

P00 – les énumérés

On parle de type énuméré lorsqu'une variable peut prendre un nombre déterminé de valeur connue. En Java, les énumérés sont des classes qui étendent `java.lang.Enum` et qui ont les mêmes visibilités que toute autre classe.

```
public enum Couleur {ROUGE, VERT, BLEU, BLANC};
```

Les valeurs sont des objets qui sont `public final static`.

Ils n'ont pas de constructeur public.

```
public class Dessin {  
    private Couleur couleur;  
    ...  
    public void setCouleur(Couleur c) {  
        couleur = c;  
    }  
}
```

```
Dessin d = new Dessin();  
d.setCouleur(Couleur.ROUGE);
```

On peut utiliser `==` et `equals` indifféremment car même si le premier porte sur l'adresse de l'objet et le second sur sa valeur, ils ne sont instanciés qu'une seule fois et sont donc constants.

Ils implémentent automatiquement `Comparable`. La méthode `compareTo` repose sur l'ordre dans lequel les constantes ont été énumérées. Dans notre cas, `Couleur.ROUGE.compareTo(Couleur.VERT)` sera négatif.

Les méthodes `toString` et `valueOf` permettent de convertir l'énuméré en chaîne de caractères et vis versa.

```
Couleur.ROUGE.toString() vaut "ROUGE"
```

```
Couleur.valueOf("ROUGE") vaut Couleur.ROUGE
```

La méthode de classe `values` permet d'itérer

```
for (Couleur c : Couleur.values()) {  
    System.out.println(c);  
}
```

Le `switch` peut se faire sur des énumérés :

```
switch (dessin.getCouleur()) {  
    case ROUGE:
```

```

        ...
        break;
case VERT:
case BLEU:
case BLANC:
        ...
        break;
}

```

Ou

```

switch (dessin.getCouleur()) {
case ROUGE:
        ...
        break;
default:
        ...
        break;
}

```

Les énumérés peuvent être définis comme classe interne.

```

public class Dessin {
public enum Couleur {ROUGE, VERT, ... };

```

C'est la même chose que

```

public class Dessin {
public static enum Couleur {ROUGE, VERT, ... };

```

On peut ajouter des méthodes à une classe énumérée

```

public enum CouleurRGB {
    // les valeurs en premier lieu
    ROUGE(255,0,0), VERT(0,255,0), BLEU(0,0,255),
    BLANC(255,255,255);
    // notez le ; qui termine les valeurs

    private int red, green, blue; // les champs après les valeurs
    CouleurRGB(int r, int g, int b) { // constructeur private
        red = r; green = g; blue = b;
    }
    public int getRed() { return red; }
    public int getGreen() { return green; }
    public int getBlue() { return blue; }
}

```