

## TP 1 : Premier programme « Hello World ! »

Dominique Blouin, Télécom Paris, Institut Polytechnique de Paris

dominique.blouin@telecom-paris.fr

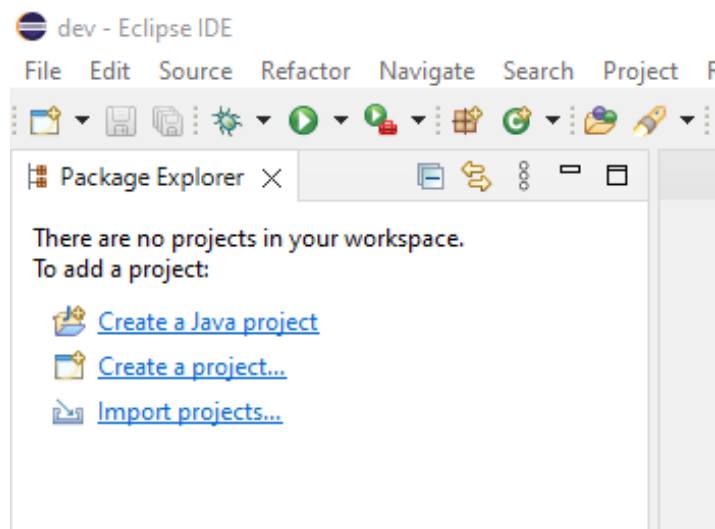
Au cours de ce TP, nous allons introduire l'environnement de développement intégré (IDE pour Integrated Development Environment) nommé **Eclipse**. Nous verrons comment créer et exécuter un programme Java simple de type "Hello World" avec Eclipse. Ensuite, nous verrons comment se servir du terminal pour 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.

