

NFA032 : TP 1

Prise en main d'Eclipse

10 février 2019

1 Préliminaires

2 Prise en main d'Eclipse

Création d'un nouveau Projet pour ce Tp

Nous allons créer un nouveau projet.

1. Cliquez sur File ⇒ New ⇒ Java Project.
2. Dans la première case, donnez un nom à votre projet, par exemple Tp1.
3. 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.
4. Cliquez sur Finish.
5. Votre projet est créé.

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.
5. Avant de passer à la suite, notez le contenu de la 1ère ligne du fichier `Terminal.java`

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, mais après il faudra créer vos propres paquetages)

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 à rien
5. Remarquez la case à 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` » (à peu près de cette forme ). Ce bouton permet de lancer votre projet, c'est à dire de lancer la méthode `main`. S'il en existe plusieurs, Eclipse vous proposera de choisir.

Packages

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

1. Créez un package nommé `nfa032.application`. Déplacez-y la classe `Salutation`.
 - Observer la première ligne du fichier `Salutation.java`;
 - Pourquoi votre fichier ne veut-il plus compiler ?
2. Déplacez la classe `Terminal` dans un nouveau package nommé `nfa032.utils`.
3. Regardez les modifications dans la classe `Terminal`, puis dans la classe `Salutation`;
4. Remplacez la ligne

```
import nfa032.utils.Terminal;  
  
par  
  
import nfa032.utils.*;
```

Cela compile toujours ?

5. Supprimez la ligne `import nfa032.utils.*` pour voir ce qui se passe. Expliquez ce qui arrive.

3 A faire dans ce Tp

Exercice 1

1. Dans le `nfa032.application` créez une classe `ArrayListTest` avec une méthode `main` qui doit déclarer et créer un `arraylist` (vide) de `String` :

```
ArrayList<String> liste = new ArrayList<String>();
```

Vous devez importer la classe `ArrayList` de la bibliothèque standard (`java.util`). Eclipse vous propose de le faire à votre place. Cliquez sur la croix rouge en première colonne de la ligne où vous utilisez un `ArrayList`.



```
package nfa032.application;

public class ArrayListTest {

    public static void main(String[] args) {
        ArrayList<String> liste = new ArrayList<String>();
    }

}
```

Un menu déroulant s'affiche avec en 1ère option : « import java.util.ArrayList; » Choisissez cette option. Cela ajoute (après la déclaration du package), la clause `import java.util.ArrayList;` dans votre fichier.

A l'aide de la javadoc et des supports de cours (NFA031) sur les arraylists, ajoutez dans cette liste 5 chaînes que vous lirez avec une boucle. Affichez ensuite ces 5 chaînes séparées par une virgule.

2. Dans le package `nfa032.utils` créez une classe `ArrayListUtils` qui contiendra des méthodes pour travailler sur les `ArrayList`. Écrire les méthodes suivantes :
 - `public static ArrayList<String> saisirArrayListString()` qui demande un entier N, puis lit N `String` à placer dans un `ArrayList`, et le retourne.
 - `public static void afficheArrayListString(ArrayList<String> a)` qui affiche les chaînes d'un `ArrayList` séparées par un espace.
3. Écrivez dans le package `nfa032.application` un petit programme qui utilise ces deux méthodes pour saisir un `arraylist` de chaînes et l'afficher.
4. Une fois vos méthodes écrites et compilées, supprimez le mot `public` devant le nom des méthodes dans `ArrayListUtils`. Que se passe-t-il ?

Exercice 2

Dans cet exercice vous allez utiliser les `arraylists` pour enregistrer/modifier les notes d'un élève sur plusieurs matières et pour afficher son bulletin. Une matière est caractérisé par son code sous forme de chaîne, et une note est un nombre à virgule. On pourra ajouter une nouvelle matière à celles d'un élève (mais sans l'ajouter plusieurs fois), obtenir la note associé à une matière, ou changer une note étant donner le code de la matière. On pourra également supprimer une matière (et sa note), obtenir la moyenne de toutes les notes, et afficher le bulletin de l'élève. Vous devez écrire les méthodes nécessaires et un programme `main` suffisamment simples et clairs.