

# NFA035 : TP 1

## Prise en main d'Eclipse

21 février 2023

### Préliminaires

1. pour la FOD : assurez vous qu'une version récente du *Java Development Kit* ou JDK (site [www.oracle.com](http://www.oracle.com)) est installé dans votre machine. Dans le cas contraire, procédez à son installation.
2. pour la FOD : Installez également l'environnement de développement Eclipse (<http://www.eclipse.org/downloads/>).

### Prise en main d'Eclipse

#### Création d'un nouveau Projet

Nous allons créer un nouveau projet pour nos manipulations.

1. Cliquez sur File ⇒ New ⇒ Java Project.
2. Dans la première case, donnez un nom à votre projet, par exemple Tp1.
3. Normalement, le bon JRE est sélectionné ;
4. **décochez** éventuellement *create module-info.java*
5. Cochez la case "create separate folders for source and class files" qui vous permet d'avoir un répertoire src pour vos sources et bin pour les fichiers compilés.
6. Cliquez sur *finish*.
7. Votre projet est créé.

Nous allons observer ce qu'Eclipse a créé sur le disque pour que vous puissiez l'utiliser.

1. sur votre projet, cliquez-droit, et sélectionnez *show in/system explorer*.
2. Le répertoire bin contiendra tous les fichiers compilés
3. Le répertoire src contient toutes vos sources
4. demandez l'affichage des fichiers cachés
5. Les autre répertoires, dont les noms commencent par un "." sont des répertoires utilisés par eclipse pour gérer le projet.

## Ajout de sources existants au projet

Nous allons ajouter un fichier source ".java" au projet en cours. Par exemple, vous souhaitez peut être utiliser la classe Terminal.

1. Aller télécharger la classe Terminal.java (dans la sections outils du site de NFA031 par exemple)
2. faites "copier" sur le fichier
3. Sous eclipse, faites "coller" dans les sources de votre projet
4. Ca y est, la classe Terminal fait partie du projet, dans le package par défaut.

## Création d'une nouvelle classe

Pour créer une nouvelle classe, Eclipse dispose d'une interface d'aide.

1. Clic droit sur votre projet ⇒ new ⇒ class.
2. Entrez un nom de classe (qui sera aussi le nom du fichier, norme java) : `Salutation`
3. Si on ne met aucun nom de package, la classe sera dans le package par défaut (c'est très bien comme ça pour le moment)
4. Superclass : par défaut, toute classe hérite de `java.lang.Object`. Quand vous en saurez plus sur l'héritage, vous manipulerez ce champ. Pour le moment, on ne touche a rien
5. Remarquez la case a cocher `public static void main (String[])`. Si vous la cochez, la méthode main sera ajoutée automatiquement à votre classe. Vous pouvez le cocher si pour le moment, aucune des classes du projet ne contient de méthode main.
6. Cliquez sur finish, votre classe est créée.

Modifiez le code du `main` pour qu'il demande votre nom (en utilisant la classe `Terminal`) et qu'il imprime ensuite "bonjour NOM" (en saluant l'utilisateur avec le nom qu'il a saisi).

## Exécuter le projet

Tout en haut se trouve un bouton vert avec le symbole "play". Ce bouton permet de lancer votre projet, c'est à dire de lancer la méthode main. Si il en existe plusieurs, Eclipse vous proposera de choisir.

## Packages

Comme on prévoit de réaliser de très gros programme, on désire placer les différentes classes dans des packages différents.

- Créez un package nommé `nfa035.application`. Déplacez-y la classe `Salutation`.
  - Observer la première ligne du fichier `Salutation.java`;
  - Pourquoi votre fichier ne veut-il plus compiler ?
- Déplacez la classe `Terminal` dans un package nommé `nfa035.utils`.
- Regardez les modifications dans la classe `Terminal`, puis dans la classe `Salutation`;
- remplacez la ligne

```
import nfa035.utils.Terminal;
```

par

```
import nfa035.utils.*;
```
- supprimer la ligne `import nfa035.utils.*` pour voir ce qui se passe.

## Un peu de pratique

Créez dans le package `utils` une classe `TableauUtils` qui contiendra des méthodes pour travailler sur les tableaux. Écrivez les méthodes suivantes :

```
public static int[] saisirTableauInt()
```

qui demande la taille d'un tableau, permet d'en saisir le contenu, et le retourne.

```
public static void afficheTableauInt(int [] tab)
```

qui affiche un tableau d'entiers.

Écrivez dans le package `nfa035.application` un petit programme qui utilise ces deux méthodes pour saisir un tableau d'int et l'afficher.

Supprimez le mot `public` devant le nom des méthodes dans `TableauUtils`. Que se passe-t-il ?

## Exercice 2 : debug sous Eclipse

Eclipse possède un outil puissant que de nombreux autres *IDE* possèdent et que vous n'avez peut-être jamais utilisé pour le moment : un *debugger*. Cet outil permet de suivre pas à pas l'exécution du code et d'observer la valeur de n'importe quelle variable. De cette manière, on comprend mieux pourquoi le code ne fonctionne pas.

### Question 1

Téléchargez et installez le projet `nfa035_tp1.zip`. Il s'agit d'un projet **Maven** — un système qui peut fonctionner avec *tous* les IDEs. Vérifiez qu'il est correctement importé dans Eclipse. Regardez éventuellement ce lien pour voir comment importer votre projet.

### Question 2

La méthode `Multiplication.multiplier` est supposée calculer le produit de deux entiers. Elle ne fonctionne pas.

1. placez un point d'arrêt sur la ligne `int r = 0;`
2. lancez le code en mode debug;
3. regardez les valeurs des variables;
4. utilisez la pile d'appel pour remonter d'un cran et regarder les valeurs des variables dans `lancer()` et dans le `main`;
5. Faites tourner la méthode en mode *pas à pas* et essayez de comprendre ce qui se passe.

### Question 3

La classe `TriBugge` est supposée faire un tri insertion. Elle comporte deux bugs, un très simple et l'autre plus subtil. À vous de les corriger !