**TD : Enumération**

**Exercice 1**

Soit l’énumération Animal défini comme suite :

public enum Animal{chat, chien, cheval, poisson, girafe};

Dans la fonction main, tester les méthodes suivantes : toSTring(), valueOf(), values(), ordinal(), compareTo().

**Exercice 2 :**

Créez le type énuméré ArcEnCiel dont les valeurs sont les couleurs de l’arc-en-ciel : rouge, orange, jaune, vert, bleu, indigo, violet.

Dans la fonction main :

* Créez une référence de type ArcEnCiel ;
* Initialisez la référence de type ArcEnCiel, utilisez la fonction valueOf,
* Initialisez une référence avec la valeur qui suit dans le type énuméré (Exemple : violet suit de rouge);
* Affichez toutes les valeurs définies dans ce type énuméré.

Exercice 3

Définissez un type énuméré Jour (lundi, mardi, etc.) dont les valeurs disposent d’un attribut **estTravaille** qui indique si ce jour est ou non un jour travaillé, sa valeur est à false pour samedi et dimanche et à **true** pour les autres jours. Les valeurs de ce type disposent de deux méthodes, l’une, appelée **lendemain**, renvoie le jour suivant dans la semaine, l’autre, appelée **estTravaille**, permet de savoir si le jour est ou non travaillé.

Dans la fonction main

* À partir d’un nom de jour, affichez s’il s’agit un jour férié ou non.   
  Définissez une boucle qui permet d’afficher les valeurs de l'énumération JoursDeSemaine.
* Méthode 1 : utilisez la fonction values() ;
* Méthodes 2 : Utilisez un EnumSet ;
* Déclarez un jour de semaine : **jour** ;
* Pour chaque valeur définit dans l’énumération JoursDeSemaine, affichez :
* une valeur négative si la valeur est inférieur à la référence **jour ;**
* zéro signifie que la valeur est considéré comme égal à **jour**
* une valeur positive signifie que la valeur est supérieur à **jour**