23 NSI 40

In [ ]:
```python
class Noeud:
    def __init__(self, valeur):
        '''Méthode constructeur pour la classe Noeud.
        Paramètre d'entrée : valeur (str)'''
        self.valeur = valeur
        self.gauche = None
        self.droit = None

    def getValeur(self):
        '''Méthode accesseur pour obtenir la valeur du noeud
        Aucun paramètre en entrée'''
        return self.valeur

    def droitExiste(self):
        '''Méthode renvoyant True si l'enfant droit existe
        Aucun paramètre en entrée'''
        return (self.droit is not None)

    def gaucheExiste(self):
        '''Méthode renvoyant True si l'enfant gauche existe
        Aucun paramètre en entrée'''
        return (self.gauche is not None)

    def inserer(self, cle):
        '''Méthode d'insertion de clé dans un arbre binaire de recherche
        Paramètre d'entrée : cle (int)'''
        if cle < ...:
            # on insère à gauche
            if self.gaucheExiste():
                # on descend à gauche et on retente l'insertion de la clé
                ...
            else:
                # on crée un fils gauche
                self.gauche = ...
        elif cle > ... :
            # on insère à droite
            if ... :
                # on descend à droite et on retente l'insertion de la clé
                ...
            else:
                # on crée un fils droit
                ... = Noeud(cle)
```

In [ ]:
```python
arbre = Noeud(7)
for cle in (3, 9, 1, 6):
    arbre.inserer(cle)
```