

## Programmation défensive et tests unitaires

### Préambule

Nous reprenons le TP du semestre dernier pour que vous ayez tous une expérience avec JUnit. Le but de ce TP est de se familiariser techniquement avec cet outil permettant de faire des tests unitaires en Java.

Les classes qu'on souhaite tester correspondent à une proposition de modèle pour un jeu de Memory dont le diagramme de classe est représenté en Figure 1

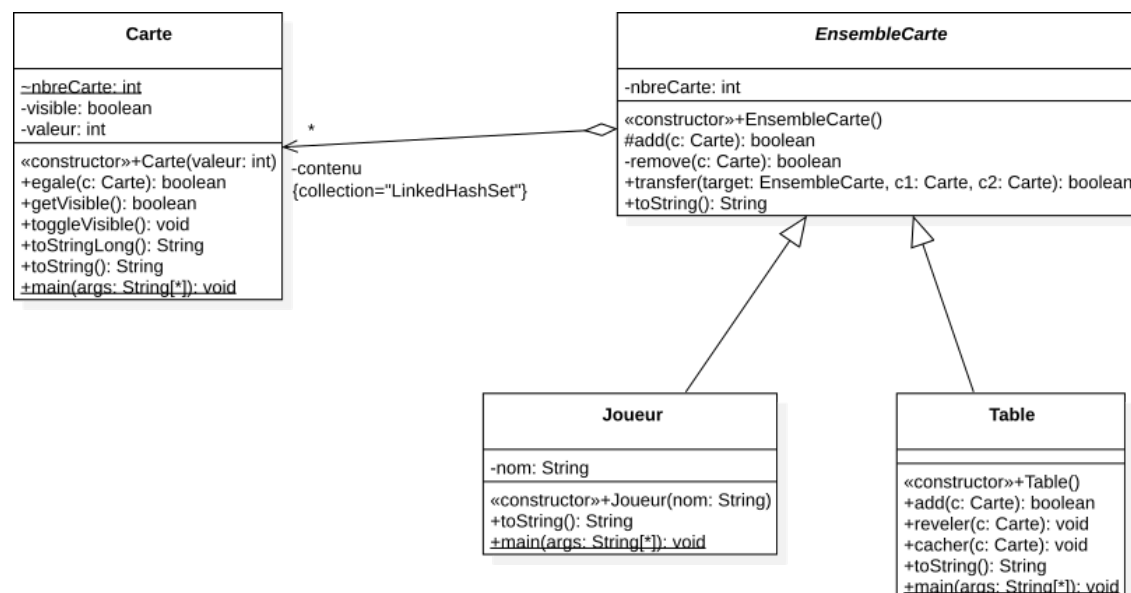


FIGURE 1 – Diagramme de Classe (outil reverse code de star UML)

**Exercice 1.** Examinez rapidement le diagramme de classe ci-dessus. Regardez le code `v0Memory.zip`. Compilez le. Lisez la documentation (vous pouvez générer la javadoc, ou bien lire les commentaires directement dans le code). Il y a un exemple de fichier JUnit pour la classe `Carte`. Vous avez aussi un main qui contient des tests à l'ancienne (avec des print).

### Programmation défensive

Nous allons mettre en œuvre le principe de programmation défensive. Pour rappel, il s'agit ici très concrètement de « bétonner » les méthodes publiques d'une classe pour qu'on ne puisse pas les utiliser en dehors de leur usage prévu.

Notez qu'il ne s'agit pas ici de voir l'utilisateur comme un dangereux criminel. En effet, le code que nous étudions dans ce TP fait partie du modèle (le M de MVC) et l'utilisateur est un développeur qui

utilisera nos classes (pour faire par exemple le V ou le C de MVC). Un développeur peut de toute façon faire ce qu'il veut en java avec nos classes (pour plus de détails, voir la réflexion et l'introspection en Java). Ce n'est pas donc dans un but de protection au sens usuel en cryptographie : il s'agit ici d'une bonne pratique de génie logiciel permettant à un développeur qui utilise notre classe de recevoir des messages d'erreurs adaptés si il utilise notre travail d'une manière qui n'est pas normale.

**Exercice 2.** Lisez la javadoc (dans le code ou en générant le html) de `Carte.java`. En particulier celui de `egale`.

À votre avis, quelle est la nuance entre `IllegalArgumentException` et `IllegalStateException` ?

## Découverte des tests unitaires avec Junit

N'oubliez pas de faire le set-up correct de votre classpath, sinon ce qui suis ne marchera pas. À l'IUT il suffit de mettre cette ligne dans votre `.bashrc`  
`export CLASSPATH=".:usr/share/java/junit.jar:/usr/share/java/hamcrest-core.jar"`

**Exercice 3.** Lancez les tests JUnit proposés (classe `TestCarte`) en faisant :

```
:$ java org.junit.runner.JUnitCore TestCarte
```

Des tests sont passés avec succès, d'autres non. Remédiez à la situation.

Bonne chasse !

## Pour aller plus loin

**Exercice 4.** Ajoutez des tests JUnit pour les autres classes. Trouvez tous les merveilleux bugs bien croquants que j'ai laissé partout (volontairement ou pas).