

## TP n°5 : Pattern Decorator

### Exercice 1 (Encapsuler un entier)

On va refaire la classe Integer, en partie. L'objectif va être de définir une classe Entier contenant un entier (normal, en décimal) et ses décorateurs pour convertir l'entier en binaire, octal et hexadécimal, avec un décorateur par base.

Pour cela, la classe Entier et ses décorateur contiendront une méthode `base()` affichant à l'écran la valeur de l'entier dans la base adéquate. Pour les décorateurs, l'appel à `base()` devra également appeler `base()` de l'objet décoré.

### Exercice 2 (Le concessionnaire automobile)

Faire sur machine l'exercice 2 du TD 5.

### Exercice 3 (Décorer InputStream)

Faire une classe ReverseInputStream décorant InputStream (conseil : dérivez de FilterInputStream) et proposant une méthode retournant la stream inversée octet par octet.

### Exercice 4 (Starbucks)

Faire sur machine l'exemple du cours, mais en prenant en compte la taille de la boisson (petite, moyenne et grande) qui va influencer le prix du café ET des suppléments de la manière suivante :

- la petite taille implique un facteur de 0.75, c'est-à-dire qu'on aura une réduction du quart du prix total normal.
- la taille normale implique un facteur de 1, donc pas de réduction ni de coût supplémentaire.
- la grande taille implique un facteur de 1.25, soit une augmentation des tarifs du quart du prix total.