TP1 : semaines 1 et 2

Avant-propos

- L'environnement de développement supporté est NetBeans (http://netbeans.org/downloads/index.html). Vous trouverez des tutoriaux sur campus : voir « **Mes premiers pas sur NetBeans** » en section 1 de la page http://campus.ece.fr/course/view.php?id=124.

- Pour ce TP et ceux qui suivront, la documentation des API Java *(javadoc)*, que je vous conseille vivement de consulter, sera votre référence pendant que vous coderez.

- Votre code sera clairement indenté, commenté et très bien découpé et testé selon les directives qui suivront.

- Le site d'openclassroom peut emmener un complément d'aide aux débutants : voir le lien Openclassrooms : apprenez à programmer en Java sur la page campus.

Hello World

Suivez les instructions de la page internet jusqu'à la section Building and Deploying the Application

http://netbeans.org/kb/docs/java/quickstart.html

Comprenez la distinction entre **Run** (pour inspecter le bon déroulement de votre projet dans l'IDE) et **Build** (pour distribuer l'application à l'utilisateur final).

Avant de faire votre Build de votre projet, respectez bien les consignes du chapitre 2 « inclure les fichiers sources .java dans le jar » du cours voir « **Mes premiers pas sur NetBeans** » en section 1 de la page http://campus.ece.fr/course/view.php?id=207.
Quand vous « buildez » votre projet avec l'icône "Marteau" ou Clean and Build Main Project (Shift-F11), un message à la compilation vous indique une commande pour lancer votre programme SANS l'IDE.

Avec un explorateur ou un finder, allez dans le répertoire Dist (pour Distribution) de votre projet. Vous y trouverez votre exécutable **.jar** ainsi que ses éventuelles « librairies » (répertoire lib, nécessaire dans les derniers TP).

Visualisation des sources
L'extension .jar est votre « .exe » java. Néanmoins le jar java n'est pas un binaire (suite de 0 et de 1 incompréhensible pour le commun des mortels) comme en C.

Changez l'extension de votre « exe » java de .jar en .zip. Pour les utilisateurs de Win dont l'extension n'apparaît pas, suivez ce lien http://www.commentcamarche.net/faq/825-afficher-les-extensions-et-les-fichiers-caches-sous-windows.
Visualisez le contenu de votre .zip. Il contient les classes de votre projet avec l'extension « .class ». « .class » est le binaire pour la JVM (http://fr.wikipedia.org/wiki/Machine_virtuelle_Java).

Pour nos TP, nous désirons voir aussi apparaître les sources de votre projet dans votre « .jar ».
POUR CHAQUE PROJET, il faudra aller dans les Options du projet et ENLEVER (car exlude) la ligne *.*.

Êtes-vous maintenant capable de :
- Créer un projet
- Compiler un projet
- Visualiser le code source d'un projet à partir de son jar
- Lancer un jar en dehors de l'IDE

Exercices

1) Sur NetBeans, créez un nouveau projet, cliquer sur **+** devant **Source Packages** et sur le nom du package (par défaut, le même nom que le projet), supprimer le **.java** créé par défaut (clic droit, puis **Delete** et **OK**). Créez ensuite chacune des classes suivantes dans le package du projet en respectant leur nom (fichier source **.java** avec le même nom que la classe) : voir « **Mes premiers pas sur NetBeans** » sur campus.

2) Copiez le code de la classe *Essais*. Compilez-la (**Build Project** dans le menu **Run** ou touche clavier **F11** ou l'icône du marteau), corrigez les erreurs et exécutez-la (clic droit sur le fichier **.java** de la classe et **Run File**).

```
public class Essais {
        public static void main(String args[]) {
                /*
                        Identifiez l'erreur 1 et corrigez
                 */
                int i = 0;
                for (int i = 0; i < 5; i++)
                        System.out.print(i + ", ");
                System.out.print("\n");

                /*
                 Identifiez l'erreur 2 et corrigez
                */
                float a = 3.0;
                double b = 4;
                float c;
                c = Math.sqrt(a * a + b * b);
                System.out.println("c = " + c);

                /*
                 Identifiez l'erreur 3 et corrigez
                */

                byte b = 42;
                char c = 'a';
                short s = 1024;
                int i = 50000;
                float f = 5.67f;
                double d = .1234;
                double resultat = (f * b) + (i / c) - (d * s);
                System.out.print((f * b) + " + " + (i / c) + " - " + (d * s));
                System.out.println(" = " + resultat);
                byte b2 = 10;
                byte b3 = b2 * b;
                System.out.println("b3 = " + b3);
        }
}
```

3) Quand il n'y a plus d'erreur de compilation, une archive exécutable **.jar** est créée. Allez dans ce dossier **dist** de votre projet (son dossier porte le nom de votre projet) et ouvrir cette archive comme un zip. Cette archive contient 2 dossiers : celui du projet (avec le nom que vous lui avez donné) et un dossier **META-INF** (contient le fichier de configuration **MANIFEST-MF**). Le dossier du projet ne contient que les 3 fichiers **.class** (fichiers compilés) mais pas les sources **.java**. Pour les intégrer, allez sur NetBeans et suivez les consignes « Inclure les fichiers sources dans le jar » indiquées sur « **Mes premiers pas sur NetBeans** ». Recompilez et vérifiez le contenu du jar. Hourrah ! les fichiers sources **.java** s'y trouvent ! Par la suite vous respecterez à la lettre toutes ces consignes.

