

TP1

Exercice – premiers pas sous Eclipse

1. Lancez Eclipse, puis validez votre espace de travail (*workspace*), c'est-à-dire le répertoire dans lequel vos projets Java (`.java`, `.class` et autres ressources) seront enregistrés. Dans le menu *Window*, sélectionnez *Open Perspective* puis *Java*.
2. Dans le menu *File*, sélectionnez *New* puis *Java Project* (ou cliquez sur l'icône raccourcie *New* puis *Java Project* dans la barre d'outils).
3. Dans la fenêtre qui s'ouvre, indiquez un nom de projet (*e.g.*, `TP1`), puis cliquez sur *Finish*. Dans votre espace de travail, un répertoire du nom de votre projet a été créé, contenant notamment deux sous-répertoires : `src` et `bin` dans lesquels seront respectivement stockés les fichiers `.java` et `.class`.
4. Votre nouveau projet apparaît dans l'onglet *Package Explorer* à gauche. Faites un clic-droit sur le nom de votre projet et sélectionnez *New* puis *Class* (ou cliquez sur l'icône raccourcie *New Java Class* dans la barre d'outils).
5. Donnez un nom à votre classe (*e.g.*, `HelloWorld`), puis cliquez sur *Finish*. La classe apparaît maintenant dans le répertoire `src`.
6. Ajoutez une méthode principale (plutôt que de taper intégralement l'entête `public static void main(String[] args)`, vous pouvez taper `main`, puis enfoncer `[Ctrl]+[espace]` et enfin appuyer sur `[Entrée]`). Il est également possible de générer automatiquement une méthode principale lors de la création de la classe en cochant la case appropriée.
7. Cette méthode doit afficher `Hello, world!` dans la console Java (plutôt que de taper intégralement `System.out.println`, vous pouvez taper `sysout`, puis enfoncer `[Ctrl]+[espace]`).
8. Compilez et exécutez votre projet. Pour cela, faites un clic-droit sur le nom de votre projet et sélectionnez *Run As* puis *Java Application* (ou cliquez sur l'icône raccourcie *Run As...* dans la barre d'outils). Le résultat de l'exécution apparaît dans la console Java en bas.
9. Faites un clic-droit sur le nom de votre projet et sélectionnez *Properties*. Dans la fenêtre qui s'ouvre, vous pouvez, entre autres, modifier la version du compilateur (*Java Compiler*) ou de l'environnement d'exécution (*Java Build Path*, onglet *Libraries*) à utiliser.

Exercice – archive (.jar)

1. Générez une archive de votre projet. Pour cela, faites un clic-droit sur le nom de votre projet et sélectionnez *Export...*, puis *JAR file* dans le dossier *Java*.
2. Dans la fenêtre qui s'ouvre, sélectionnez les fichiers à archiver (ici, tout votre projet, y compris les fichiers `.classpath` et `.project`), cochez uniquement la case *Export Java source files and resources*, indiquez un emplacement et un nom d'archive, puis cliquez sur *Next*. Cochez la case *Create source folder structure*, puis cliquez sur *Finish*.
3. Renommez votre projet en le suffixant par "bis". Pour cela, faites un clic-droit sur le nom de votre projet et sélectionnez *Refactor*, puis *Rename...*
4. Dans le menu *File*, sélectionnez *Import...*, puis *Existing Projects into Workspace* dans le dossier *General*.

5. Dans la fenêtre qui s'ouvre, sélectionnez l'archive que vous avez précédemment générée, puis cliquez sur *Finish*. Le projet archivé est recréé. Compilez et exécutez-le.
6. Supprimez votre projet terminant par "bis". Pour cela, faites un clic-droit sur le nom de votre projet et sélectionnez *Delete*. Dans la fenêtre qui s'ouvre, cochez la case pour supprimer définitivement le projet du disque dur (sinon, il reste dans votre espace de travail, vous empêchant ainsi de créer un projet avec le même nom), puis cliquez sur *OK*.
7. Recompilez et exécutez le projet.
8. Générez une archive exécutable de votre projet. Pour cela, faites un clic-droit sur le nom de votre projet et sélectionnez *Export*, puis *Runnable JAR file* dans le dossier *Java*.
9. Dans la fenêtre qui s'ouvre, sélectionnez la bonne configuration de lancement et indiquez un emplacement et un nom d'archive exécutable, puis cliquez sur *Finish*.
10. Ouvrez un terminal et exécutez l'archive : `java -jar archive.jar` (pensez à donner le chemin absolu ou relatif de l'archive). Si vous examinez le contenu de l'archive (*e.g.*, en l'extrayant avec la commande : `jar xfv archive.jar`), vous trouverez tous les fichiers `.class` nécessaires à l'exécution du projet, ainsi qu'un fichier `MANIFEST.MF` dans le répertoire `META-INF`. Ce fichier contient le nom de la classe à exécuter et les archives à mettre dans le `CLASSPATH`.

Exercice – outil d'automatisation Ant

Nous souhaitons créer un script Ant permettant d'automatiser les tâches suivantes :

- Supprimer le répertoire `bin` contenant les fichiers `.class` (tâche *delete*),
 - Recréer le répertoire `bin` vide (tâche *mkdir*),
 - Compiler les fichiers sources `.java` (tâche *javac*),
 - Afficher un message indiquant que la compilation est en cours (tâche *echo*),
 - Générer l'archive exécutable (tâche *jar*),
 - Exécuter le programme (tâche *java*).
1. Générez de nouveau une archive exécutable de votre projet en cochant cette fois-ci la case *Save as ANT script*.
 2. Créez un fichier `build.xml` à la racine de votre projet et recopier le contenu du script Ant créé précédemment par Eclipse. Pour cela, faites un clic-droit sur le nom de votre projet et sélectionnez *New*, puis *File*.
 3. Complétez ce fichier en y ajoutant les tâches listées précédemment (la tâche *jar* étant déjà écrite). Commencez par mettre toutes les tâches dans une même cible (balise `target`). Une cible correspond à un ensemble de tâches. Pour comprendre comment écrire les différentes tâches Ant, aidez-vous du tutoriel suivant : <http://www.jmdoudoux.fr/java/dej/chap-ant.htm#ant-2>.
 4. Dans le menu *Window*, sélectionnez *Show View*, puis *Ant*.
 5. Cliquez sur l'icône *Add Buildfiles* (première icône) de la vue.
 6. Dans la fenêtre qui s'ouvre, ajoutez le fichier `build.xml` de votre projet à la vue.
 7. Sélectionnez la cible et exécutez-la. Le résultat de l'exécution apparaît dans la console Java.
 8. Créez trois nouvelles cibles, une nommée `clean` pour le nettoyage, une autre nommée `build` pour la compilation, et une dernière nommée `run` pour l'exécution. Puis, utilisez l'attribut `depends` de la balise `target` de sorte que les cibles génération de l'archive exécutable et exécution dépendent de la cible compilation qui elle-même dépend de la cible nettoyage.
 9. Modifiez les propriétés `default` et `name` de la balise `project`, et exécutez de nouveau votre script Ant.