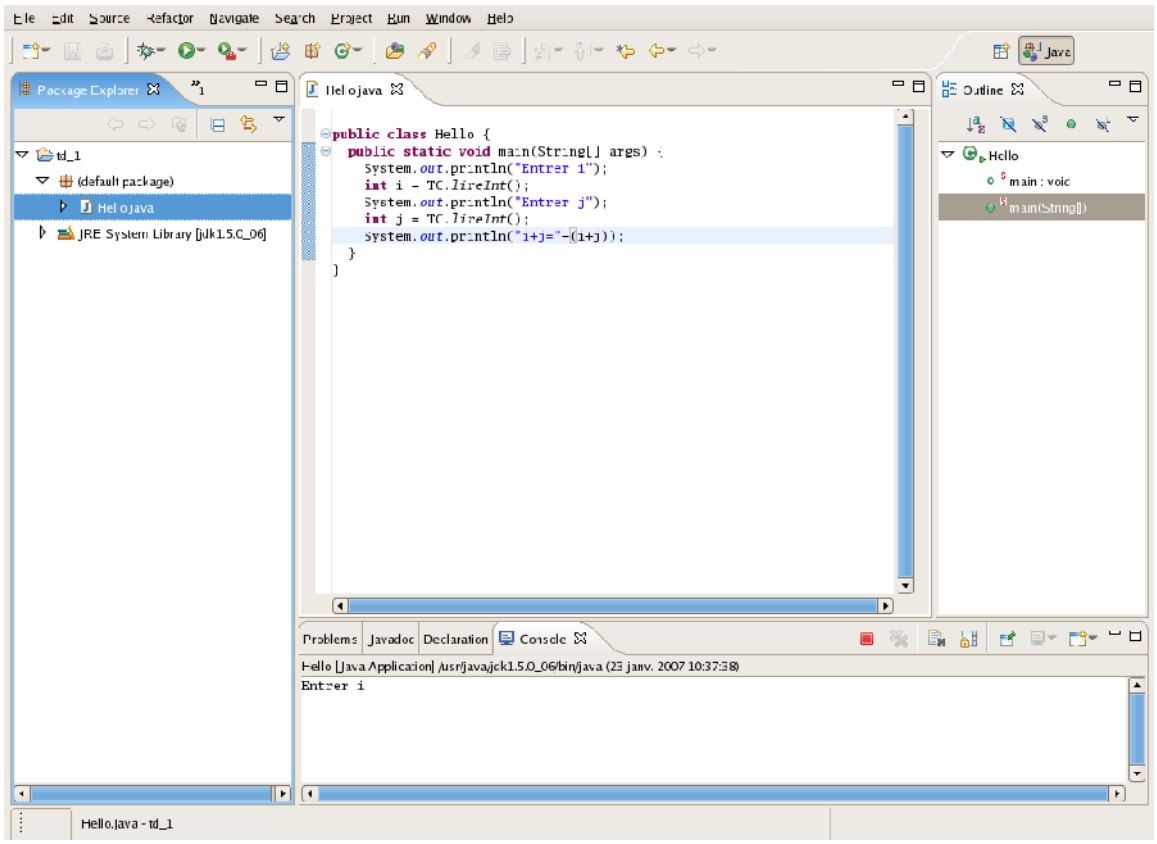

Java Package
Create a Java package.
Creates folders corresponding to packages.
Source folder: td_1
Browse...
Name: fr.polytechnique.td1.exo1
Finish
Cancel

Pour déplacer des classes entre paquetages, il suffit de le faire par glisser-déplacer, ou avec un clic-droit sur la classe puis Refactor -> Move...

