TP1 : Premier programme “HelloWorld” avec Eclipse et en ligne de commande

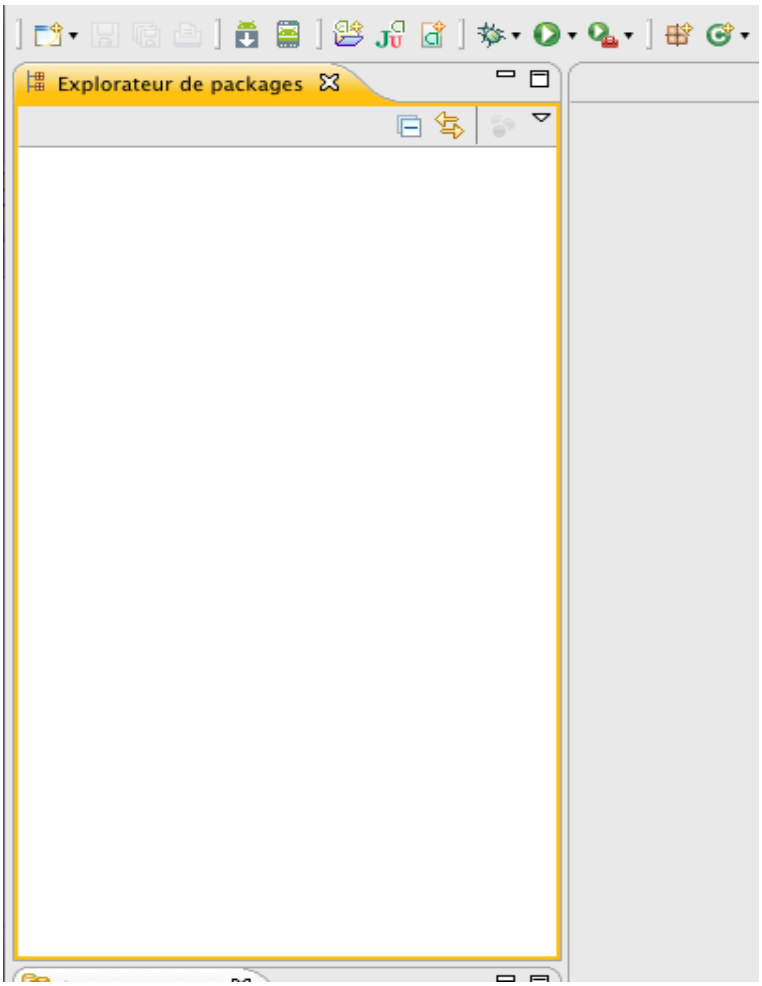
Au cours de ce TP, nous allons introduire l’environnement de développement intégré (IDE pour Integrated Development Environment) nommé **Eclipse**. Nous verrons comment créer et exécuter un programme Java simple de type “Hello World” avec Eclipse. Ensuite, nous verrons comment se servir du terminal pour faire la même chose afin de comprendre comment on peut compiler et exécuter un programme en Java sans l’aide d’un IDE.

Premier contact avec Eclipse

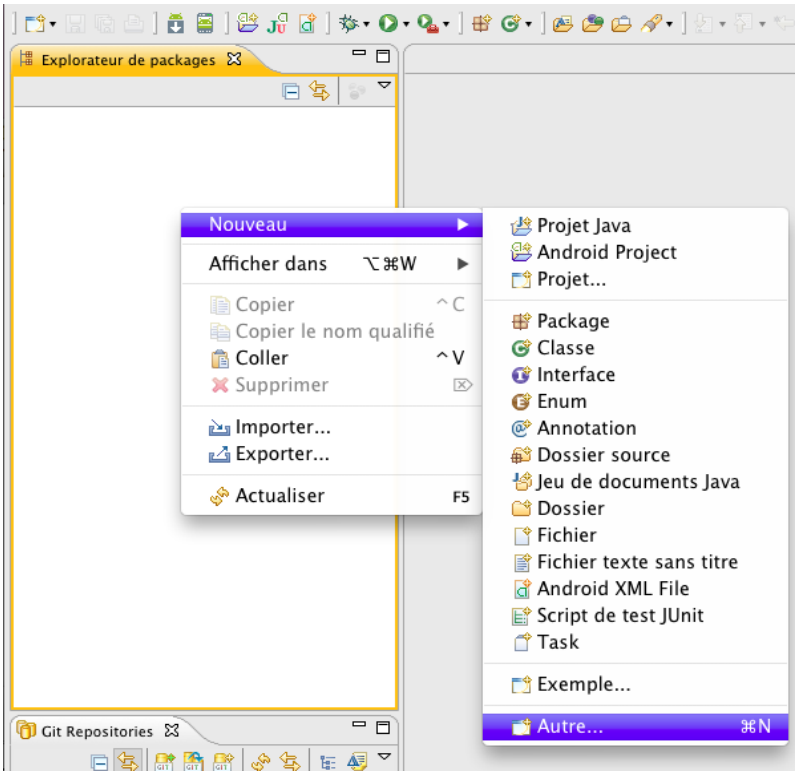
Lancez Eclipse à partir des menus de votre environnement Linux ou Windows.