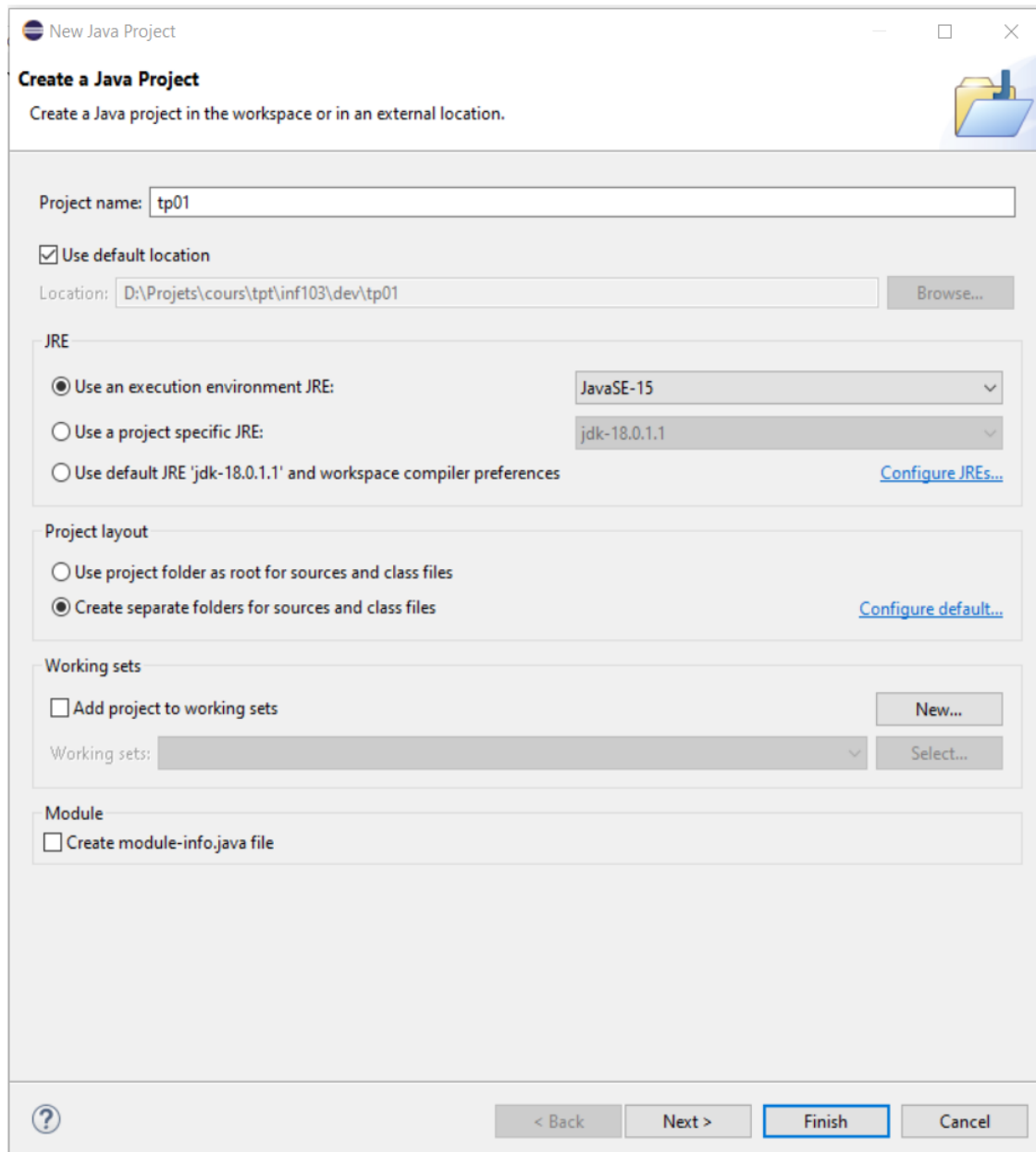
### Explorateur et Projet

Dans la fenêtre principale d'Eclipse qui s'affiche, identifiez un panneau nommé *Package Explorer* tel que représenté ci-dessous :

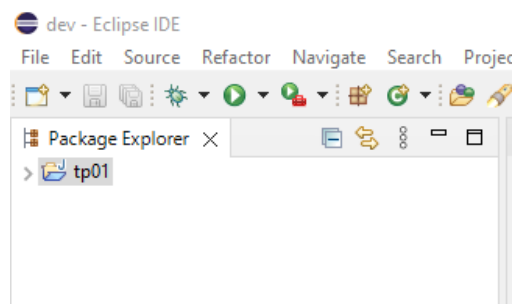


A l'intérieur de l'explorateur de packages, cliquez sur *Create a Java Project*. Une fenêtre apparaît (voir ci-dessous) qui vous demande d'entrer le nom du projet. Ecrire *tp01* dans le champ *Nom du projet*.

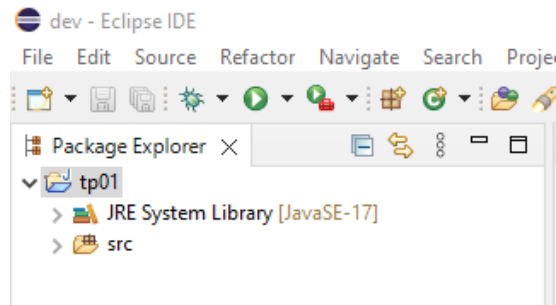
Vérifiez également que l'option *create module-info.java file* au bas de la fenêtre n'est **pas** sélectionnée (décochez-la si besoin). La notion de module, que nous n'utiliserons pas pour ce cours, sert à améliorer la modularité des applications Java, en particulier celles de très grande taille. Ne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



