

# P.O.O. : Java — Sujet n°1

## Les bases du langage

### Avant de commencer : Eclipse

*Eclipse* est un IDE (Integrated Development Environment ... EDI en français). Il est développé par IBM, est gratuit et disponible pour la plupart des systèmes d'exploitation.

C'est un logiciel qui va simplifier votre programmation Java en proposant un certain nombre de raccourcis et d'aide à la programmation. Par exemple, au fur et à mesure que vous écrivez le programme, *eclipse* analyse le code et souligne en rouge les parties qui ne passeront pas à la compilation et en jaune les parties qui pourront passer à la compilation mais qui peuvent éventuellement poser problème. Cela peut parfois paraître déroutant puisque tant que la ligne n'est pas terminée (le point virgule est donc manquant), *eclipse* indique une erreur en soulignant le tout en rouge !

### Si vous devez installer Eclipse :

*Eclipse* est simplement une interface, écrite en Java, qui va vous permettre de développer des programmes en ... Java. Ce logiciel peut donc être simplement vu comme un « enrobage » de Java. *Eclipse* peut également être utilisé pour le développement d'applications écrites avec d'autres langages de programmation.

La première chose à vérifier est que l'environnement Java (JRE) est installé sur votre machine. Si ce n'est pas le cas, vous le trouverez à l'adresse : <http://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html>.

Une fois, l'environnement Java installé sur votre machine, *eclipse* peut alors être installé. Rendez-vous sur : <http://www.eclipse.org/downloads/> et récupérez l'archive « Eclipse IDE for Java Developers ». Décompressez-la où vous le souhaitez. Vous obtenez un dossier qui contient, entre autres, le logiciel *eclipse*.

### Utilisation d'Eclipse

Lancez le logiciel *eclipse*. La fenêtre « Workspace Launcher » s'ouvre alors. Elle vous demande d'indiquer le dossier dans lequel tous vos programmes Java seront rangés. Créez donc un dossier « Java » dans votre répertoire personnel et spécifiez son nom dans Eclipse ; pensez à cocher la case « Use this as the default and do not ask again » pour que la question ne vous soit plus posée.

La page d'accueil d'*eclipse* s'affiche alors ... vous pouvez immédiatement la fermer :-). Vous voilà prêts à coder en Java.

Pour avoir un espace de travail plus grand, vous pouvez fermer les 2 petites fenêtres qui se trouvent sur la droite : « Task List » et « Outline ».

