

Compte Rendu du TD numéro 3 de l'équipe Eirb'reteau

Pour le 29 septembre

Coordinateur : Reda Boudjeltia
Tandem 1 : Pierre Gaulon, Lionel Adotevi
Tandem 2 : Victor Dury, Aurélien Nizet
TD du 29 septembre 2014

1 Boutez vos neurones (partie 4 du TD3)

Tout d'abord, nous avons commencé par modifier les trois états DEHORS, ASSIS et DEBOUT. Pour les partager, nous en avons fait des variables d'instance (static), et constantes (final), pour qu'elles ne soient plus modifiées. Ces trois variables de la classe **EtatPassager** sont des instances de la classe **EtatPassager** elle-même, instanciées par le constructeur par défaut, lors de la compilation de cette classe. En réalité, ce qui différencie ces trois états, c'est simplement leur référence, leur adresse mémoire qui est différente. Il n'y a plus d'état courant. On a donc supprimé les anciens constructeurs, et toutes les références à l'état courant. Pour toutes les méthodes d'accès de cette classe, on compare la référence de la classe utilisée (this), avec les références DEBOUT, ASSIS, DEHORS. Par exemple `this==DEHORS` va caractériser l'état *estExterieur()* d'un `PassagerStandard`. De même pour les autres méthodes d'accès d'état. Pour changer d'état, il suffit simplement de renvoyer l'adresse de l'état ciblé. Par exemple, pour faire assoir un **PassagerStandard**, il suffit de faire *returnASSIS*;. La méthode *toString()* a aussi dû être changée. En effet, le switch case n'était pas compatible avec des comparaisons de références, on l'a alors changé en succession de *if*. Enfin, comme nous n'avions plus de constructeur de **EtatPassager**, mais qu'il faut bien initialiser un état par défaut lors de la construction d'un **PassagerStandard**, nous avons créé une nouvelle méthode de classe static public `EtatPassager construct()`, qui renvoie la référence de DEHORS. Ainsi, il est possible d'initialiser un **EtatPassager**.

2 Recette

Commande pour lancer les tests unitaires dans le répertoire `tstPassager` :

```
java -ea -cp build-tstPassager/:build/ tec.LancerTests
```

Commande pour lancer les tests unitaires dans le répertoire `tstAutobus` :

```
java -ea -cp build-tstAutobus/:build/ tec.LancerTests
```

Il faut les exécuter séparément, car les tests d'Autobus dépendent du faussaire de **PassagerStandard**, et de la classe **Autobus**, et les tests de **PassagerStandard** dépendent du faussaire de **Autobus**, et de la classe **PassagerStandard**. Pour exécuter correctement les tests, il est donc nécessaire de ne pas mélanger les faussaires (notamment lors des tests de messages envoyés). Ces faussaires sont dans deux répertoires différents, et donc il faut séparer ces répertoires dans l'option *-classpath*, ce qui n'aurait pas été possible en une seule exécution.

Commande de compilation de la classe Simple :

```
javac -d build/ src/* && javac -d build/ -cp build/ recette/Simple.java
```

Commande d'exécution de la classe Simple :

```
java -cp build/ Simple
```