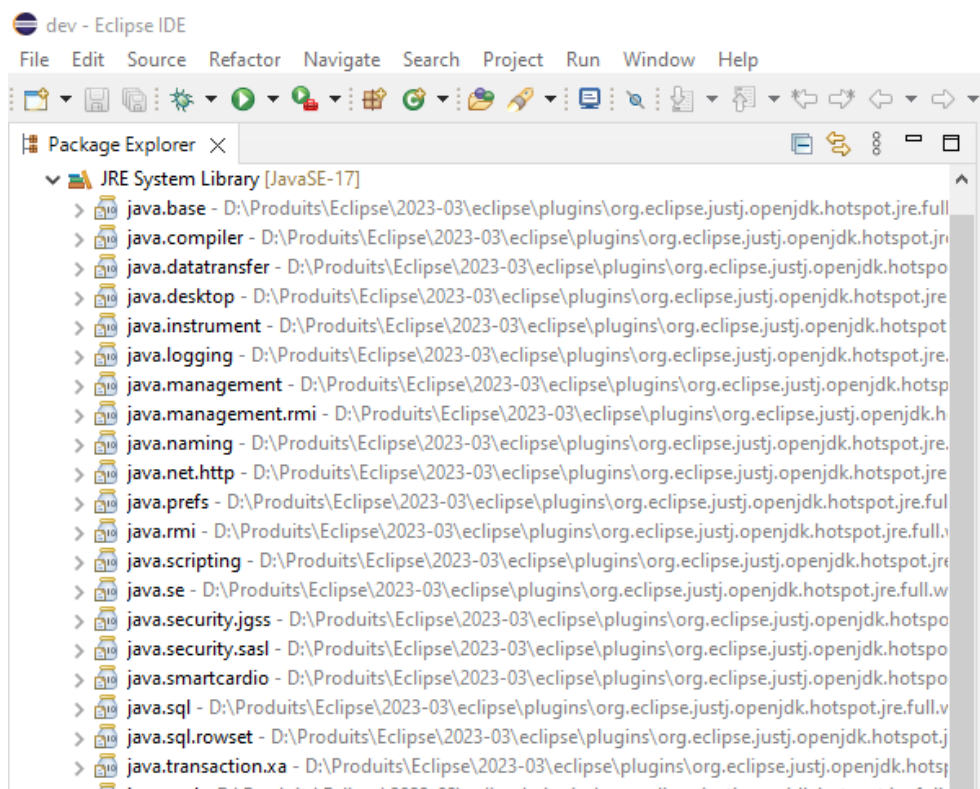
Le projet créé (*tp01*) apparaît dans le panneau *Package Explorer* (voir ci-dessous).



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



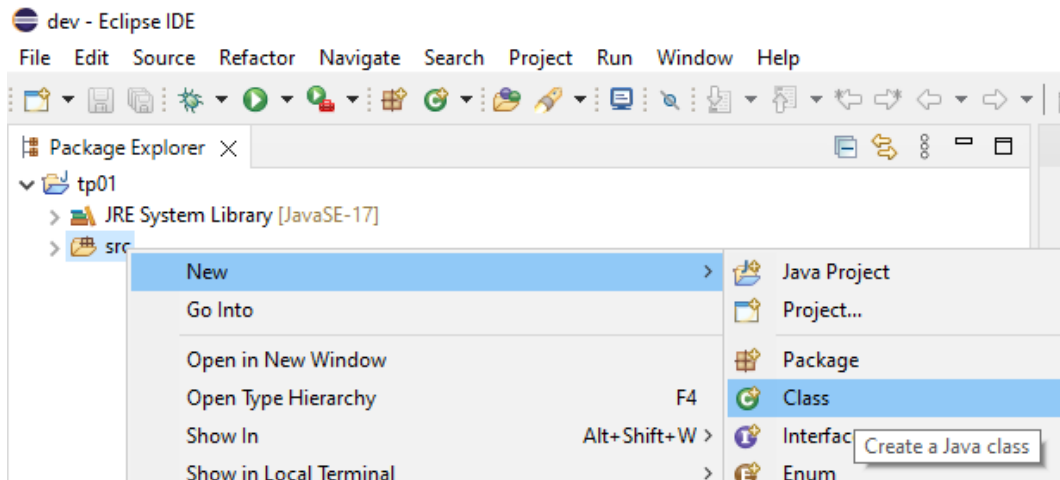
On voit apparaître le dossier *src* (abréviation de source) destiné à recevoir le code source Java de vos programmes ainsi que le dossier *JRE System Library* qui contient l'ensemble des bibliothèques que Java met à votre disposition. En cliquant sur le triangle (>) à gauche du dossier *JRE System Library*,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