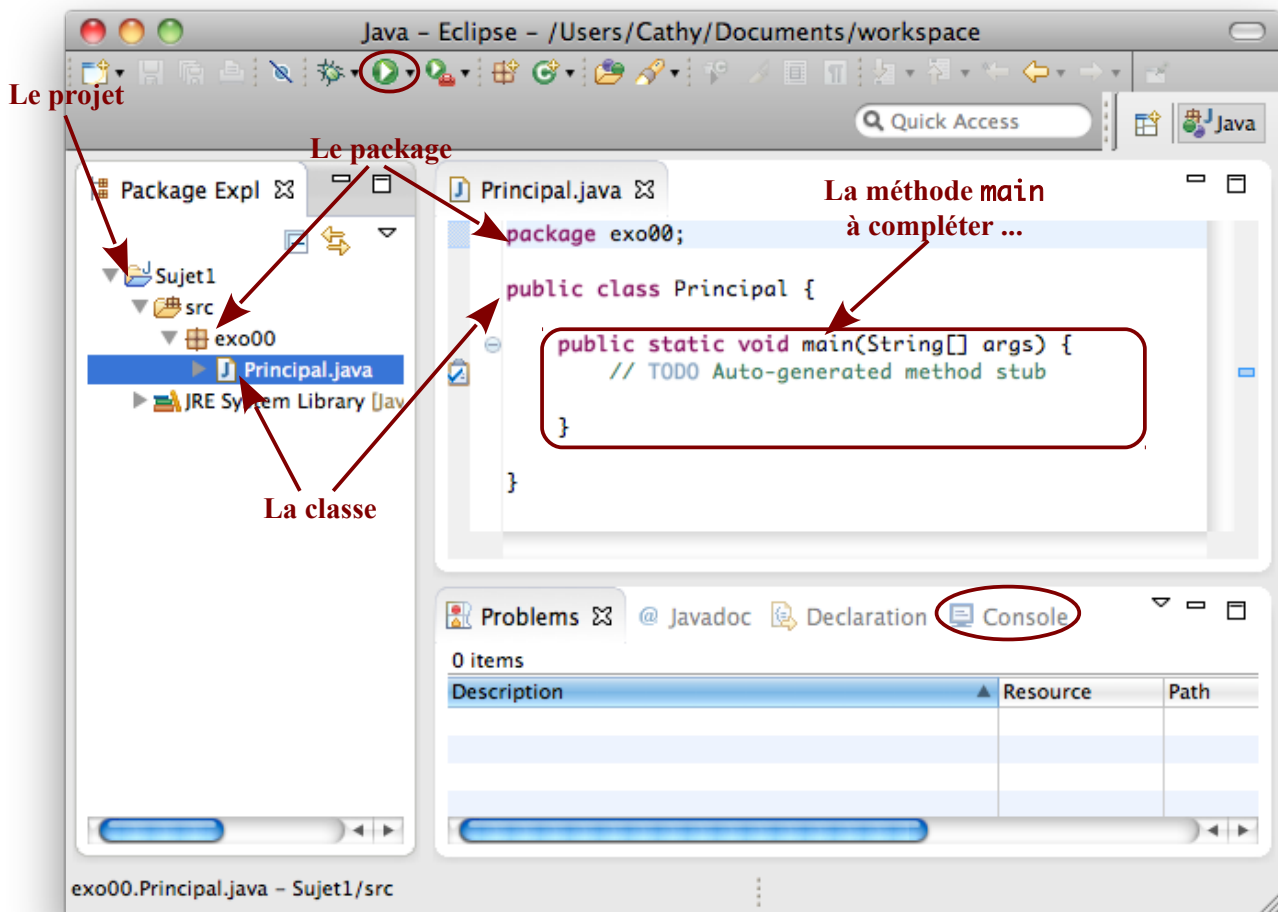
La fenêtre « Package Explorer » va permettre de retrouver vos programmes. La grande partie, vide pour l'instant, contiendra le code Java des programmes. La partie basse, permettra de suivre l'exécution des programmes.

**Chaque feuille d'exercices correspondra à un projet (project).** À chaque nouvelle feuille, il faudra donc aller dans le menu File et choisir l'entrée New/Java Project. Dans la fenêtre qui s'ouvre alors, donner un nom au projet (« Sujet1 » pour cette première feuille d'exercices) et cocher la case qui vérifie que le JRE utilisé sera celui par défaut. Valider enfin en cliquant sur le bouton finish. Une fois créé,

le nouveau projet doit apparaître dans la fenêtre « Package Explorer ». Ainsi, tous les exercices d'un sujet seront regroupés dans un même projet.

**Chacun exercice fera l'objet d'un *package*.** Avant de commencer un exercice il faudra donc créer un nouveau package (File/New/Package) que l'on nommera `exo00` pour ce petit test.

Le package contiendra tous les fichiers nécessaires à un exercice. Nous n'aurons pour l'instant qu'un seul fichier par package (celui contenant le programme principal) mais cela ne sera bientôt plus le cas. Après s'être assuré que le package est bien sélectionné (il devrait être surligné en bleu), créer une classe (File/New/Class). Dans la fenêtre qui s'ouvre, donner le nom «Principal» à la classe et indiquer qu'elle contient le programme principal (appelé méthode `main` en Java) en cochant la case « `public static void main(String[] args)` », cela sera déjà ça que l'on n'aura pas à écrire ! En principe, on obtient quelque chose qui ressemble à :



Il n'y a plus qu'à compléter la méthode `main` avec par exemple l'instruction :

```
System.out.println ("Youpi ! J'ai écrit mon 1er programme JAVA :-)") ;
```

Pour exécuter le programme il faut simplement cliquer sur le bouton représentant une flèche blanche dans un rond vert. Le déroulement de l'exécution apparaît alors dans la Console qui se trouve dans la partie basse de la fenêtre.

## Exercices

Les exercices 1, 2, 3, 4, 5, 7, 8, 11, 12 et 14 sont à faire en priorité.