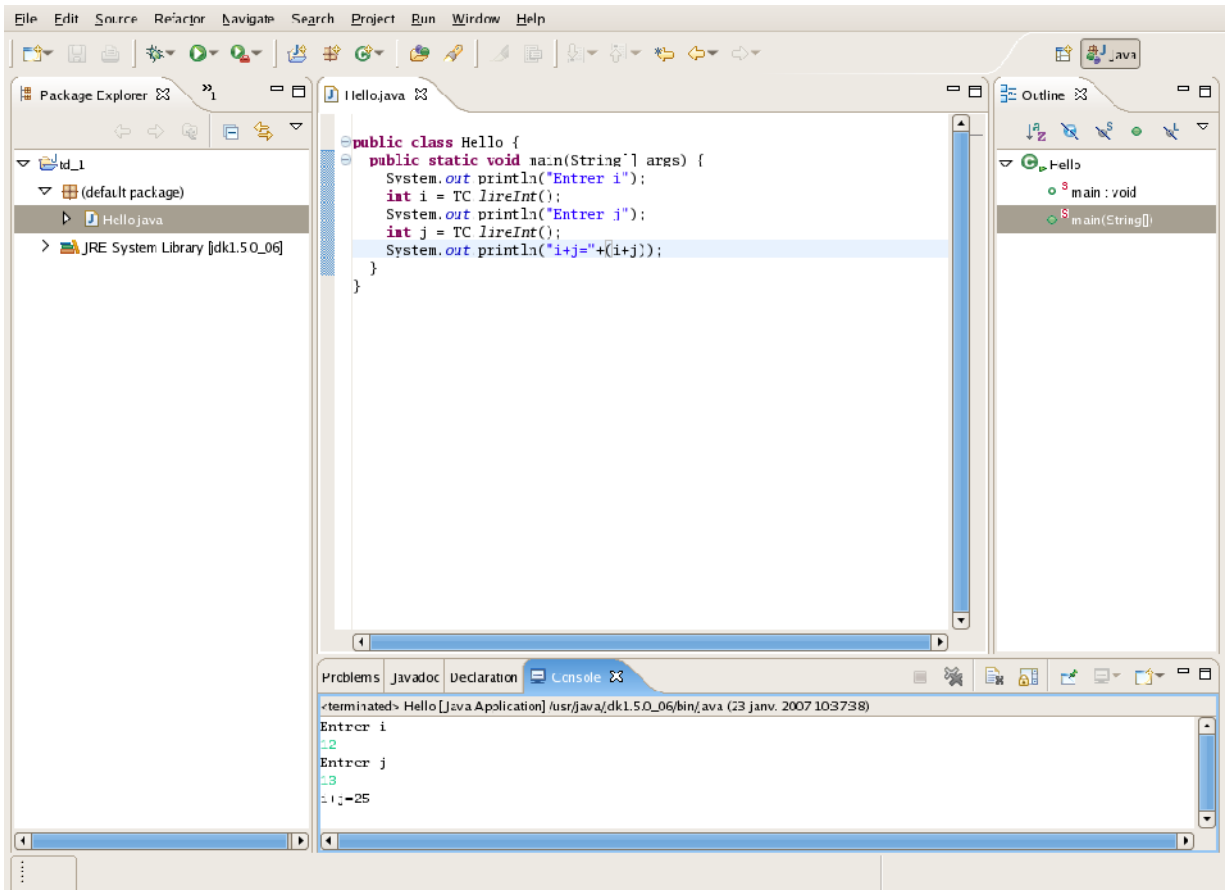
Pour exécuter un programme, il suffit de faire un clic-droit sur le .java de la classe, puis de sélectionner Run As -> Java application :