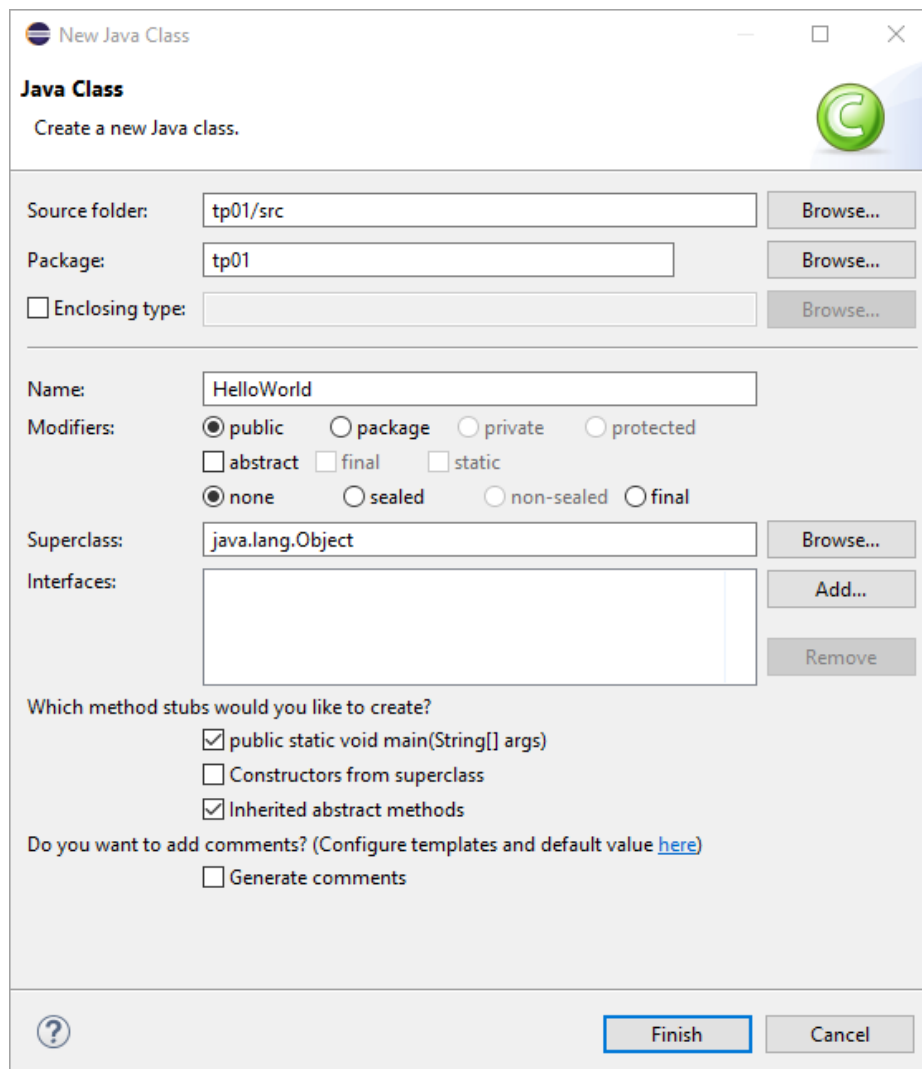
Ces bibliothèques sont des ensembles de classes prédéfinies mises à disposition par Java. Nous en étudierons quelques-unes en cours. En cliquant à nouveau sur le triangle (v) à gauche du dossier *JRE System Library*, on referme ce dossier.

