TP N°1 : Préparation de l'environnement de travail et exemples de programmes simples

1. Installation du JDK

Eclipse a besoin d'un Runtime Java (JRE) ou d'un JDK pour fonctionner. Préférez l'installation d'un JDK au lieu d'un JRE pour bénéficier automatiquement de la *Javadoc* et du code source de l'API Java Standard. La version de la JDK à installer est la dernière version stable disponible sur le site d'Oracle.

http://www.oracle.com/technetwork/java/javase/downloads/index.html

2. Java en ligne de commande

Bien que ce ne soit pas le mode de développement le plus utilisé, Java peut être utilisé en ligne de commande. Il est toujours utile de savoir compiler et exécuter une application Java en ligne de commande.

Pour compiler votre programme, vous devez utiliser le compilateur Java (javac).

Après compilation, s'il n'y a pas eu d'erreur, un fichier .class sera créé.

Pour exécuter le fichier *.class*, vous devez appeler l'interpréteur Java à l'aide de la commande : java MaClasse

3. Installation et configuration d'Eclipse

Eclipse est un logiciel qui simplifie la programmation en proposant un certain nombre de raccourcis et d'aides. Il est développé par IBM, il est gratuit et disponible pour la plupart des systèmes d'exploitation.

Au fur et à mesure que vous programmez, Eclipse compile automatiquement le code que vous écrivez, en soulignant en rouge ou jaune les problème qu'il décèle. Il souligne en rouge les parties du programme qui ne compilent pas (Errors), et en jaune les parties qui compilent mais peuvent éventuellement poser problème (warning).

Dans les prochains TP de Java, nous utiliserons Eclipse comme environnement de développement.

Il faudrait utiliser la dernière version stable d'Eclipse qui peut être téléchargée de la page *http://www.eclipse.org/downloads*. Eclipse est utilisé pour le développement C/C++ , PHP, Java, JEE, … pour le développement Java standard le package à télécharger est « Eclipse IDE for Java Developers » . Eclipse est un logiciel portable ainsi pour l'installer il suffit de décompressez l'archive d'Eclipse dans un répertoire de votre disque dur.

a) Lancer Eclipse.

b) Créer votre WorkSpace dans le répertoire X:\Programmation_JAVA\WorkSpaceTP

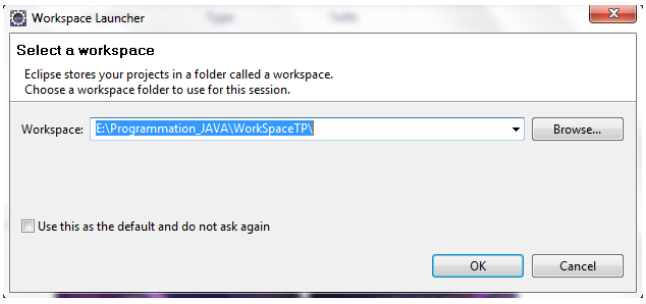


Figure 1

Qu'est-ce qu'un Workspace ?

Le workspace est le répertoire dans lequel sont stockés les différents programmes que vous allez réaliser et la configuration d'Eclipse. Ainsi si vous supprimer le workspace votre configuration Eclipse sera perdu. En toute rigueur les paramètres de configuration sont stockés dans un sous-dossier caché nommé .metadata, ainsi, effacer ce dossier ou démarrer Eclipse avec l'option -clean va supprimer ces paramètres.

Comment changer le workspace ?

Il est possible de changer le workspace à tout moment. Le workspace en cours est alors fermé.

Procédure :

- *Aller dans le menu « File »*
- *Sélectionner « Switch workspace».*

c) Eclipse s'ouvre sur la page

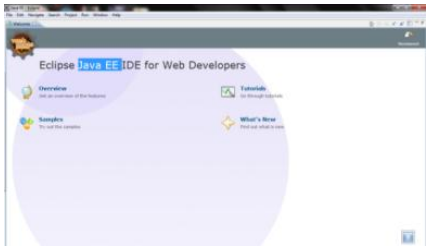


Figure 2

d) Cliquer sur Workbench pour commencer à travailler.

Comme la plupart des IDE, Eclipse contient un nombre considérable d'options. La configuration par défaut est acceptable pour ce premier TP, il faudrait juste vérifier la version du « compiler » et configurer Eclipse pour qu'il utilise la JDK.

e) Vérifier que la version du compilateur : *Windows ➔ Preference ➔ JAVA ➔ Compiler*

f) Configurer Eclipse pour qu'il utilise la JDK : *Windows ➔ Preference ➔ JAVA ➔ Installed JREs ➔ Add ➔Standard VM* . Ensuite il faut sélectionner le répertoire où la JDK est installée sur votre disque dur (Son emplacement par défaut est : C:\Program Files \Java)

g) Cliquer sur Finish

h) En fin sélectionner la JDK et cliquer sur OK

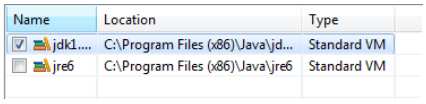


Figure 3

4. Création d'un projet JAVA

a) Pour créer un projet JAVA : File ➔ New ➔ Project.

