Objet : **TP Java sous Eclipse : quelques consignes et aides**

De : A. Morelle

Ces consignes et informations ne sont pas censées être contradictoires avec celles qui vous sont notifiées sur Campus. Si néanmoins tel pouvait être localement le cas, merci de me le signaler.

**1- CONSIGNES**

**1.1- A la création du projet, créer des répertoires séparés pour les fichiers sources et binaires**

Exemple : créer un projet TP2\_java comportant les répertoires src/ et bin/

**File > New > Project… > Java Project > Next**

Puis dans la page “Create a Java Project” :

Project name : ***TP2\_java***

Sélectionner : **Create separate folders for sources and class files**

Puis : **Finish**

**1.2- In fine, exporter l’ensemble du projet (fichiers sources inclus) en un fichier .jar exécutable**

Exemple : exporter le projet TP2\_java en un fichier exécutable complet de nom

nom1-nom2.jar

**Clic droit sur le projet > Export... > Next > Java > JAR file > Next**

Puis dans la page “JAR File Specification” :

Select the resources to export: ***TP2\_java***

Cocher : **Export generated class files and resources**

Cocher : **Export Java source files and resources**

Jar file: ***nom1-nom2.jar*** (le fichier .jar est créé par défaut

dans le répertoire racine du workspace)

Puis : **Next**

Puis dans la page “JAR Packaging options” : choisir éventuellement des options puis **Next**

Puis dans la page “JAR Manifest Spécification” : sélectionner la classe contenant la méthode "main" !! par **Main class: Bownse…**

Puis **Finish**

Puis vérifier (en mode console sous Windows) que le fichier .jar généré est bien exécutable :

Sous Windows : **Démarrer > Tous les programmes > Accessoires > Invites de commande** (ou bien : Démarrer > Exécuter > cmd)

Puis en mode console :

* se déplacer (**cd *chemin***) dans le répertoire contenant le fichier .jar
* lancer l’exécution par la ligne de commande : **java –jar *fichier.jar***

**2- FAQ Eclipse (3.6)**

**2.1- Quels sont les raccourcis clavier les plus utiles en phase d’édition ?**

Outre les classiques CTRL+F, CTRL+C, CTRL+X, CTRL+S, … :

**CTRL + Espace** Propose une liste de choix d’actions

**CTRL + SHIFT + F** Formate le code (toute la classe ou la seule partie sélectionnée)

**ALT + SHIFT + R** Renomme un identificateur (de variable, de méthode, de classe, ...) et

met à jour automatiquement le reste du code. Equivalent de :

**clic droit > Refactor > Rename…**

**CTRL + SHIFT + /** Met en commentaire (par //) ou "dé-commente" les lignes

sélectionnées

**2.2- Comment créer un diagramme UML ?** (plug-in Omondo supposé installé)

**Clic droit sur un package > New UML Diagram** (ou New > Other… > UML Diagrams)

Choisir **UML Class Diagram** permet de créer dans un diagramme de classes UML dont le code source sera ensuite automatiquement généré. Les mises à jour sont ensuite bidirectionnelles : dans le code et sur le diagramme.

Pour ajouter des membres (attribut, constructeur, méthode) à une classe affichée dans le diagramme UML :

**clic droit sur la classe > New > Attribute > Java Attribute**

ou **clic droit sur la classe > New > Method > Java Method**

ou **clic droit sur la classe > New > Constructor**

puis spécifier toutes les caractéristiques (nom, types, paramètres, visibilité, modificateur static ou final, … selon).

Nota1. Le code est généré au fur et à mesure que les membres de la classe sont créés dans la classe du diagramme UML. Le code produit ne sera donc pas forcément présenté dans l’ordre logique habituel (attributs, constructeurs, accesseurs, méthodes, main, …). Une remise en forme du code source est donc généralement nécessaire.

Nota2. Un diagramme de classes est sauvegardé dans un fichier .ucd (Uml Class Diagram).

**2.3- Comment visualiser un diagramme UML ?** (plug-in Omondo supposé installé)

Cas d’un diagramme de classes.

* Si le diagramme de classes UML est déjà disponible dans le package, le visualiser par :

**clic droit sur le fichier .ucd du package > Open With > UML Class Diagram Editor**

(ou plus simplement : double clic gauche sur le fichier .ucd)

* Sinon générer le diagramme UML à partir des fichiers sources du package (« reverse engineering ») :

**clic droit sur le package > Open / Reverse UML > Class Diagram**

Nota.Dorénavant, toute mise à jour du code source affectera la classe UML correspondante et réciproquement !