Explorateur et Projet

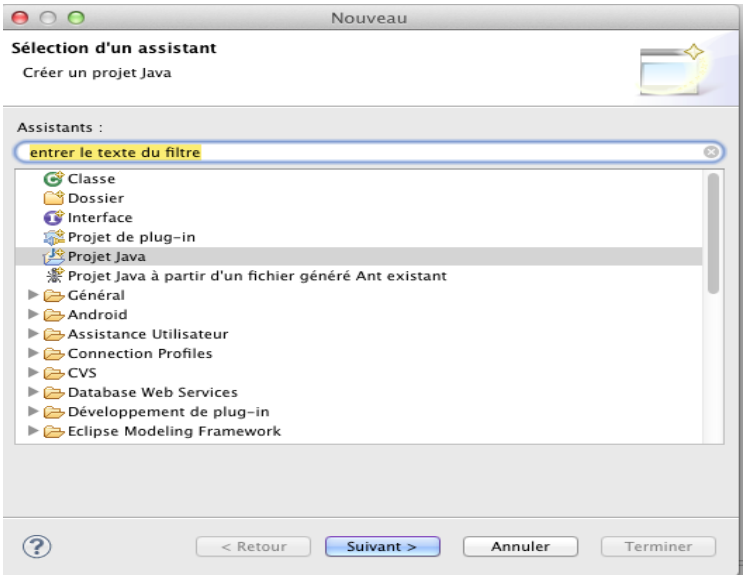
Dans la fenêtre principale d’Eclipse qui s’affiche, identifiez un panneau nommé *Explorateur de packages* tel que représenté ci-dessous :



Faire un clic droit à l'intérieur de l'Explorateur de packages. Dans le menu qui apparaît (voir ci-dessous) emmenez la souris sur *Nouveau* puis choisir *Autre*… lorsque le second menu apparaît.

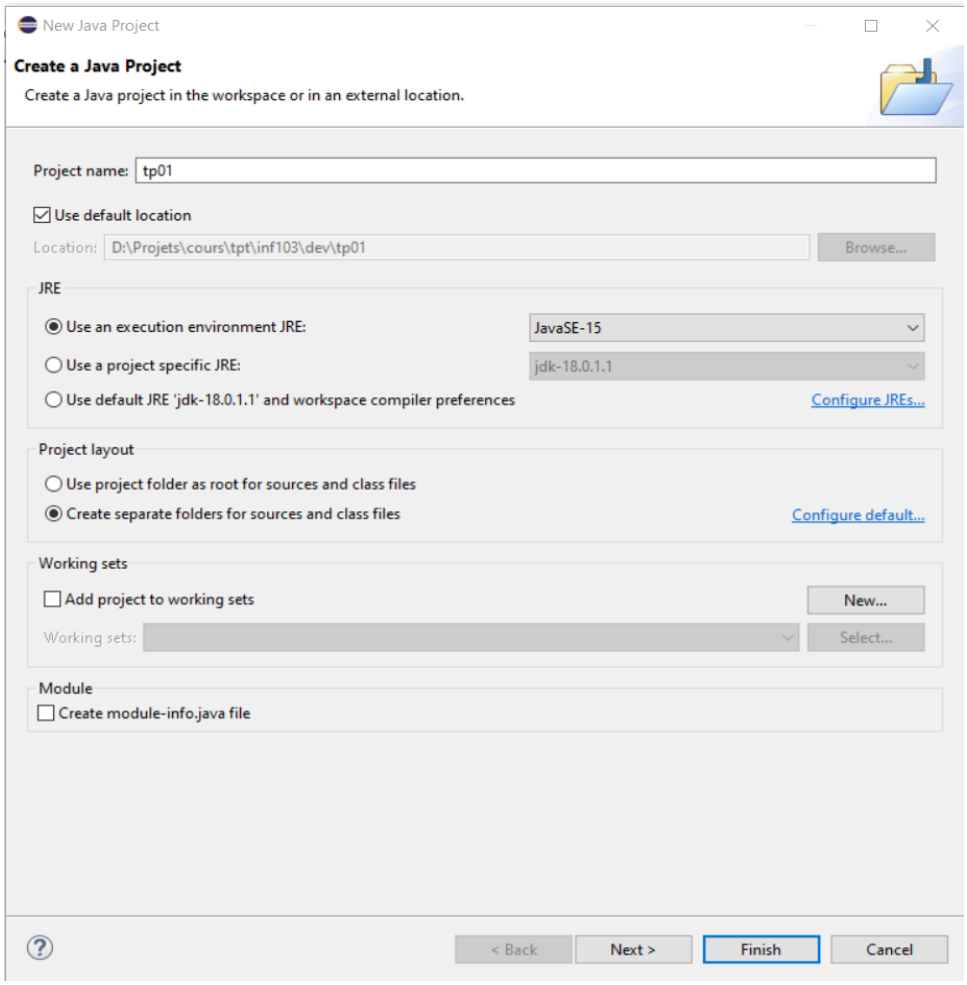


Une fenêtre apparaît (voir ci-dessous) qui vous demande de sélectionner un type de projet. En effet, Eclipse est polyvalent et permet l'édition de différents types de projets dans une multitude de langages. Cliquez sur *Projet Java* puis sur le bouton *Suivant*.

