



- Décorateur (voiture)
- Flyweight (expression mathématique)
- Composite (expression mathématique)

- L'affichage des caractéristiques du véhicule (liste des options)  
un véhicule sans option a comme attribut sa marque et son modèle.  
un véhicule avec option a comme attribut le libellé de l'option
- Le calcul du prix final du véhicule  
le prix d'un véhicule sans option est un attribut de la classe.  
le prix d'un véhicule avec option se calcule en ajoutant le prix de l'option au prix du véhicule.

- VehiculeAvecOptionToitOuvrant (prix : 1000€)
- VéhiculeAvecOptionPeintureMétalisée (prix : 800€)
- VéhiculeAvecOptionJantesAlu (prix : 600€)

- 2- Ecrire un main dans lequel vous créerez un véhicule Clio de marque Renault (10 000€), avec les trois options précédentes.  
Vous afficherez le prix final du véhicule, ainsi que la liste des options

## EXPRESSIONS MATHÉMATIQUES (*Composite et Flyweight*)

Le but de cet exercice est de dériver des expressions mathématiques.

Ces expressions seront composées :

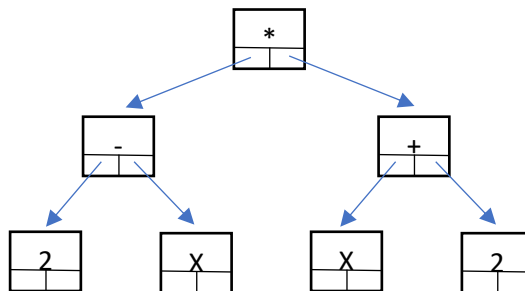
- D'opérateurs : +, -, \* et /
- De constantes numériques (de type réel)
- De la variable  $x$

Ex :  $(2 - x) * (x + 2)$

Le principe de base est la construction d'un arbre binaire composé de nœuds :

- Les opérations représentent les nœuds internes de l'arbre
- Les constantes numériques et les variables  $x$  constituent les feuilles de l'arbre

Ex : pour l'expression précédente :



Cette hiérarchie sera traduite par le design pattern **Composite**.

La création des nœuds sera assurée par la mise en place du design pattern **Factory**.

La classe Nœud sera pourvue des deux méthodes suivantes :

- `afficher()` : qui affiche l'expression issue du nœud (fonction récursive)
- `deriver()` : qui calcule l'expression dérivée, en suivant les règles de dérivation

- 1- Implémenter la hiérarchie de classes permettant de représenter une expression mathématique : *Nœud, Operateur, OperateurPlus, OperateurMoins, OperateurMult, OperateurDiv, OperandeConstante et OperandeX*
- 2- Implémenter la factory permettant de créer une expression mathématique
- 3- Ecrire un main qui
  - crée l'expression donnée plus haut en exemple
  - calcule l'expression dérivée
  - et l'affiche

Le design pattern **FlyWeight** peut être judicieusement utilisé :

- Pour garantir l'unicité du nœud  $x$  (utilisation du design pattern **Singleton**)
- Et faire en sorte que deux nœuds constantes fassent référence au même objet (utilisation d'un dictionnaire de nœuds constantes déjà créés)

- 4- Modifier la factory pour prendre en compte les design pattern FlyWeight et Singleton.