

1.5. Constructeur, accesseurs et mutateurs

Parmi les différentes méthodes que comporte une classe, on a souvent tendance à distinguer :

- les **constructeurs**, sans valeur de retour, porte un nom imposé par le langage Python : `__init__()`. Cette méthode sera appelée lors de la création de l'objet. Le constructeur peut disposer d'un nombre quelconque de paramètres, éventuellement aucun ;
- les **accesseurs** (en anglais *accessor*) ou "**getters**" qui fournissent des informations relatives à l'état d'un objet, c'est-à-dire aux valeurs de certains de ses attributs (généralement privés) sans les modifier ;
- les **mutateurs** (en anglais *mutator*) ou "**setters**" qui modifient l'état d'un objet, donc les valeurs de certains de ses attributs.

On rencontre souvent l'utilisation de noms de la forme `get_XXXX()` pour les accesseurs et `set_XXXX()` pour les mutateurs, y compris dans des programmes dans lesquels les noms de variable sont francisés.

2. Utilisation d'objet

On suppose écrite la classe `Carte` dont on vous donne les en-têtes de méthodes. Cette classe diffère légèrement de la classe `Carte` vue précédemment, prenez le temps d'analyser ce code et de voir les différences :

```
class Carte:
    def __init__(self, nom, couleur):
        """ Affectation de l'attribut nom et de l'attribut couleur """
        couleur = ('CARREAU', 'COEUR', 'TREFLE', 'PIQUE')
        noms = ['2', '3', '4', '5', '6', '7', '8', '9', '10', 'Valet', 'Dame', 'Roi', 'As']
        valeurs = {'2':2, '3':3, '4':4, '5':5, '6':6, '7':7, '8':8, '9':9, '10':10, 'Valet':11,
        'Dame':12, 'Roi':13, 'As':14}

    def setNom(self, nom):
        """ modifie l'attribut nom (de la liste noms) """

    def getNom(self):
        """ renvoie le nom de la carte (de la liste noms) """

    def getCouleur(self):
        """ renvoie la couleur de la carte (de la liste couleur) """

    def getValeur(self):
        """ renvoie la valeur de la carte (du dictionnaire valeurs) """

    def egalite(self, carte):
        """ Renvoie True si les cartes self et carte ont même valeur, False sinon
        carte: Objet de type Carte """

    def estSuperieureA(self, carte):
        """ Renvoie True si la valeur de self est supérieure à celle de carte, False sinon
        carte: Objet de type Carte """

    def estInferieureA(self, carte):
        """Renvoie True si la valeur de self est inférieure à celle de carte, False sinon
        carte: Objet de type Carte"""
```

✍ Écrire (sur feuille) un **programme** qui va :

- Créer la carte Valet de COEUR que l'on nommera c1.
- Afficher le nom, la valeur et la couleur de c1.
- Créer la carte As de PIQUE que l'on nommera c2.
- Afficher le nom, la valeur et la couleur de c2.
- Modifier le nom de la carte c2 en Roi et afficher le nom, la valeur et la couleur de c2.
- Créer la carte 8 de TREFLE que l'on nommera c3.
- Comparer les cartes c1 et c2 puis c1 et c3.