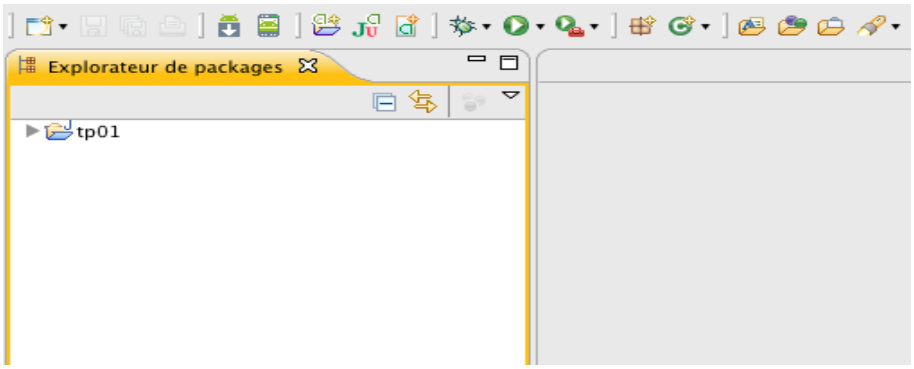


Une nouvelle fenêtre apparaît (voir ci-dessous) qui vous demande d'entrer le nom du projet. Ecrire *tp01* dans le champ *Nom du projet*.

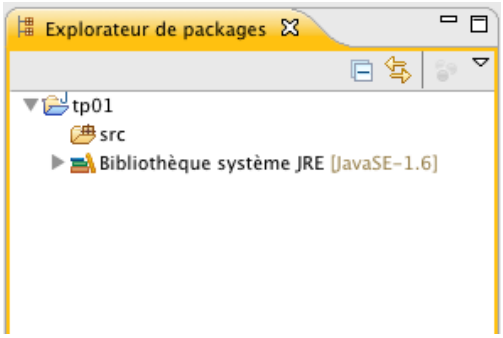
Vérifiez également que l'option *create module-info.java file* au bas de la fenêtre n'est pas sélectionnée (décochez-la si besoin). La notion de module, et que nous n'utiliserons pas pour ce cours, sers à améliorer la modularité des applications Java, en particulier celles de très grande taille. Ne pas modifier les autres options. Cliquer sur le bouton *Terminer (Finish)*.



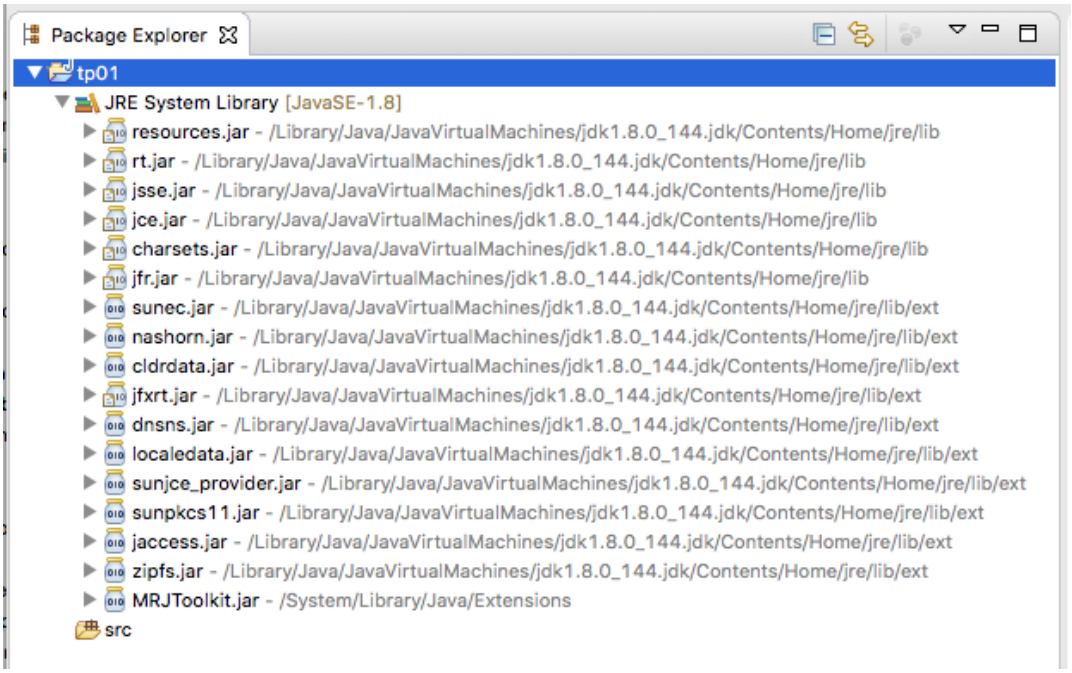
Le projet créé (*tp01*) apparaît dans le panneau *Explorateur de packages* (voir ci-dessous).



Cliquez sur le petit triangle (▶) à gauche du nom du projet. Le triangle se tourne vers le bas (▼) et le contenu du projet est dévoilé (voir ci-dessous).