## Classe

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

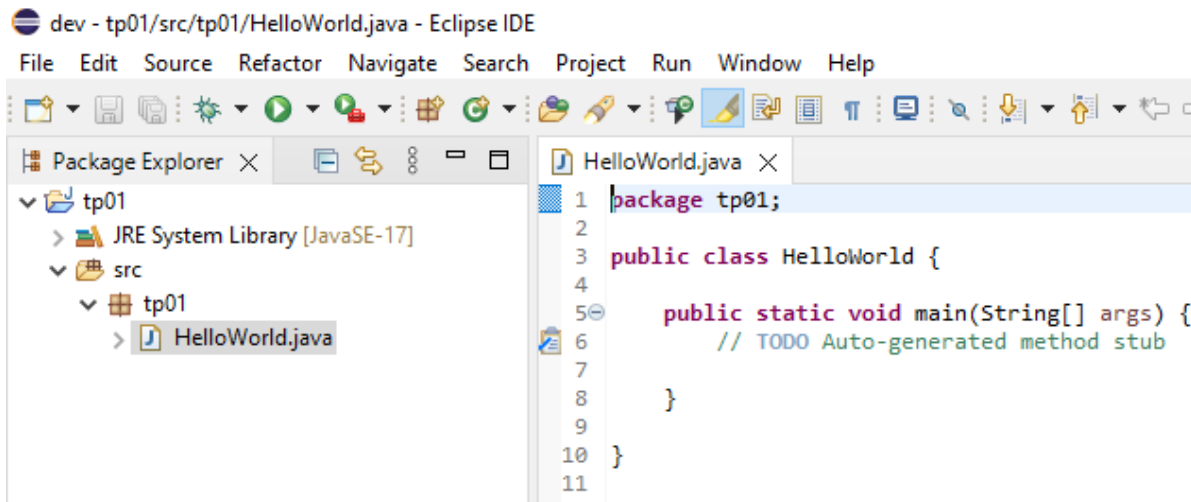


Une fenêtre telle que celle de la capture d'écran ci-dessous apparaît.



Dans le champ *Name*, écrire *HelloWorld* puis sélectionner la boîte (checkbox) devant *public static void main(String[] args)*. Cliquer sur le bouton *Finish*.

Dans la vue *Package Explorer*, 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 clés dans un cours ultérieur.

Une méthode telle que la méthode `main` est celle qui sera exécutée au lancement du programme. Cette méthode, générée par Eclipse, est vide. Dans son corps, écrire l'instruction suivante :

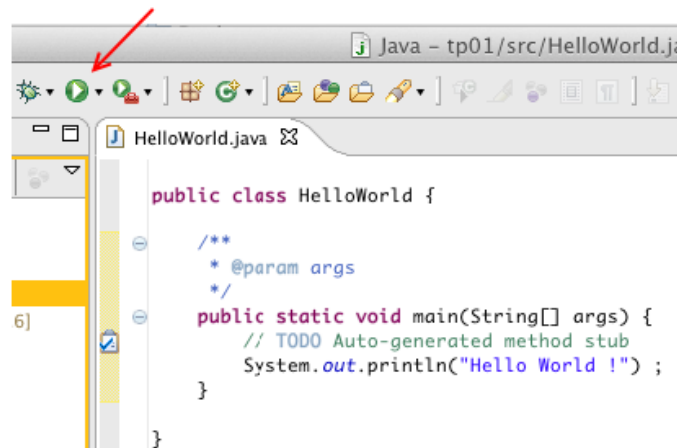
```
System.out.println("Hello World !");
```

Vous obtenez :

