

TP n°3 : NFA035 – Bibliothèques et patterns

V. Aponte

22 février 2018

Création, inspection, utilisation d'objets pour des classes fournies via debugueur Eclipse. Utilisation d'interfaces.

Préparation de l'environnement

Importez les sources du projet tp3. Une erreur sera signalée dans le package tp3.exo2, nous nous en occuperons plus tard.

Exercice 1 : exploration d'objets via le *debugueur* d'Eclipse

L'objectif de cet exercice est d'utiliser des classes fournies et prêtes à l'emploi pour créer des objets à partir de de classes fournies, inspecter leur contenu, et d'explorer leur fonctionnement. Nous ferons cela à l'aide du *debugueur* d'Eclipse.

1. Ouvrez une fenêtre avec les transparents du premier cours sur Eclipse et placez vous dans la partie dédiée au débogueur. Cela pourra vous servir.
2. Assurez vous d'avoir bien recopié tous les sources dans le paquetage exo1, et qu'il n'y ait pas d'erreur de compilation. En particulier tous les fichiers doivent débiter par `package tp3.exo1;`
3. Sélectionnez la classe `TesteComptes`. Vous placerez dans ce fichier un point d'arrêt sur toutes les lignes signalées par un commentaire.
4. Dans le menu *Run*, sélectionnez *Debug as ... Java Application*. Une vue débogueur doit s'afficher avec la pile d'exécution, une vue du contenu des variables, la console pour les sorties, etc.
5. L'action précédente a du vous placer sur le premier point d'arrêt. Dans la vue où sont listées les variables du programme, cliquez sur la petite flèche permettant d'inspecter le contenu de chacune d'entre elles.
6. Utilisez le bouton *resume* (flèche bleu) afin d'avancer jusqu'au prochain point d'arrêt. Notez que certaines variables sont surlignées lorsque leurs valeurs changent. Explorez à chaque fois toutes les variables.
7. Répétez jusqu'à finir l'exécution de `TesteComptes`.
8. Appliquez ces mêmes pas sur le fichier `TesteClient`.

Exercice 2 : utiliser des interfaces

Récupérez sur le site du cours les sources pour l'exercice 2 et incorporez-les dans votre projet eclipse. Vous trouverez dedans une interface `AvecNote` et plusieurs classes.

L'interface possède des méthode pour obtenir le code d'un module d'enseignement et pour calculer de la note du module. Il n'y a aucune variable, pas d'implantation des méthodes.

Question 1

La classe `Prog035` produit une erreur à la compilation. Pourquoi ? Corrigez cette erreur.

Question 2

Déclarez ensuite une classe de démonstration avec une méthode `main` qui effectue les actions suivantes :

1. Déclarez un tableau capable de contenir 4 objets de type `AvecNote`. Initialisez ce tableau avec des objets instance de `Prog032` dans les case 0 et 2, et de `Prog035` dans les case 1 et 3 du tableau.
2. Ajoutez dans votre `main` l'instruction `String s1 = UEs[0].getSession1();` Expliquez l'erreur signalée par le compilateur. Enlevez ensuite cette ligne de votre programme `main`.
3. Ecrivez une méthode statique `moyenneTabNotes` qui prend en argument un tableau d'objets de type `AvecNote` et qui calcule la moyenne des notes pour ses composantes.
4. Utilisez cette méthode pour calculer et afficher la moyenne des notes pour les UEs dans le tableau.
5. Ajoutez une méthode `afficher` dans la classe `Prog032` : elle doit utiliser la méthode `toString`.
6. Ajoutez l'interface suivante dans votre projet :

```
interface Affichable{
    void afficher();
}
```

Quelle est la classe qui implante cette interface ? Ajoutez cette interface dans sa clause `implements`.

7. Peut-on écrire `Affichable a = new Prog035(12, 6, 10.5);`. Pourquoi ?

Question 3

Etudiez la classe `Eleve`. Expliquez comment elle utilise l'interface `AvecNote`. Un élève pourra-t-il ajouter parmi ses UES des objets de classes `Prog032` et `Prog035` ? Ajoutez dans votre programme `main` un élève `Bob`. Ajoutez des ues `Prog032` et `Prog035` pour `Bob`.

Question 4

Ecrivez une classe `Prog031` qui implante `AvecNote` et ajoutez une ue de ce type pour `Bob`.

Question 5

On voudrait ajouter une méthode qui permet d'afficher toutes les UES d'un élève. Que faut-il modifier d'après vous ?