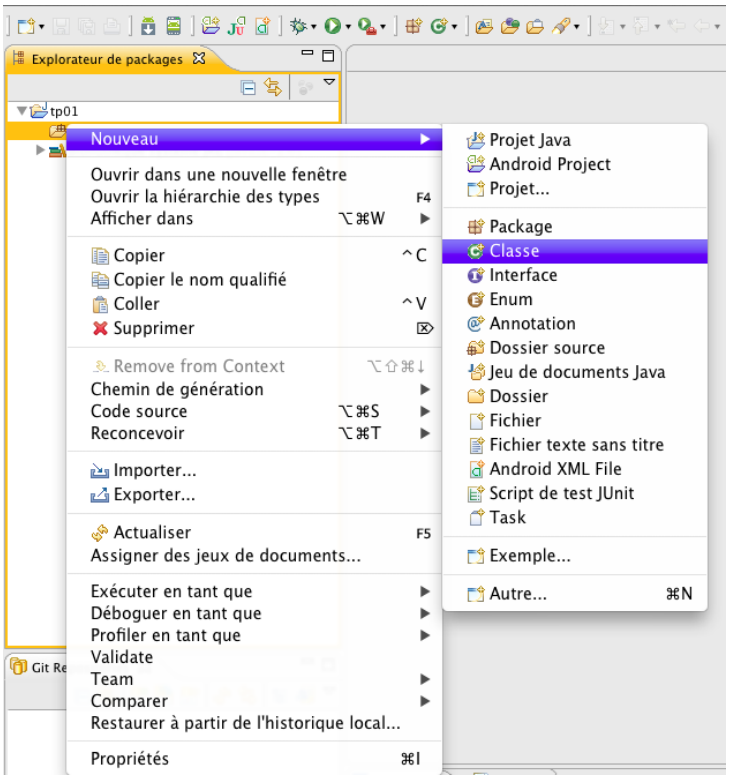
On voit apparaître le dossier *src* (abréviation de source) destiné à recevoir le code source Java de vos programmes ainsi que le dossier Bibliothèque système JRE qui contient l'ensemble des bibliothèques que Java met à votre disposition. En cliquant sur le triangle (▶) à gauche du dossier bibliothèque système JRE, on voit apparaître la liste de ces bibliothèques (voir ci-dessous).



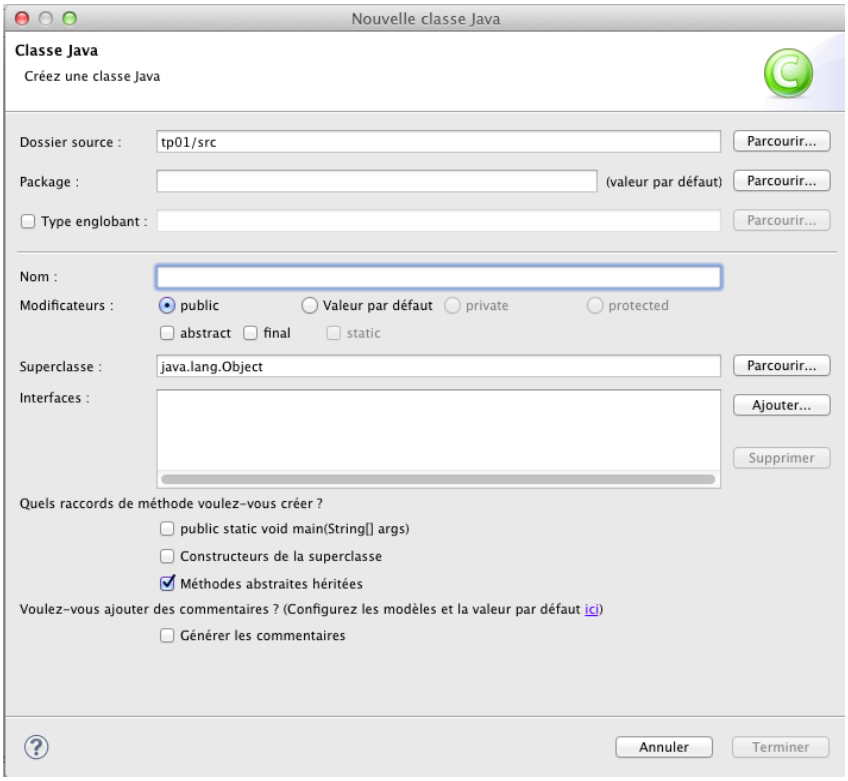
Ces bibliothèques sont des ensembles de programmes, les classes prédéfinies, que Java met à votre disposition. Nous en étudierons quelques-unes en cours. En cliquant à nouveau sur le triangle (▼) à gauche du dossier Bibliothèque système JRE, on referme ce dossier.