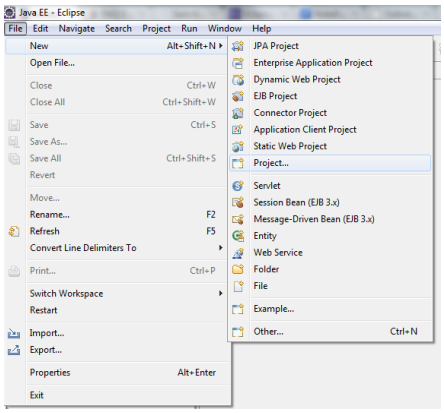


Figure 4

b) Choisir Java Project

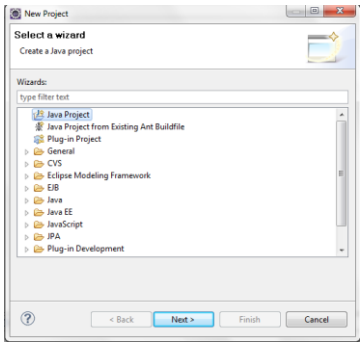


Figure 5

c) Donner un nom à votre projet (par exemple MyFirstProject) et cliquer sur Finish

5. Création d'un package

En Java, les classes sont organisées en package qui forment une arborescence. Un package est un ensemble de classes attachées à un même concept. Un exemple est donné par l'API Java fournie en standard on trouve le packages suivants : java.io qui regroupe toutes les classes associées aux entrées sorties.

L'organisation sur disque des fichiers source doit refléter l'organisation en packages

Pour créer un package clique droite sur le dossier des sources (src) , donner un nom à votre package (par exemple com.ac.ensah.tp) et cliquer sur Finish

6. Création d'une classe

Rappel sur l'architecture générale d'un programme Java :

Programme source Java = ensemble de fichiers « .java » chaque fichier « .java » contient une ou plusieurs définitions de classes. Au plus une définition de classe public par fichier « .java » avec nom du fichier = nom de la classe publique.

a) Pour créer une classe : click droite sur le package dont vous voulez ajouter la classe ➔ New ➔ Class

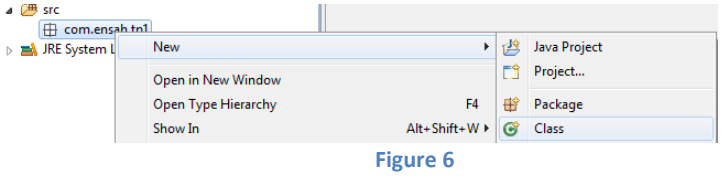


Figure 6

b) Donner un nom à votre classe et cliquer sur Finish

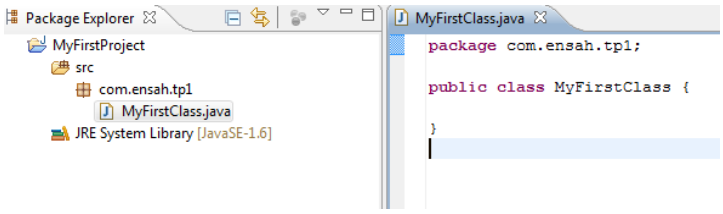


Figure 7

7. Compilation

Par défaut, lorsqu'un projet est créé, Eclipse active l'option de « compilation automatique ».

Ainsi, votre projet est compilé en temps réel, dès que vous tapez du code et sauvegardez un fichier ou que vous en lancez l'exécution. Vous pouvez désactiver la compilation automatique dans le menu Project en décochant « *Build Automatically* ».

8. Exécution d'un programme

a) Ajouter la méthode main suivante dans votre classe

```
public static void main(String[] args) {
 System.out.println("Bonjour ENSAH");
}
```

Astuce

On peut générer une méthode main avec la façon suivante :

Ecrire main dans le corps de la classe puis cliquer sur CTRL+ESPACE et puis sur la touche Entrer.

On peut générer l'instruction « System.out.println(); » ainsi :Ecrire Syso puis cliquer sur CTRL+ESPACE .

