

# TD N°1 : Classes internes en Java

Philippe Lahire

## Objectif

Expérimenter l'usage des différentes sortes de classes imbriquées de Java.

## Exercice 1 : Expérimenter les différentes alternatives de classes « imbriquées »

Plusieurs classes de la bibliothèque, notamment quand elles ont un paramètre générique, proposent des méthodes ayant des paramètres formels dont le type est représenté par une interface qui s'appuie elle aussi sur le même paramètre générique. C'est le cas de la classe *Arrays* (interface *Comparator*) vue en cours, et de la classe *ArrayList* (interface *Consumer*) qui fait l'objet de cet exercice.

On se propose de gérer une liste de modules d'enseignement (UE) par l'intermédiaire de la classe *ArrayList*. Un module d'enseignement est décrit par les propriétés suivantes :

- Nom du module (chaîne de caractères)
- Année de création (nombre)
- Nom du diplôme (chaîne de caractères)
- Nom de l'enseignant responsable du module (chaîne de caractères)
- Type de contrôle : QCM, Projet, questions de synthèse
- L'existence éventuelle d'une session de rattrapage : booléen

Il est demandé d'initialiser une instance de *ArrayList* avec une dizaine de modules et d'exécuter trois traitements différents (et indépendants), en utilisant la fonction *forEach*. Naturellement l'implémentation devra utiliser les classes internes. Lors de la description des traitements vous mettrez en œuvre une « trace » sur la console (*System.out*). Les trois traitements demandés sont :

- Incrémenter d'une année la propriété « Année de création », pour l'ensemble des modules.
- Incrémenter d'une année la propriété « Année de création » lorsque l'enseignant est « Dupont ».
- Supprimer le rattrapage éventuellement prévu si le type de contrôle associé est un QCM.

Ecrire le code qui permet d'expérimenter pour chacun de ces trois traitements, une implémentation qui repose successivement sur les techniques suivantes (5 implémentations sont donc à réaliser) :

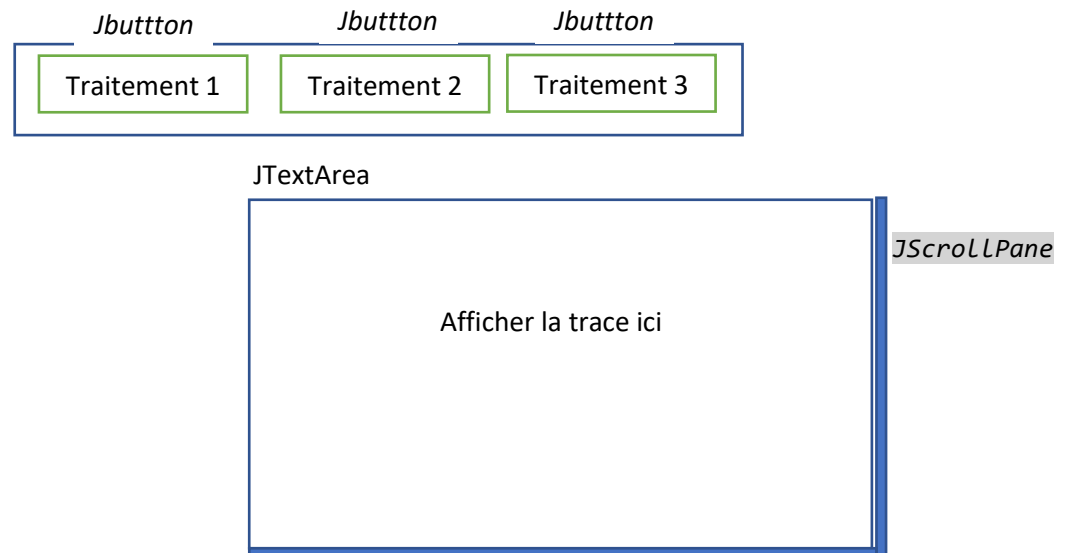
- Membre statique
- Membre non statique
- Classe locale
- Classe Anonyme
- Lambda

Que pouvez-vous dire de l'influence de ces choix par rapport à l'emplacement des traitements ? Donnez chaque fois les avantages et les inconvénients de la solution.

## Exercice 2 : Accès au traitement par interface graphique

### Exercice 2a

Utiliser les classes *JTextArea*, *Jbutton*, *JDialog* et *JScrollPane* afin de déclencher l'exécution de chacun des trois traitements par un bouton différent. Les messages initialement affichés sur la console le seront dans un objet de type *JTextArea*.



Vous ferez l'hypothèse que deux instances de l'interface graphique ne partagent pas la même zone de texte (*JTextArea*).

### Exercice 2b

Revisiter l'exercice 2a afin de l'adapter à un nouveau fonctionnement : la zone de texte est partagée par toutes les instances d'interface graphique, Pour cela vous devrez donc choisir un type de « classe imbriquée » différent.