Classe

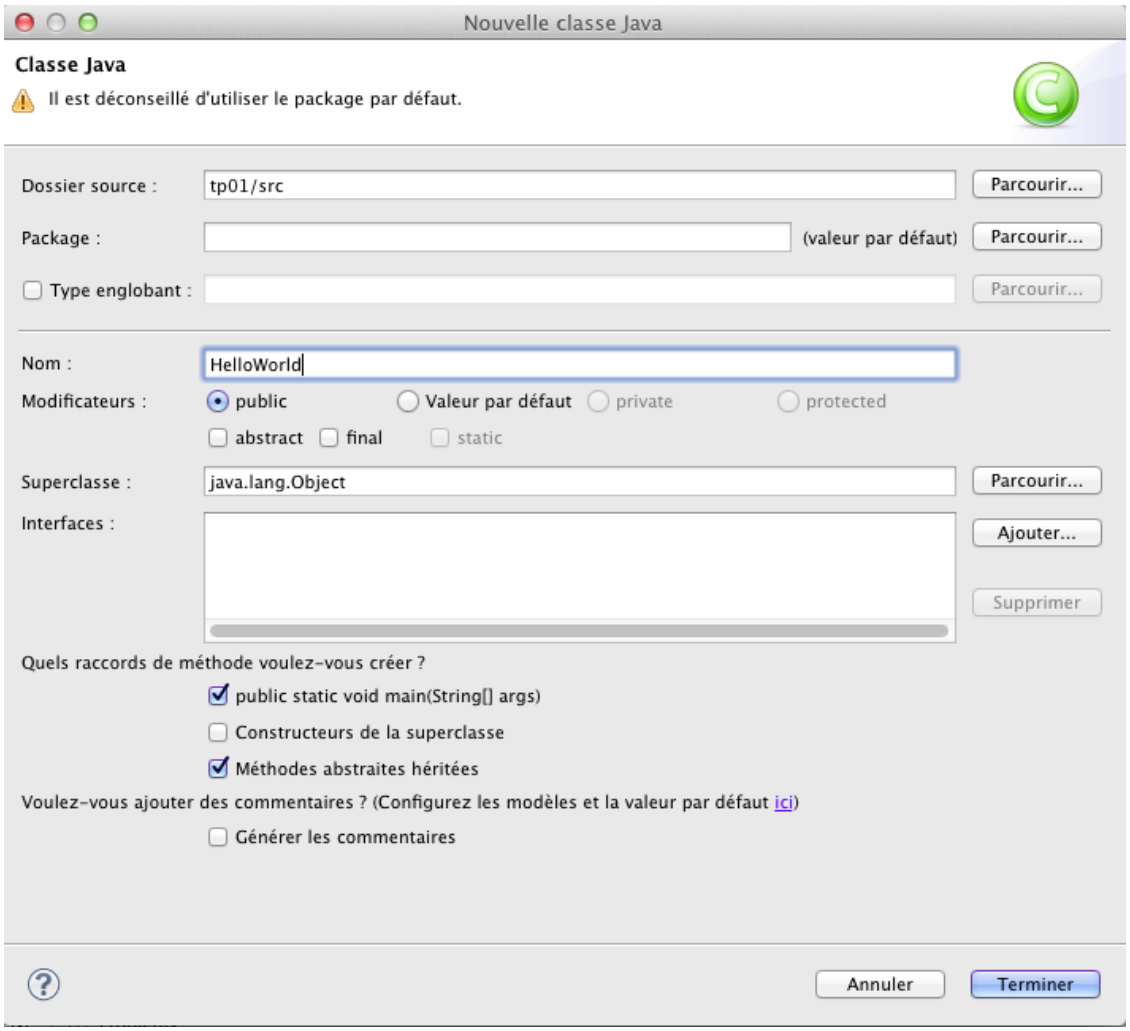
Nous allons créer une classe. Pour cela, faire un clic droit sur le dossier *src*. Un menu apparaît. Amener la souris sur *Nouveau*. Un sous-menu apparaît, amener la souris sur *Classe* et cliquer.



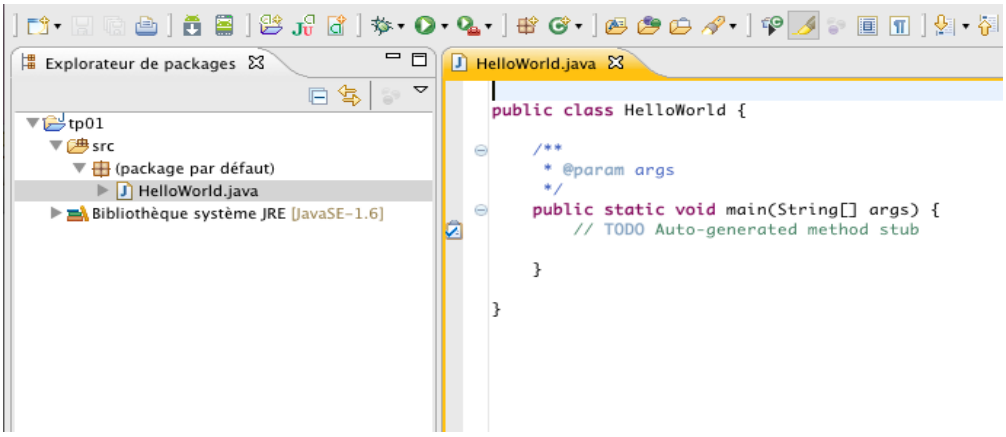
Une fenêtre telle que celle de la capture d'écran ci-dessous apparaît.