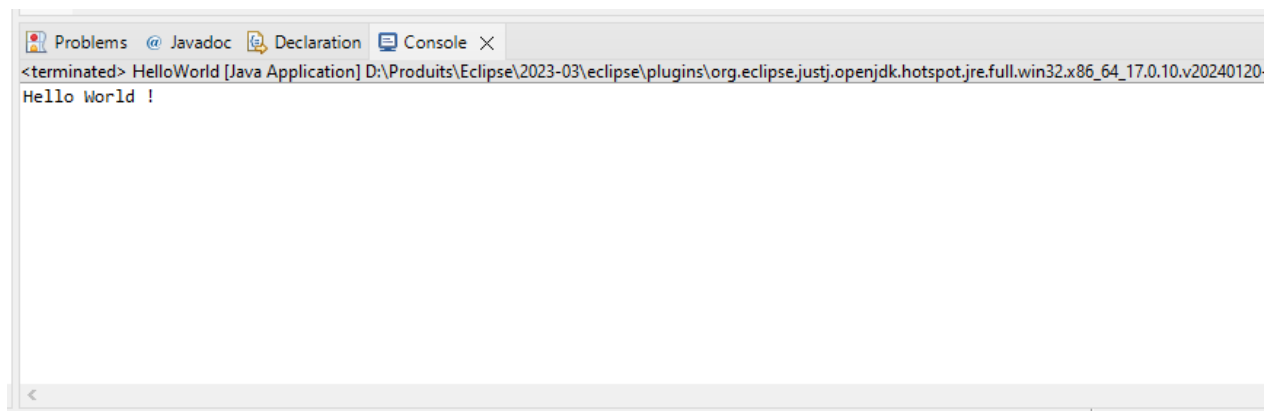


Sauvegarder le fichier. Par défaut, Eclipse va compiler la classe dès que son fichier sera sauvegardé. Cette instruction a pour fonction d'écrire la chaîne de caractères *Hello World !* dans ce que l'on appelle la *Console*. Pour exécuter ce programme, cliquer sur le bouton rond vert avec un triangle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

## Bouton Exécuter



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* apparaîtra en bas de fenêtre du code. On y voit le résultat de l'instruction d'écriture. L'exécution du programme est aussitôt terminée.



Vous venez d'exécuter votre premier programme !

### 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, ce qui permet de développer une application avec une meilleure productivité. Mais comment ferions-nous sans IDE ?

Le code Java est d'abord compilé en code binaire. L'exécution d'un programme Java se fait ensuite par *interprétation* de ce code binaire 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 vous avez sans doute déjà utilisé.

## Compiler votre programme *Hello World* ! en ligne de commande

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, ainsi que le projet OpenJDK.

Les exécutables de la JRE servant à compiler et exécuter des programmes Java sont stockés dans un sous-répertoire nommé *Java/<nom de la version>/bin* qui sera 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, dans un terminal, 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 : l'installation JRE 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.