Pour afficher tous les compartiments (attributs, méthodes, …) des classes du diagramme : **clic droit dans la zone d’édition du diagramme > Afficher/Masquer > Afficher/Masquer Compartiments > Tous les compartiments**

Pour afficher / mettre à jour toutes les liaisons (héritage, association, …) entre les classes du diagramme : **clic droit dans la zone d’édition du diagramme** puis

**Associations > Afficher Toutes les Associations**

**Héritage > Afficher Tous les Héritages**

**2.4- Comment intégrer dans un projet des fichiers sources existant ailleurs (d’un projet précédent, par exemple) ?**

Deux façons de faire :

* Soit hors Eclipse : copier ces fichiers .java dans le répertoire cible du projet (répertoire src/ ou sous-répertoire (package) approprié). Puis, si besoin, actualiser l’affichage dans le navigateur Eclipse par : **clic droit sur le projet** **>** **Refresh**
* Soit sous Eclipse : **clic droit sur le projet** **>** **Import... > File System > Browse… (**choisir le répertoire d’origine puis les fichiers à importer) **> Browse…** (choisir le répertoire destination) **> Finish**

**2.5- Comment intégrer dans un projet des fichiers .class existant ailleurs ?**

Supposons que l’on veuille intégrer dans le projet TP un fichier Clavier.class qui a été généré par ailleurs dans un package utilitaires/

Deux façons de faire :

* Soit hors Eclipse : créer sous le répertoire TP un sous-répertoire lib/ (exemple) ; dans lib/ créer un sous-répertoire utilitaires/ ; copier dans utilitaires/ le fichier Clavier.class. Puis, si besoin, actualiser l’affichage dans le navigateur Eclipse par : **clic droit sur le projet TP** **>** **Refresh**
* Soit sous Eclipse :
* **clic droit sur le projet TP > Properties > Java Build Path > Librairies > Add Class Folder… > Create New Folder…** (choisir un nom pour le répertoire, lib/ par exemple) **> Finish**
* Puis **clic droit sur le répertoire lib créé > New > Folder** (fournir le nom du répertoire : utilitaires/) **> Finish**
* Puis **clic droit sur le répertoire utilitaires créé > Import... > File System > Next > Browse…**: sélectionner le répertoire contenant le fichier Clavier.class à importer, puis cocher le fichier Clavier.class dans la liste affichée, puis vérifier si le champ Into Folder est correct, puis > **Finish**

Toute classe Java du projet TP peut maintenant utiliser la classe Clavier après l’avoir importée par la directive : import utilitaires.Clavier

**2.6- Comment intégrer dans un projet une bibliothèque .jar existant ailleurs ?**

Ce fichier .jar est supposé contenir les fichiers .class à importer.

Deux cas de figure : soit on souhaite importer la bibliothèque (c’est-à-dire la recopier dans le projet afin d’obtenir un projet autonome), soit on souhaite simplement pouvoir y référer en spécifiant son chemin d’accès (au risque qu’un utilisateur externe n’y ait pas accès : voir nota ci-dessous).

* 1er cas : importation
* **clic droit sur le projet > Properties > Java Build Path > Librairies > Add Class Folder… > Create New Folder…** (choisir un nom pour le répertoire, lib/ par exemple) **> Finish**
* Puis **clic droit sur le répertoire lib créé > Import... > Archive File > Next > Browse…** (choisir la bibliothèque .jar à importer, puis désélectionner les éventuels fichiers dont les noms commencent par un "." (fichiers .classpath, .project, …)), puis **> Finish**
* 2ème cas : simple spécification du chemin d’accès
* **clic droit sur le projet > Properties > Java Build Path > Librairies > Add External JARs…** (choisir la bibliothèque .jar concernée) **> Finish**
* Attention. Si l’on souhaite maintenant produire un .jar exécutable de ce projet, il faut inclure les jar externes auxquels il réfère. Pour cela : exécuter au moins une fois le projet (produit une configuration de lancement) puis : **clic droit sur le projet > Export… > Runnable JAR File > Next > Browse…** (spécifier le nom du fichier export .jar) **> Finish**

**2.7- Comment générer la Javadoc ?**

Pour générer la documentation d’un projet :

Menu **Project > Generate Javadoc…** , puis sélectionner le projet concerné **(Select types for which Javadoc will be generated**), sélectionner **Use standard doclet,** puis **> Finish**