

# TD N°5 : TD Noté

Philippe Lahire

## A lire

Au plus tard à l'issue du contrôle, ou à tout moment pendant la durée de celui-ci, vous pourrez déposer la réponse à chacun des exercices. Chaque réponse pourra contenir plusieurs fichiers (dans ce cas faire une archive). Le nom du fichier commencera par le numéro de la question et sera suivi de votre nom et prénom. Exemple: 1-votre-nom votre-prénom, 2- votre-nom votre-prénom... 4- votre-nom votre-prénom. Lorsque l'heure sera dépassée vous ne pourrez plus déposer de fichier dans le devoir.

On met à votre disposition le squelette des classes *Personne*, *PersonneVaccinee* et *Vaccin*.

## Exercice 1 (5 points)

D'abord implémenter la méthode *ajusterEfficacite()* de la classe *PersonneVaccinee* qui baisse à « Faible » l'efficacité des différents vaccins réalisés si la personne a plus de 65 ans et augmente à « Forte » celle-ci lorsque la personne vaccinée a moins de 40 ans.

Vous utiliserez la méthode *forEach(...)* de la classe *java.util.ArrayList* et implémenterez le traitement ci-dessus à l'aide d'une classe interne anonyme.

Ensuite, créer une classe *ContainerDePersonnes*. Cette classe contiendra un tableau (*java.util.ArrayList*) *personnes* des personnes vaccinées qui devra être initialisé avec quelques instances de la classe *PersonneVaccinee*.

Enfin, créer dans cette classe un méthode « *main* » qui permettra de tester l'utilisation de la méthode *ajusterEfficacite()* sur l'ensemble des personnes du tableau. Utiliser obligatoirement une approche similaire à celle employée pour implémenter la méthode *ajusterEfficacite()* mais en choisissant une lambda en remplacement d'une classe anonyme.

## Exercice 2 (4 points)

Créer une annotation répérable *@UseMethod* qui n'est utilisable que pour les méthodes et qui pour une méthode donnée permette d'indiquer le nom et le nombre de paramètres des méthodes qu'elle utilise. Cette annotation doit être accessible pour l'introspection.

Exemple :

```
@UseMethod (nomMethod= «getAge », nombreParametres = 0)

@UseMethod (nomMethod= « forEach », nombreParametres = 1)

public void ajusterEfficacite() {...}
```

Vous annoterez ainsi toutes les méthodes de la classe **PersonneVaccinee** en utilisant **UseMethod** comme ci-dessus. L'annotation pourra apparaître 0, 1 ou plusieurs fois selon le cas (pour les besoins du test vous pourrez créer des méthodes supplémentaires juste dans le but de les annoter).

### Exercice 3 (5 points)

Dans une classe **PersonneVaccineeIntrospection**, écrire la méthode « *String[] methodesAppellees(int nombre)* » qui retourne un tableau contenant le nom des méthodes de la classe **PersonneVaccinee** qui appellent plus de *nombre* méthodes.

Vous pourrez faire deux versions de cette méthode :

- Une qui utilise l'annotation `UseMethod` (voir exemple ci-dessus)
- Une autre qui utilise l'annotation **UseMethods** (voir exemple ci-dessous), induite par le caractère répétable de l'annotation **UseMethod**. Le nom **UseMethods** sera celui mentionné lors de la définition de la répétabilité.

Exemple :

```
@UseMethods({@UseMethod (nomMethod= «getAge », nombreParametres = 0), @UseMethod
(nomMethod= « forEach », nombreParametres = 1) }

public void ajusterEfficacite() {...}
```

La classe **PersonneVaccineeIntrospection** devra contenir une méthode « main » permettant de tester la méthode *methodesAppellees*.

### Exercice 4 (6 points)

Ajouter à la classe **ContainerDePersonnes** créée dans l'exercice 1 deux méthodes :

- une méthode *sauverPersonnes(String nomfichier)* qui sérialisera le tableau *personnes* dans un fichier dont le nom est passé en paramètre.
- une méthode *restaurerPersonnes* qui récupérera les objets se trouvant dans le fichier dont le nom est passé en paramètre et affichera les personnes (utiliser la méthode *display* de la classe *PersonneVaccinee*).

Modifier la méthode « main » pour appeler ces deux méthodes. Vous donnerez comme nom de fichier « resultatTD ».

Configuration des objets sauvegardés :

- Une instance de la classe *Vaccin* doit pouvoir être sauvegardée mais il est fait le choix que Le nom du vaccin n'est lui pas sauvegardé
- On ne peut pas sauver des instances de la classe *Personne* mais on veut quand même sauvegarder tous les champs contenus dans la classe *Personne* lorsque l'on sauvegarde une instance de la classe *PersonneVaccinee*.

Modifier s'il y a lieu les classes *Personne*, *PersonneVaccinee*, *Vaccin* pour permettre la mise en œuvre de cette configuration.

### Exercice supplémentaire (bonus de 2 points)

Même chose que l'exercice 4 mais avec la configuration suivante des objets sauvegardés :

- Une instance de la classe *Vaccin* doit pouvoir être sauvegardée mais il est fait le choix que Le nom du vaccin n'est lui pas sauvegardé (aucun changement avec la 1ere configuration)

- On peut sauver des instances de la classe `Personne` et on veut sauvegarder tous les champs contenus dans la classe `Personne` et la classe `PersonneVaccinee` lorsque l'on sauvegarde une instance de la classe *PersonneVaccinee*.

Les modifications pourront être faites dans le fichier (par exemple en faisant une recopie entière du source de la classe à la fin du fichier et en la mettant en commentaire)