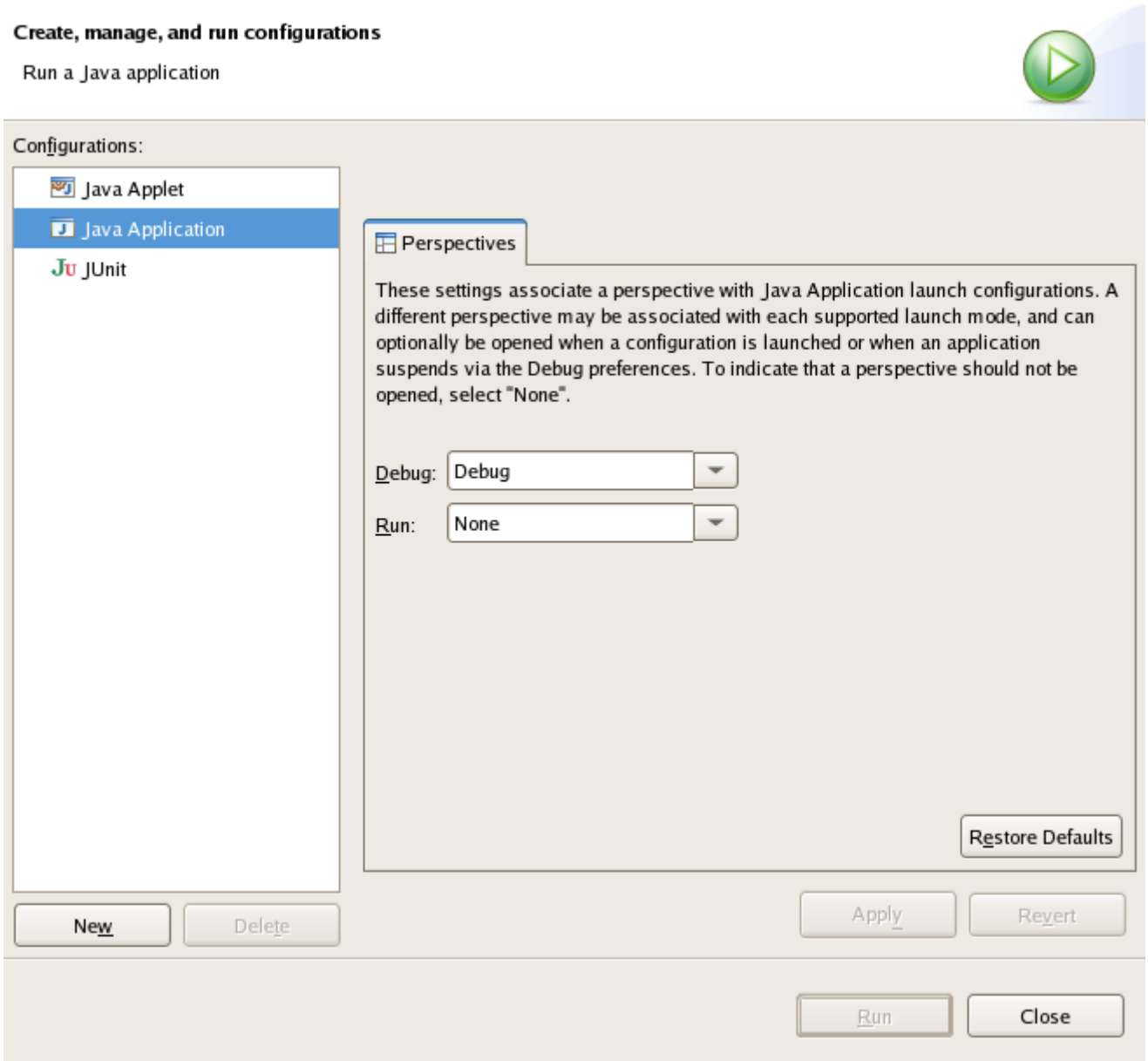
Il est alors possible d'intéragir avec le programme sous l'onglet "Console", en bas de la fenêtre principale. Par exemple, en utilisant la classe `TC`, on peut écrire et lancer le programme suivant :



