

# Exercices sur les Collections

## Exercice 1

### Question 1

On considère la classe Carte :

```
package cartes;
```

```
public class Carte {
```

```
    // NOTE : commencez par modifier ce code pour utiliser un enum afin de représenter la  
    // couleur.
```

```
    public static final String PIQUE = "pique";  
    public static final String TREFLE = "trèfle";  
    public static final String COEUR = "coeur";  
    public static final String CARREAU = "carreau";  
    public static final String[] couleurs = { PIQUE, TREFLE, COEUR, CARREAU };
```

```
    private int valeur;  
    private String couleur;
```

```
    public Carte(int valeur, String couleur) {  
        this.valeur = valeur;  
        this.couleur = couleur;  
    }
```

```
    public int getValeur() {  
        return valeur;  
    }
```

```
    public String getCouleur() {  
        return couleur;  
    }
```

```
    @Override  
    public String toString() {  
        return "" + valeur + " de " + couleur;  
    }
```

```
}
```

Tester le programme suivant :

```

import java.util.ArrayList;

public class TestCartes0 {

    public static void main(String[] args) {
        ArrayList<Carte> cartes = new ArrayList<Carte>();

        cartes.add(new Carte(10, Carte.CARREAU));
        cartes.add(new Carte(1, Carte.COEUR));

        if (cartes.contains(new Carte(10, Carte.CARREAU))) {
            System.out.println("Le jeu contient le 10 de carreau");
        } else {
            System.out.println("Le jeu ne contient pas le 10 de carreau");
        }
    }
}

```

Quel résultat donne-t-il ? Pourquoi ?

## Question 2.

Ajouter à la classe Carte les méthodes equals et hashCode correctes. Tester la classe. Pour vérifier que hashCode fonctionne, remplacez l'ArrayList par un HashSet.

## Question 3. Classe MainDeJoueur1.

On veut représenter la main d'un joueur. Pour cela on écrit la classe MainJoueur1.

Une main de joueur contient un certain nombre de cartes. Pour l'instant, on désire la doter des méthodes :

- add(Carte) : ajoute une carte à la main.
- contient(Carte) : renvoie vrai si la main contient la carte en question
- toString() : renvoie une représentation de la main sous forme de String.

Écrire la classe MainDeJoueur1 en utilisant un HashSet pour implémenter la main.

Vérifier le bon fonctionnement de la classe :

- écrire un programme où vous créez une main, y ajoutez des cartes, et affichez le résultat.

## Question 4.

Classe Paquet de carte

On représente un paquet de cartes par la classe « Paquet ».

Quand on crée un paquet, il est complet (et a priori mélangé).

On dote la classe paquet d'une méthode toString et d'une méthode creerMainDeJoueur1.

Celle-ci tire 5 cartes du paquet (les 5 premières, par exemples), pour constituer une main, puis elle retourne la main.

Sa signature est :

```
public MainDeJoueur1 creerMain1()
```

Test: faire un main qui

crée un paquet  
affiche le paquet.  
Crée une main  
la visualise.

### Question 5. Test d'intégration.

Tester vos classes avec le programme suivant :

```
package cartes;

import java.util.Scanner;

public class TestPaquet {
    public static void main(String[] args) {
        Paquet paquet = new Paquet();
        MainJoueur1 maMain = paquet.creerMain1();
        System.out.println("mon jeu " + maMain);

        Scanner scanner = new Scanner(System.in);
        System.out.print("Entrez une valeur de carte :");
        int valeur = scanner.nextInt();
        System.out.print("Entrez une couleur de carte :");
        String couleur = scanner.next();

        Carte c = new Carte(valeur, couleur);
        if (maMain.contient(c) {
            System.out.println("La main contient la carte");
        } else
            System.out.println("La main ne contient pas la carte");

    }
}
```

### Question 6.

On veut pouvoir disposer d'un moyen simple de parcourir une « MainDeJoueur1 ». Pour cela : Implémentez l'interface Iterable<Carte> dans MainDeJoueur1.

Tester le résultat :

Il est normalement possible d'écrire :

```
for (Carte c: maindejoueur1) {
    ...
}
```

### Question 7.

Même chose, mais avec MainDeJoueur2, implémentée en utilisant un TreeSet de cartes. L'ordre des cartes sera valeur puis couleur.

### Question 8.

Dans mainDeJoueur2, implémenter la méthode « estSuite() », qui retourne vrai si la main est une suite de cartes.

**Question 9.**

On veut implémenter « `getNombreCouleurs` » (qui renvoie le nombre de couleurs qui apparaissent dans une main). Pour cela, on désire travailler sur un jeu trié par couleur.

En utilisant un `Comparator`, et en travaillant sur un `TreeSet` temporaire, écrire la méthode `getNombreCouleurs` dans la classe `MainDeJoueur`.