Dans le champ *Nom*, écrire *HelloWorld*. Sélectionner la boîte (checkbox) devant *public static void main(String[] args)*. Cliquer sur le bouton *Terminer*.



Dans la vue *Explorateur de packages* située à gauche dans la fenêtre principale, on voit apparaître le fichier de la classe nommé *HelloWorld.java*. Dans le panneau central à droite de l'explorateur de package, on voit apparaître le code Java de cette classe. Ce panneau sert également à éditer le code de la classe.



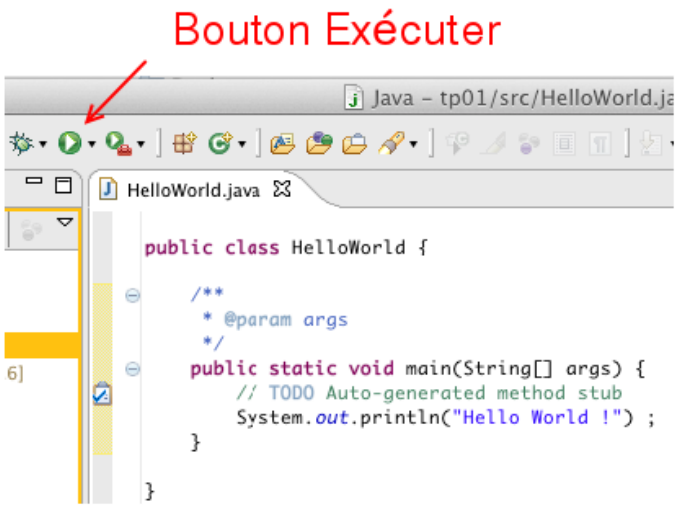
Dans cette classe, tout ce qui se situe entre les chaînes de caractères **/**** et ***/** et tout ce qui se situe entre la chaîne de caractère **//** et la fin de la ligne sont des zones dédiées à l'écriture de commentaires. Ces zones ont été générées par Eclipse. Notez également les mots clés public devant les déclarations de la classe et de la méthode main. Nous verrons la signification de ces mots dans un cour ultérieur.

Une méthode telle que la méthode main est celle qui sera exécutée au lancement du programme (nous expliquerons cette méthode plus en détails dans le cours). Cette méthode, générée par Eclipse, est vide. Dans son corps, écrire l'instruction suivante :

```
System.out.println("Hello World !");
```

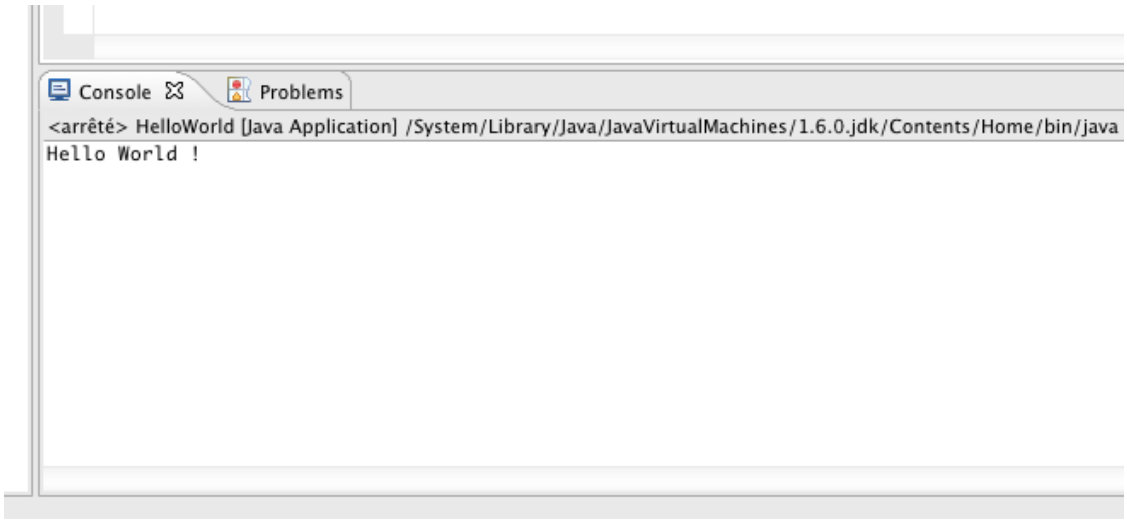
Vous obtenez :

Sauvegarder le fichier (par défaut, Eclipse va compiler la classe dès que son fichier sera sauvegardé). Cette instruction a pour fonction d'écrire la chaîne de caractères **Hello World** ! dans ce que l'on appelle la Console. Pour exécuter ce programme, cliquer sur le bouton rond vert avec un triangle blanc à l'intérieur. Ce bouton se situe dans la barre de menu.



Eclipse va alors exécuter la classe, ce qui aura pour effet d'exécuter la méthode main de votre classe qui constitue le point d'entrée du programme.

Le panneau Console apparaît en bas de votre fenêtre Eclipse. On y voir le résultat de l'instruction d'écriture. L'exécution est aussitôt terminée.



Vous venez d'exécuter votre premier programme Java !