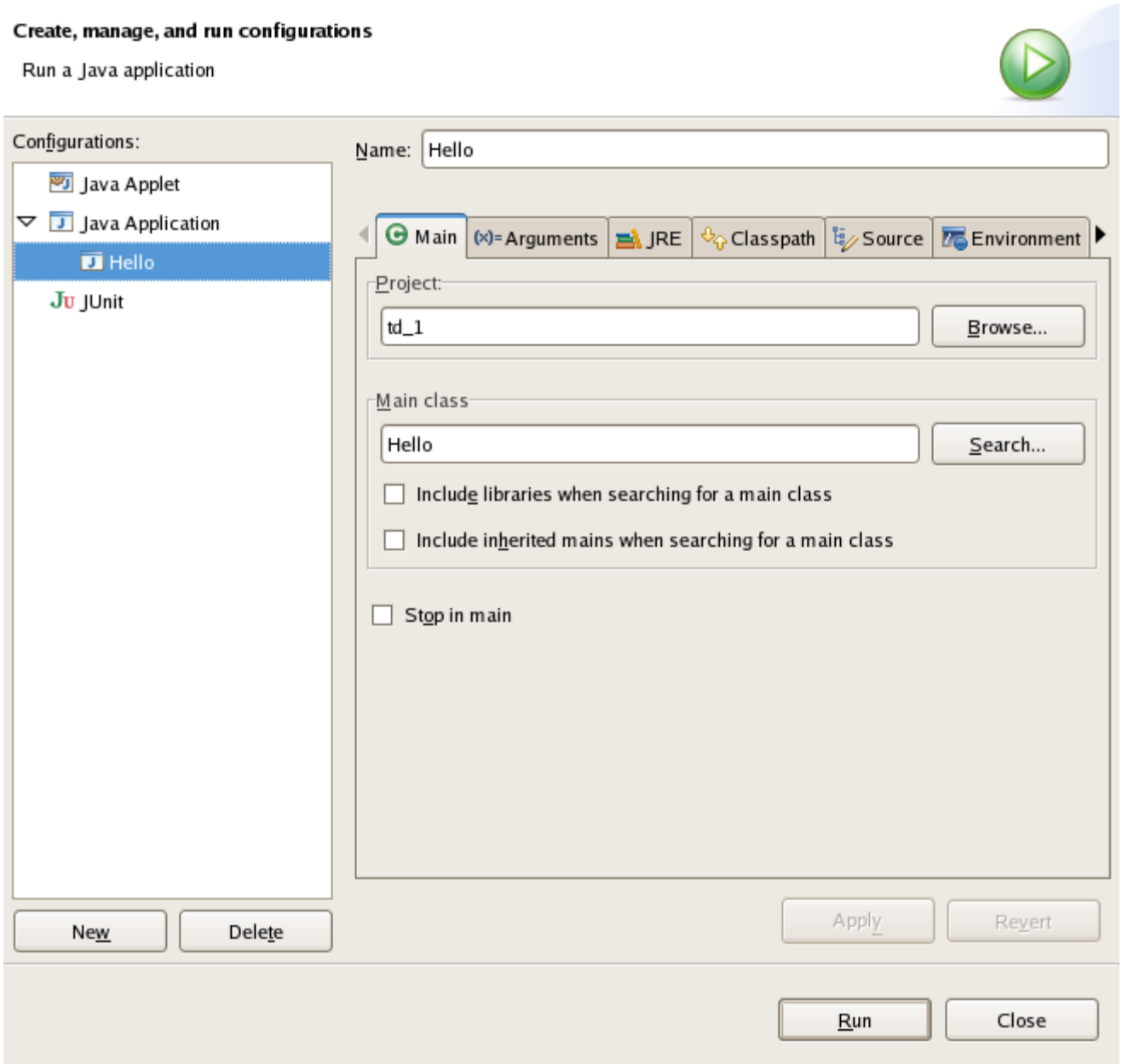
en entrant les deux valeurs, on obtient :



Pour exécuter le programme avec des arguments, on effectue un clic droit sur le .java de la classe, puis Run As -> Run... Si la programme a déjà été exécuté dans eclipse, le nom de la classe apparaît dans la liste à gauche, et il faut cliquer sur ce nom. Sinon, on clique sur "Java application" puis sur "New" :