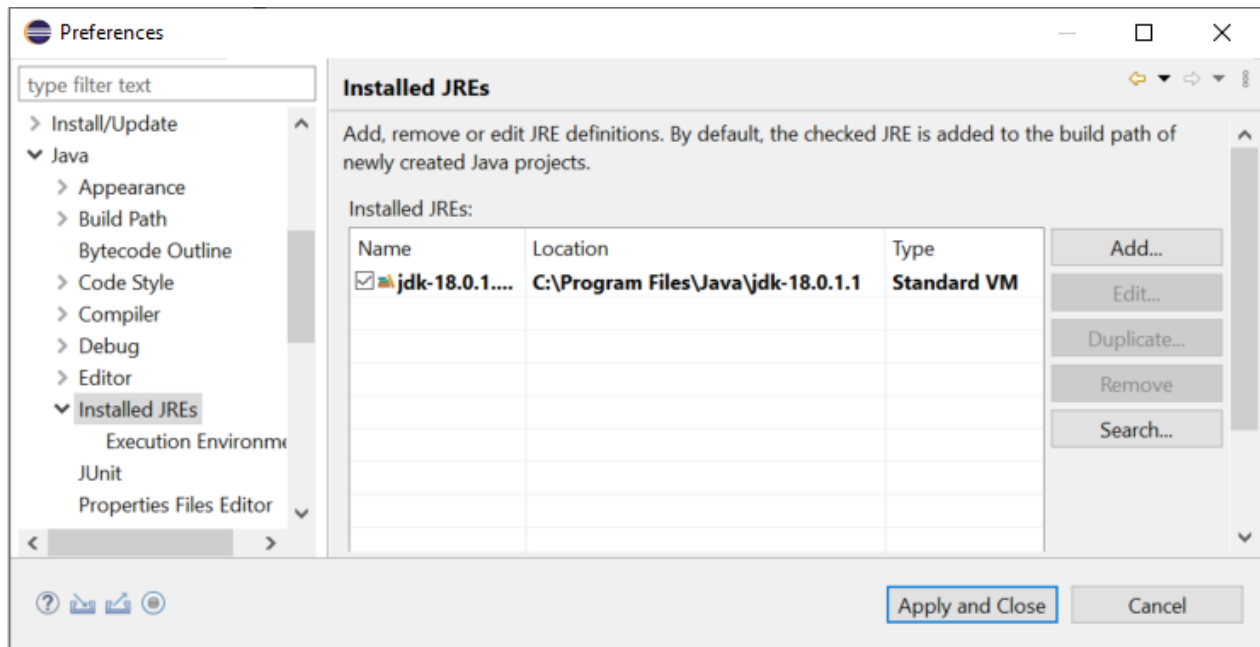
### Installation de la machine virtuelle

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](#).

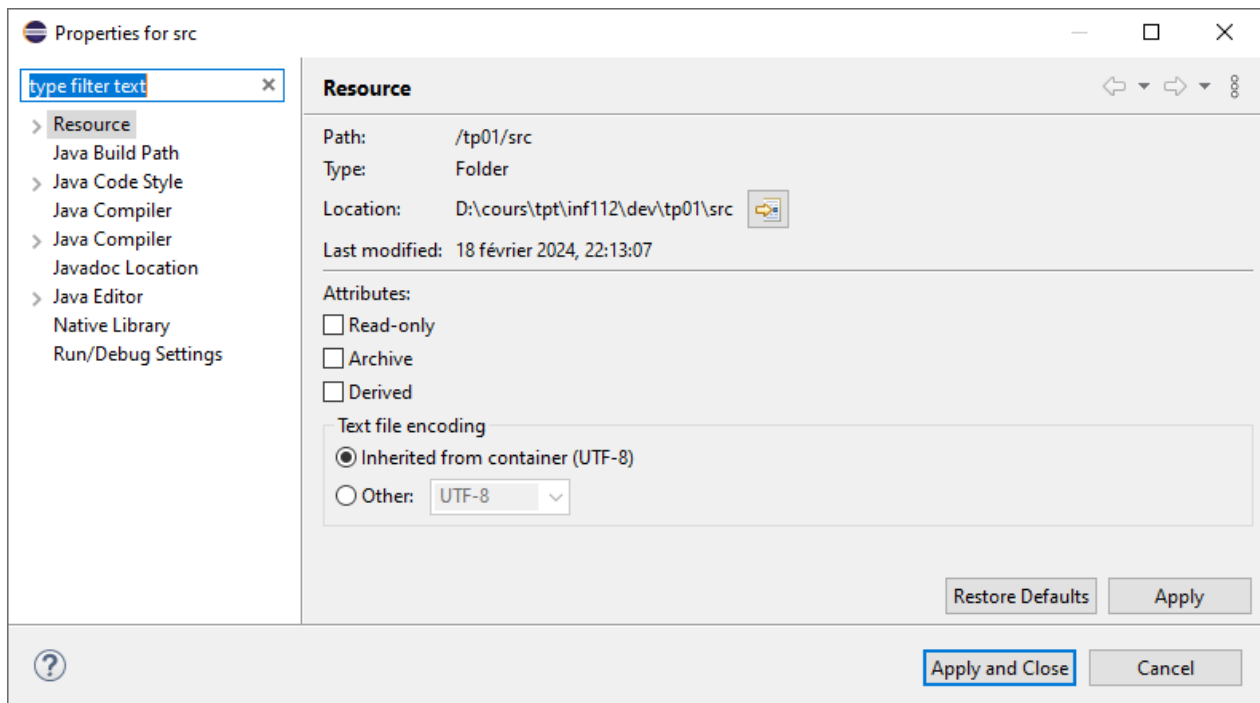
Note : les versions récentes d'Eclipse contiennent leur propre JDK, qu'il serait également possible d'utiliser. Le cas échéant, celui-ci se trouvera dans le répertoire d'installation d'Eclipse nommé :