Ceux qui se sentent vraiment à l'aise pourront ensuite s'amuser avec les exercices 6, 10, 9/13 et 15.

### exo01 - *Location de voitures*

Un loueur de voitures propose deux formules :

1. un forfait de 40€ par jour puis 0,5€ par kilomètre parcouru,
2. un forfait de 50€ par jour puis 0,3€ par kilomètre parcouru.

Après avoir créé un package `exo01`, écrire le programme qui

- demande le nombre de jours de location prévus et une prévision du kilométrage
- affiche la formule la plus avantageuse pour le client

### exo02 - *Calcul de prix TTC*

Un magasin commercialise des produits qu'elle vend par lot.

Après avoir créé un package `exo02`, écrire le programme qui

- demande la saisie du prix hors taxe et de la catégorie d'un produit (« alim » ou autre),
- demande le nombre d'articles présents dans un lot,
- affiche le prix TTC d'un seul produit et le prix TTC du lot (le taux de TVA d'un produit de catégorie 1 est de 5,5%, le taux de TVA d'un autre type de produit est de 19,6%).

### exo03 - *Simple calculette*

Après avoir créé un package `exo03`, écrire un programme qui simule une calculette qui ne peut effectuer qu'une opération simple :

- Saisie de la première opérande réelle,
- Saisie du symbole de l'opération (+, -, \*, /),
- Saisie de la seconde opérande réelle,
- Affiche l'opération et son résultat.

### exo04 - *Tables de multiplication*

Après avoir créé un package `exo04`, écrire un programme qui affiche la table de multiplication (de 1 à 10) d'un entier donné par l'utilisateur.

### exo05 - *Division Euclidienne*

Après avoir créé un package `exo05`, écrire un programme qui permet d'obtenir le quotient et le reste d'une division euclidienne par la méthode des soustractions successives. Les deux opérandes entières seront saisies au clavier.

Par exemple  $13 \div 5$  se calcule :  $13 - 5 = 8$  ;  $8 - 5 = 3$ . 3 étant inférieur à 5, on en déduit que le quotient vaut 2 (on a fait 2 soustractions) et le reste vaut 3 (c'est le résultat de la dernière soustraction).

### exo06 - *Nombre mystérieux*

Le but de cet exercice est l'écriture d'un programme qui permet à l'utilisateur de deviner un nombre entier choisi au hasard par l'ordinateur.

Les instructions qui permettent le tirage aléatoire d'un entier appartenant à l'intervalle  $[0, n[$  sont dans la bibliothèque « `java.util` ». Ces instructions sont les suivantes :

```
Random generateur = new Random();  
r= generateur.nextInt(n);
```

### Version 1 : on est obligé de gagner !

Après avoir créé un package `exo06`, écrire un programme qui permet

- le tirage aléatoire d'un entier appartenant à l'intervalle  $[1, 50]$ ,
- permet à l'utilisateur de faire des propositions pour deviner ce nombre ; à chaque proposition le joueur sera informé que le nombre qu'il a donné est « trop grand » ou « trop petit »,
- à la fin du jeu, le nombre d'essais nécessaires pour trouver le nombre mystérieux est affiché.

### Version 2 : on peut perdre !

Modifier le programme pour n'autoriser qu'un maximum de 10 propositions. Dans le cas où le joueur ne découvre pas le nombre mystérieux au bout des 10 essais, la solution lui sera donnée.

### exo07 - *Pairs et impairs*

Après avoir créé un package `exo07`, écrire un programme qui

- permet de remplir un tableau avec 10 entiers saisis au clavier,
- affiche le nombre de nombres pairs et le nombre de nombre impairs.

### exo08 - *Bon élève*

Après avoir créé un package `exo08`, écrire un programme qui

- permet la saisie des notes obtenues par les 10 élèves d'une classe. Il faudra s'assurer que ces notes soient comprises entre 0 et 20 !
- affiche la moyenne de la classe ainsi que la meilleure note.

### exo09 - *Matrices*

Une matrice  $M_{n,m}$  peut-être modélisée par un tableau à deux dimensions :  $n$  lignes et  $m$  colonnes.