Compiler et exécuter un programme Java en ligne de commande (sans IDE)

Comme vous avez pu le constater, créer une simple classe et exécuter son programme est très facile avec un IDE comme Eclipse, ce qui permet de développer une application complexe avec une meilleure productivité. Mais comment ferions-nous sans IDE ?

L'exécution d'un programme Java se fait par interprétation du code binaire, issu de la compilation, grâce à une JVM (Java Virtual Machine). L'installation de cette machine se réalise en déployant un environnement JRE (Java Runtime Environment), qui fournit différents exécutables pour compiler et exécuter un programme Java, ainsi que différents utilitaires tels que `javadoc` pour la génération automatique de la documentation sous forme de pages html.

L'utilisation de ces différents exécutables se fait via un terminal, outil que nous allons introduire dès maintenant.

Introduction au terminal

Un terminal est une fenêtre permettant d'exécuter des lignes de commande. Certains d'entre vous connaissent peut-être déjà la ligne de commande. Si vous ne l'avez pas encore utilisée, voici une petite introduction.

Les trois plateformes de système d'exploitation les plus connues proposent chacune une façon de lancer une interaction avec le système en ligne de commande ; le terminal.

Linux

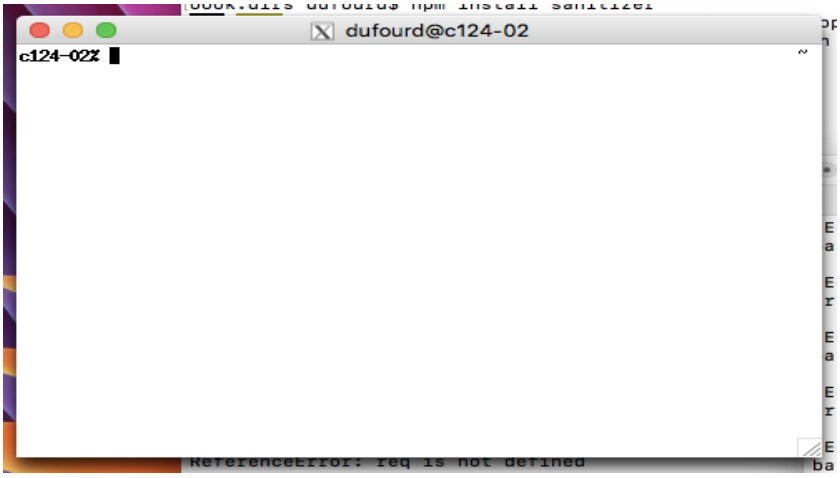
Chaque distribution Linux est différente, mais vous pouvez typiquement lancer un terminal avec l'application *Terminal*.

Windows

Si vous avez Windows, alors lancez l'application Windows *Powershell* qui inclut les commandes introduites dans ce paragraphe. Pour trouver *Powershell* sur Windows, effectuez une recherche dans la barre de recherche en bas à gauche.

Mac
Le terminal du Mac se trouve dans */Applications/Utilities*.

Pour tous les systèmes
Ouvrez un terminal.



Vous voyez une ligne avec le nom de la machine suivi du caractère *%* et un curseur. Dans Powershell, il y a le dossier courant et le caractère > avant le curseur. Cette ligne est l'invite de commande.

Différentes commandes existent sous Linux pour effectuer différentes opérations telles que :

- *ls* — pour afficher (**ls**ter) des fichiers and dossiers.
- *pwd* — pour afficher le dossier courant (**p**rint **w**orking **d**irectory)
- *cd* — pour **c**hanger de **d**ossier

Certaines commandes ont besoin d'arguments. Par exemple, pour changer de dossier, il faut savoir où aller. On rajoute donc des arguments à la commande :

- *cd workspace* — pour aller le dossier *workspace* (dans le dossier courant, qu'on peut voir avec *pwd*)
- *cd ..* — pour aller dans le dossier parent, celui qui contient le dossier courant.
  - en Linux il y a deux dossiers magiques :
    - . — Le dossier courant
    - .. — Le dossier parent

Essayons quelques commandes :

Tapez *pwd*↵

La réponse du système affichée dans le terminal sera votre répertoire courant.

Dans ces TPs, nous allons utiliser le symbole → pour indiquer la réponse du système, ainsi :

Tapez *pwd* ↵

→ */<votre répertoire>*.

Tapez *ls*↵

→ Vous voyez la liste des fichiers et dossiers dans votre dossier personnel. On y trouvera un dossier nommé *workspace* qui est le dossier pour votre espace de travail Java, crée par défaut par Eclipse. Si vous l'avez créé ailleurs que dans votre dossier perso, il faudra y naviguer avec la commande *cd*.

Tapez *cd workspace*↵ pour aller dans le dossier *workspace*.

Tapez *ls*↵ pour afficher les fichiers du dossier courant.

→ Vous voyez au moins un dossier nommé *tp01*.

Tapez *cd tp01*↵

Puis *ls*↵

→ Vous voyez au moins un dossier *bin* et un dossier *src*.

Tapez *ls src*↵

→ Vous voyez un fichier nommé *HelloWorld.java* ainsi que tous les autres fichiers que vous avez potentiellement créés pendant le TP.

Compiler votre programme Hello World !