JP Segado

Exercice 2

- Ecrire une classe **Etudiant** :

- Définissez comme attributs privés un identifiant de type **String**, un tableau *notes* de flottants, un entier *taille* et des attributs publiques que le nom et prénom.

- Implémentez un premier constructeur sans paramètre qui initialise par défaut les valeurs des attributs (0 pour un nombre, **null** pour un objet)**.**

NB : un constructeur est forcément public et porte toujours le nom de classe. Son rôle est d'initialiser les attributs. Pour instancier un objet, il faudra appeler ce constructeur avec **new**. Un constructeur peut être redéfini (surchargé) à condition que les types des paramètres ne soient pas les mêmes pour un nombre de paramètres identique

- Implémentez un second constructeur (surcharge) qui prend en paramètre tableau de flottants, initialise la taille de ce tableau dans l'attribut *taille*, et copie ses valeurs dans le tableau *notes*.

NB : la taille d'un tableau se mesure avec l'attribut prédéfini **length**

- Implémentez une méthode *modifier* qui permette de modifier les valeurs des attributs avec les paramètres (autant de paramètres que d'attributs).

- Ecrire une classe **TestEtudiant** qui implémente le **main** (voir exemples de l'exercice 1) dont les traitements sont les suivants :
- Déclarez 2 objets de la classe **Etudiant** et initialisez-les.
- Instanciez le premier objet en appelant le premier constructeur. Tentez d'afficher les valeurs des attributs de cet objet. Pourquoi y-a-il un problème ? Quelle solution proposez-vous résoudre de problème sans changer la visibilité des attributs ? Mettez en œuvre votre solution.
- Déclarez un entier et un tableau de flottants.
- Saisissez au clavier la valeur d'un entier en vérifiant que qu'il est positif (boucle de blindage)
- Instanciez le tableau avec pour taille l'entier saisi et saisissez au clavier les valeurs de ce tableau

La lecture d'information au clavier durant l'exécution d'un programme en mode console s'utilise avec la classe **Scanner** : voir exemples en section 1 de la page http://campus.ece.fr/course/view.php?id=207

- Instanciez le second objet en appelant le second constructeur avec pour paramètre le tableau.
- Saisissez au clavier les données suivantes : un identifiant, un nom et un prénom
- Modifiez les valeurs des attributs du premier objet à partir des données saisies (en paramètres) en appelant la méthode *modifier*
- Proposez une solution pour pouvoir afficher les valeurs du tableau *notes* qui est privé.

- Dans la classe **Etudiant**, implémentez les méthodes suivantes à tester dans le **main** avec les 2 objets cités ci-dessus :

- Afficher le tableau *notes* si ce n'est pas déjà fait ☺

- Remplir aléatoirement le tableau *notes* entre 0 et 20 et l'afficher en appelant la méthode ci-dessus.

La classe **Random** permet de générer l'aléatoire : consultez la documentation des API Java *(javadoc)*

- Trier le tableau *notes* par ordre croissant. Afficher le tableau avant et après le tri

- Afficher les statiques suivantes à partir du tableau *notes* : minimum, maximum, moyenne, pourcentages des notes < 8, entre 8 et 12 et > 12.

- Retourner un booléen indiquant si un nombre passé en paramètre est dans le tableau *notes*.

- Une autre méthode **toString()** de type **String** retourne l'expression composée de l'identifiant, du nom et du prénom séparés par un espace. Cette expression sera affichée avec une autre méthode.

Un objet contient par défaut plusieurs méthodes de base dont la méthode surchargée **toString()** :
Réécrivez toujours la méthode **toString()** ne serait-ce que pour déjà faciliter le débogage

```
@Override
public String toString()
{
        // Une description synthétique de votre objet ou texte pour une sortie console
        return … ;
}
```

- Toute autre méthode utile qui vous passe par la tête, comme par exemple les getters (accesseurs en lecture) et setters (accesseurs en écriture)

- Essayer alors de lancer votre exécutable **.jar** en ouvrant une fenêtre de commande (cmd pour Win, Terminal pour Mac/Linux). Retapez « **java –jar** » espace. « Drag&Drop » votre exécutable **.jar** du répertoire **Dist** de votre projet (clic-droit appuyé, déplacez le fichier jar et lâchez) vers la fenêtre.

- Commentez sous la forme **Javadoc** /** commentaires */ devant chaque classe, attribut et méthode. Puis générez la javadoc. Ouvrez-la et vérifiez que les commentaires sont visibles. Ajoutez-la dans le **.jar** : voir les consignes indiquées sur « **Mes premiers pas sur NetBeans** ».