La déclaration d'un tableau à 2 dimensions s'écrit : `<type> <nomTab>[ ][ ]`;

La création du tableau spécifie chacune des dimensions : `<nomTab> = new <type>[n][m]`;

L'accès à la valeur stockée en  $(i, j)$  se note : `<nomTab>[i][j]`

Après avoir créé un package `exo09`, écrire un programme qui :

- demande à l'utilisateur la taille de la matrice (nombre de lignes et de colonnes), crée et initialise le tableau à 2 dimensions correspondant,
- permet la saisie d'un réel qui sera multiplié à la matrice ; le résultat sera rangé dans une nouvelle matrice qui sera ensuite affichée,
- crée la transposée de la matrice dans une nouvelle matrice, puis l'affiche.

### exo10 - *Fréquence*

On effectue un tirage aléatoire de 100 entiers appartenant à l'intervalle  $[1, 10]$ .

Après avoir créé un package `exo10`, écrire le programme qui effectue ce tirage puis affiche la fréquence de sortie de chacun des 10 entiers de l'intervalle.

**Indication** : On utilisera un tableau de 10 cases, la première case servira à comptabiliser le nombre de fois où le chiffre 1 est sorti, la seconde case contiendra le nombre de sorties du chiffre 2, etc.

### exo11 - *Tables de multiplication - version 2*

Après avoir créé un package `exo11`, copier et modifier le code écrit à l'exercice `exo04` pour :

- Écrire la méthode de classe, `multiplication`, qui affiche la table de multiplication (de 1 à 10) pour un entier dont la valeur sera donnée en paramètre. Écrire également la méthode `main` qui permettra de tester la méthode `multiplication`.
- Modifier la méthode `main` pour qu'elle affiche la table de multiplication de tous les entiers compris entre 2 et 9.

### exo12 - *Calcul de prix TTC - version 2*

Après avoir créé un package `exo12`, copier puis modifier le code écrit à l'exercice `exo02` pour :

- Écrire la méthode de classe, `calculTVA`, qui calcule (mais n'affiche pas) le montant de la TVA d'un produit dont le prix hors taxe et la catégorie seront donnés en paramètre. Penser à écrire la méthode `main` pour tester la méthode `calculTVA`.
- Modifier la méthode `main` pour que l'utilisateur puisse calculer le taux de TVA global pour plusieurs produits (une boucle lui posera une question pour savoir s'il veut effectuer un autre calcul).

### exo13 - *Matrices - version 2*

Après avoir créé un package `exo13`, copier le code écrit pour l'exercice sur les matrices (`exo09`) puis le modifier pour :

- Écrire la méthode de classe, `affMatrice`, qui affiche le contenu d'une matrice.
- Modifier la méthode `main` pour que les affichages soient maintenant effectués grâce à des appels à la méthode `affMatrice`.

### exo14 – *Moyennes*

Après avoir créé un package `exo14`

- Écrire la méthode `affTab` qui affiche le contenu d'un tableau d'entiers passé en paramètre,
- Écrire dans la méthode `main`, le code permettant de déclarer puis de remplir un tableau d'entiers. Le contenu de ce tableau sera ensuite affiché grâce à un appel à `affTab`,
- Surcharger la méthode `affTab`, pour qu'elle puisse être appliquée à un tableau de réels,
- Compléter la méthode `main` pour tester la surcharge de la méthode `affTab`.

### exo15 - *Palindrome*

Un palindrome est un texte ou un mot dans lequel l'ordre des lettres reste inchangé qu'on le lise de gauche à droite ou de droite à gauche. Par exemple : « Esope reste ici et se repose ». Nous allons considérer dans cet exercice, uniquement des palindromes composés d'un unique mot (pas d'espace) comme, par exemple, les mots : « laval », « kayak » ou encore « ABBA ».

Après avoir créé un package `exo15`

- Écrire une méthode, `palindrome`, qui admet un tableau de caractères en paramètre, et qui retourne si le contenu du tableau représente un palindrome ou pas.
- Écrire également un `main` qui teste la méthode `palindrome` avec un mot donné par l'utilisateur.

**Indication** : la méthode `toCharArray()` transforme la chaîne de caractères à laquelle elle est appliquée en tableau de caractères. Exemple d'utilisation : `tabCaracteres = laChaine.toCharArray()` ;