Tel que mentionné précédemment, pour compiler et exécuter programme Java, il faut disposer d'un environnement JRE qui sera spécifique au système d'exploitation donné (Windows, Linux, Mac). La machine virtuelle fournie par cet environnement interprétera le code binaire compilé afin de le traduire en instructions spécifiques au système. Il existe différents fournisseurs d'environnements JRE, les plus connus étant Oracle et le projet OpenJDK.

Différents exécutables servant à compiler et exécuter des programmes Java sont stockés dans un sous-répertoire nommé *Java/<nom de la version>/bin* créé dans le répertoire des programmes du système d'exploitation lors de l'installation de la machine virtuelle. Pour connaître la version de java qui est installée, tapez la commande suivante :

java -version↵

Cette version correspond à la version la plus récente du langage Java supportée par la machine virtuelle. Pour cet exercice, veuillez-vous assurer que la version de Java est supérieure à 1.8.

Pour une version donnée de Java, il existe deux types d'installation de la machine virtuelle ayant un contenu différent : l'installation JRE (Java Runtime Environment) telle que mentionnée précédemment et l'installation JDK (Java Development Kit). L'installation JDK contient, en plus de tout ce que contient une JRE, le code source de Java ainsi que l'exécutable *javac* servant à la compilation.

Si vous travaillez avec votre propre ordinateur (et non pas un ordinateur de l'école), et qu'un message stipulant que la commande `javac` n'est pas trouvée lors de l'exercice suivant, installez un JDK sur votre ordinateur. Vous pouvez utiliser le JDK fourni par Oracle ou celui du projet OpenJDK. Note : certaines versions récentes d'Eclipse contiennent leur propre JDK, qu'il serait également possible d'utiliser. Le cas échéant, celui-ci se trouvera dans le répertoire d'installation d'Eclipse nommé :

../eclipse/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.<version>

Si ce n'est déjà fait, dans un terminal, naviguez dans le répertoire src du projet Eclipse de votre programme *Hello World !* en utilisant la commande *cd*.

Ensuite tapez la commande *javac HellowWorld.java*↵

Examinez le contenu du répertoire avec la commande *ls*. Vous verrez que l'exécution du programme *javac* a produit un fichier nommé *HellowWorld.class* à partir du fichier *HellowWorld.java* passé en paramètre. Ce fichier contient le code binaire généré par la compilation du code Java.

Il est également possible de spécifier un répertoire (différent de *src*) où stocker les fichiers compilés. Par exemple, Eclipse va plutôt stocker les fichiers compilés dans un répertoire nommé `bin` (pour binary) au même niveau que le répertoire *src* dans le répertoire du projet.

Exécuter votre programme Hello World !

Ce fichier contenant le code compilé peut maintenant être exécuté par la machine virtuelle. Pour ce faire, il faut utiliser l'exécutable nommé *java* en lui passant en paramètre le nom de la classe :

Tapez la commande *java HellowWorld*↵

Vous devriez voir s'afficher la chaîne de caractère *Hello World !* dans le terminal.

Il est également possible de réaliser ces deux opérations en une seule commande avec l'exécutable *java*. Supprimez d'abord le fichier complié avec la commande *rm* :

Tapez la commande *rm HellowWorld.class*↵

Utilisez la commande *ls* pour vérifier que le fichier a bien été supprimé.

Tapez la commande *java HellowWorld.java*↵

Vous verrez s'afficher dans le terminal la chaîne de caractère *Hello World !*. Notez que le fichier compilé n'a pas été enregistré sur disque ; il a simplement été créé puis exécuté dans la foulée sans sauvegarde sur disque. Ainsi, il est préférable de compiler d'abord un programme en utilisant *javac* car le programme pourra ensuite être exécuté autant de fois que l'on souhaite sans avoir à le recompiler.

Note : Ces commandes fonctionnent pour une classe qui n'est pas contenue dans un *package*[1] (i.e., son fichier *.java* se trouve directement dans le répertoire *src*). Si vous avez déclaré votre classe dans un package (par exemple un package nommé *test*), son fichier *.java* sera alors contenu dans un répertoire nommé *test* contenu dans le répertoire *src*. Pour exécuter cette classe, il faudra alors se positionner dans le répertoire *src* et préfixer le nom de la classe par son nom de package tel qu'illustré dans la commande suivante :

java test.HellowWorld↵

Si vous tentez d'exécuter la classe dans le sous-répertoire *test*, vous verrez apparaître un message d'erreur tel que *Error: Could not find or load main class HelloWorld*. Il faut donc bien vérifier que vous lancez les commandes à partir du répertoire *src*.

Ces commandes simples nous montrent comment Eclipse (ou tout autre IDE) compile et exécute un programme Java pour nous. Sous Eclipse, par défaut (il est possible de changer ce comportement), la compilation est exécutée à chaque fois qu'un fichier Java est sauvegardé. Lorsqu'un projet contient plusieurs fichiers Java, Eclipse ne recompilera que les fichiers modifiés et ceux qui en dépendent, ce qui s'appelle la *compilation incrémentale*, permettant de minimiser le temps de compilation.

---

[1] Nous verrons la notion de package dans un cours ultérieur.