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

Mais cela peut dépendre de la façon dont Eclipse a été installé (par exemple via l'installateur ou en via une archive) ou même du système d'exploitation. Pour connaître le répertoire de la JVM que Eclipse utilise, il suffit de cliquer sur le menu *Window>>Preferences*. Dans la boîte de dialogue qui s'affiche, sélectionner la branche *Java/Installed JREs* tel qu'illustré dans la capture d'écran suivante. La partie droite de la fenêtre indique le répertoire d'installation de la JVM utilisée par Eclipse.



Si ce n'est déjà fait, dans un terminal, naviguez dans le répertoire *src* du projet Eclipse de votre programme « *Hello World !* ». Pour connaître ce répertoire, dans Eclipse, sélectionner le répertoire *src* dans l'explorateur de package puis faites un clic-droit et sélectionner le menu *Properties*. Dans la fenêtre qui s'affiche, sélectionner la branche *Resource*. Le répertoire du dossier *src* est affiché dans le côté droit de la fenêtre tel qu'indiqué par le libellé *Location*. Un bouton vous permet également d'ouvrir un navigateur de fichier positionné directement sur le répertoire.



Ensuite taper la commande `javac tp01/HelloWorld.java`



Examiner le contenu du répertoire *src/tp01* contenant le fichier Java. Vous verrez que l'exécution du programme *javac* a produit un autre fichier nommé *HelloWorld.class* à partir du fichier *HelloWorld.java*. Ce fichier contient le code binaire généré par la compilation du code Java.

Il est possible de spécifier au programme *javac* un répertoire (différent de *src*) où stocker les fichiers compilés. Par exemple, lors de la compilation, Eclipse va stocker les fichiers compilés dans un répertoire différent de *src* nommé *bin* (pour *binary*) et situé au même niveau que *src* dans le répertoire du projet.

### Exécuter votre programme Hello World !

Le fichier *HelloWorld.class* 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.

Toujours dans le répertoire *src*, tapez la commande *java tp01/HelloWorld*<sup>4</sup>

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 compilé *HelloWorld.class* et vérifiez qu'il a bien été supprimé.

Tapez la commande *java tp01/HelloWorld.java*<sup>4</sup>

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* (ce que fait Eclipse), 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 est contenue dans un *package*<sup>1</sup> nommé *tp01* (i.e., son fichier *.java* se trouve directement dans le sous-répertoire *tp01* de *src*). Si vous avez déclaré votre classe dans un autre package (par exemple un package nommé *test*), son fichier *.java* sera alors contenu dans un sous-répertoire nommé *test* de *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 *test* au lieu de *tp01*.

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, la compilation est exécutée à chaque fois qu'un fichier Java est sauvegardé. Ainsi, 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*, ce qui permet de minimiser le temps de compilation.

---

<sup>1</sup> Nous verrons la notion de package dans un cours ultérieur.