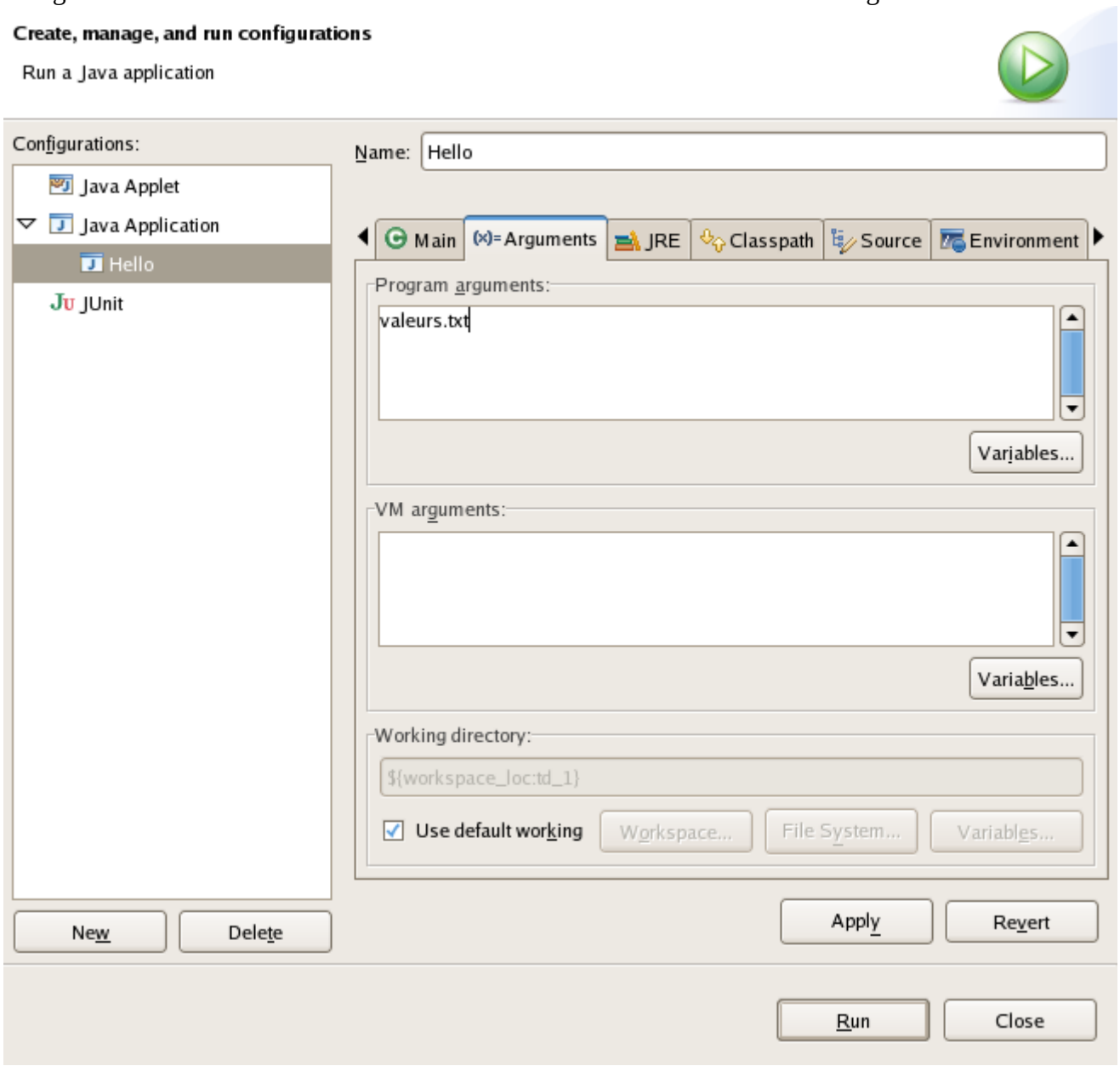


On obtient alors la fenêtre suivante :



En choisissant l'onglet arguments, on peut ajouter des arguments dans la boîte intitulée

"Program arguments" :



Par défaut, le répertoire de travail dans lequel le programme cherche les fichiers est le répertoire du projet. On peut le changer dans la boîte "Working directory" en décochant "Use default working directory", et en cliquant soit sur "Workspace" pour choisir un répertoire du workspace, soit "File System" pour choisir un autre répertoire.

Pendant l'exécution, il peut survenir une exception qui apparaît en rouge sous l'onglet console en bas de la fenêtre principale. Il est possible en cliquant sur l'une des lignes du *stack trace* d'accéder à l'endroit où a été levée l'exception. Par exemple, dans la capture suivante, en cliquant sur la ligne "at Hello.main(Hello.java:5)", eclipse saute à la ligne 5

du fichier Hello.java :