En général lorsque vous écrivez votre programme, Eclipse peut vous aider à compléter ce que vous écrivez, en utilisant le raccourci clavier CTRL+ESPACE .

b) Exécuter le programme en cliquant sur l'icône de la barre d'outils ou click droite sur la classe ➔ Run As ➔ Java Application. Le résultat d'exécution du programme s'affiche dans la console :

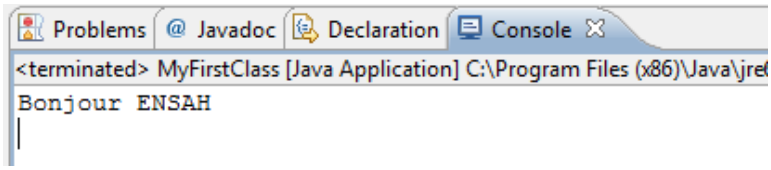
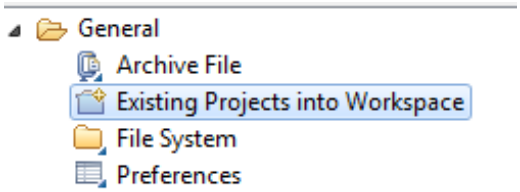


Figure 8

9. Ouverture d'un projet existant

a) Importer le projet TBoudaaProjet dans votre WorkSapce File ➔ import ➔ General ➔ Existing projects into Workspace



b) Chercher dans le projet une classe MyClass (Utiliser le raccourcis clavier CTRL+SHIFT+R)

c) Chercher dans le projet une méthode nommée myMethode (Utiliser le menu search d'Eclipse)

Raccourcis clavier indispensables:

Vous pouvez afficher la liste des raccourcis clavier pour Eclipse par CTRL + SHIFT + L

Il est possible de possible de configurer soi-même sa combinaison pour les raccourcis clavier en allant dans le menu *Window > Preferences > General > Keys.*

Les raccourcis suivant sont très utiles :

CTRL + Espace : auto-complétion

CTRL + SHIFT + O : Organize Imports

CTRL + SHIFT + R : Recherche d'un fichier dans le workspace

CTRL + SHIFT + T : Rechercher une classe dans le workspace

CTRL + T : Affiche l'arborescence d'héritage de la classe courante

CTRL + SHIFT + F : Formatage du code (vous pouvez surligner une zone de code pour restreindre le formatage)

CTRL + SHIFT + I : Indentation du code (vous pouvez surligner une zone de code pour restreindre l'indentation)

CTRL + SHIFT + C : pour commenter / dé-commenter les lignes sélectionnées

CRTL + SHIFT + P : Pour se déplacer d'une accolade à l'autre

CTRL + K et CTRL + SHIFT + K : Positionne le curseur sur l'occurrence suivante ou précédente de la sélection de départ.

10. Exemples de programmes simples

Réaliser sur Eclipse les programmes suivants :

Exercice 1

Ecrire un programme Premier.java qui permet de tester si un nombre introduit par l'utilisateur est premier ou non.

Exercice 2

Ecrire un programme FormatHour qui prend en paramètre un nombre de secondes et qui permet de le formater en heures-minutes-secondes. Si on ne fournit pas un paramètre lors de l'appel du programme ou si le paramètre fourni n'est pas un entier, le programme devrait afficher message d'erreur.

Voici un exemple d'exécution du programme :

> java FormatHour 520

8 minutes 40 secondes

> java FormatHour

> Arg Error.

> java FormatHour 25217

7 heures 17 secondes

Exercice 3

Ecrire un programme qui affiche les nombres amis inférieurs ou égales à 10000.

Exercice 4

Soit n un nombre naturel, un diviseur d positif de n est appelé diviseur propre si d est différent de n.

Deux entiers naturels sont des nombres amis si chacun d'entre eux est égal à la somme des diviseurs

Propres de l'autre.

Ecrire un programme Java qui affiche la liste des couples de nombres amis inférieurs à 20000.

Exercice 5

Ecrire un programme Java qui calcule et affiche la valeur de π approchée sachant que $\frac{\pi^2}{6} = \sum_{i=1}^{\infty} 1/i^2$.

Exercice 6

Une méthode qui calcule la valeur approchée de cos(x) en utilisant le développement en séries entières de la fonction cosinus :

$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \cdots + (-1)^n \frac{x^{2n}}{(2n)!} + \cdots$$

La boucle de traitement devra s'arrêter quand la précision obtenue sera suffisante.

Exercice 7

On considère deux suites numériques (Un) et (Vn) telles que pour n strictement supérieur à 2 :

$U_n = U_{n-1} + U_{n-2}$ Et $V_n = U_n/U_{n-1}$

La suite (Vn) tend vers une limite ℓ appelée nombre d'or. (ℓ = 1,6180339887498948482045868343656 4).

Ecrire un programme qui calcule une valeur approchée du nombre d'or avec une précision fixée.

Exercice 8

On appelle nombres d'Armstrong les nombres entiers positifs tels que la somme des cubes de leurs chiffres est égale au nombre lui-même. Exemple : 153 est un nombre d'Armstrong. En effet : $1^3 + 5^3 + 3^3 = 153$

Écrire un programme JAVA qui affiche tous les nombres d'Armstrong inférieurs à 1000000. Résoudre cet exercice sans utiliser ni tableaux ni collections.