NFA035 – Bibliothèques et patterns : TP n°6 (2ème Collections)

V. Aponte

April 1, 2016

Exercice 1: for each et itérateurs

Question 1

On veut écrire une méthode statique `afficheColl(Collection<String> c)` permettant d'afficher n'importe quelle collection contenant des String, par exemple, un ArrayList de String, un HashSet de String, etc. Ecrivez **deux versions** de cette méthode: la première utilise une boucle for-each; la deuxième utilise un itérateur. Dans le main, testez votre méthode sur un ArrayListe et sur un HashSet que vous aurez défini et initialisé avec quelques éléments au préalable. N'oubliez pas d'importer `java.util.*` :

```
// Dans le main
ArrayList<String> ac = new ArrayList<String>();
HashSet<String> sc = new HashSet<String>();
// Ajout elements ...
afficheColl(ac);
afficheColl(sc);
```

Question 2

Voici une implantation de la méthode `afficheListeSt(ArrayList<String> c)` demandée dans le Tp précédent (exercice 1). Elle permet d'afficher le contenu d'un ArrayList de String. Est-il possible de reprendre le corps de cette méthode **tel quel** pour implanter la méthode `afficheColl(Collection<String> c)` de la question 1? Expliquez votre réponse.

```
// Cette methode est correcte (testez-la!!)
 static void afficheListeSt(ArrayList<String> c){
   for (int i=0; i< c.size(); i++){
       System.out.println(c.get(i));
   }
 }

// Cette implantation est-elle correcte?
 static void afficheCollNo(Collection<String> c){
    for (int i=0; i< c.size(); i++){
        System.out.println(c.get(i));
    }
 }
```

Exercice 2: jeu de cartes (partie 1)

Dans cet exercice nous allons implanter les classes nécessaires à un jeu de cartes. Dans le site du cours, récuperez les sources pour Tp. Vous trouverez 3 classes: `Carte`, `TestCarte0` et un type *énuméré* (classe `Couleur`).

Question 1: type énuméré Couleur

Un type énuméré est formé d'un ensemble de mots considerés comme des constantes. Dans le type `enum Couleur` chacun de ces mots est de type `Couleur`. Avant de porsuivre: allez voir comment utiliser les types énumérés dans la documentation et tutoriels java (site oracle). Testez `Couleur` avec le programme `TestCouleur0`. Notez également comment ce type est utilisé dans la classe `Carte`.

Question 2 : cartes

Testez la classe `Carte` avec le programme `TestCarte0`. Expliquez les affichages obtenus.

Question 3 : `hashCode()` et `equals()`

Ajoutez des méthodes `hashCode()` et `equals()` correctes dans la classe `Carte`. Vous pouvez demander à Eclipse de les générer pour vous. Testez le bon fonctionnement de ces deux méthodes:

1. recommencez l'exécution du programme `TestCarte0`.
2. toujours dans ce programme, utilisez un `HashSet` au lieu d'un arraylist et testez à nouveau (en y ajoutant des doublons).

Question 4 : MainJoueur1

Ecrivez une classe `MainJoueur1` afin de représenter la main d'un joueur, contenant plusieurs cartes. Utilisez un `HashSet` pour stocker les cartes dans la main. Dans un premier temps on souhaite doter cette classe des méthodes suivantes:

- `addCarte(Carte)`: ajoute une carte à la main.
- `addCarte(Carte)`: teste si la main contient une carte.
- `toString()`: renvoie une représentation de la main sous forme de chaîne.

Vérifiez le bon fonctionnement de votre classe avec un petit programme qui l'utilise.

Question 5 : itération sur MainJoueur1

On veut disposer d'un moyen simple de parcourir les cartes d'une `MainJouer1`. Pour cela, modifiez cette classe de sorte qu'elle implante l'interface `Iterable<Carte>`. Testez le résultat en écrivant dans un programme de test :

```
public static void main(String[] args) {
        MainJoueur1 m = new MainJoueur1();

        m.add(new Carte(10, Couleur.Carreau));
        m.add(new Carte(1, Couleur.Coeur));
        m.add(new Carte(10, Couleur.Trefle));

        if (m.contient(new Carte(10, Couleur.Carreau))) {
                System.out.println("Le␣jeu␣contient␣le␣10␣de␣carreau");
        } else {
                System.out.println("Le␣jeu␣ne␣contient␣pas␣le␣10␣de␣carreau");
        }
        // Test pour la question 7: itération sur une MinJoeur1
        for( Carte c: m){
                System.out.println(c.toString());
        }
}
```