

## TP4 – Généricité

Pour débiter, créez un nouveau projet dans Eclipse nommé **TP Arbre binaire**.

Un arbre binaire est un arbre qui ne peut avoir au maximum que 2 fils. On parle généralement de fils gauche et fils droit. On peut représenter sur les nœuds tout type d'élément. Afin de pouvoir classer les éléments à droite ou à gauche de l'arbre, il est nécessaire de pouvoir définir une notion d'infériorité ou de supériorité entre les éléments : ainsi un élément inférieur à un autre sera placé à la gauche de ce dernier. Inversement, un élément supérieur sera placé à droite.

### Exercice 1 – Arbre binaire d'entiers

L'élément le plus simple représenté sur un nœud est un entier. Nous prendrons ce type d'élément pour la suite. La figure 1 présente l'arbre construit à partir de la liste  $E = \{50, 30, 38, 12, 86, 18, 92, 27, 88, 35\}$ .

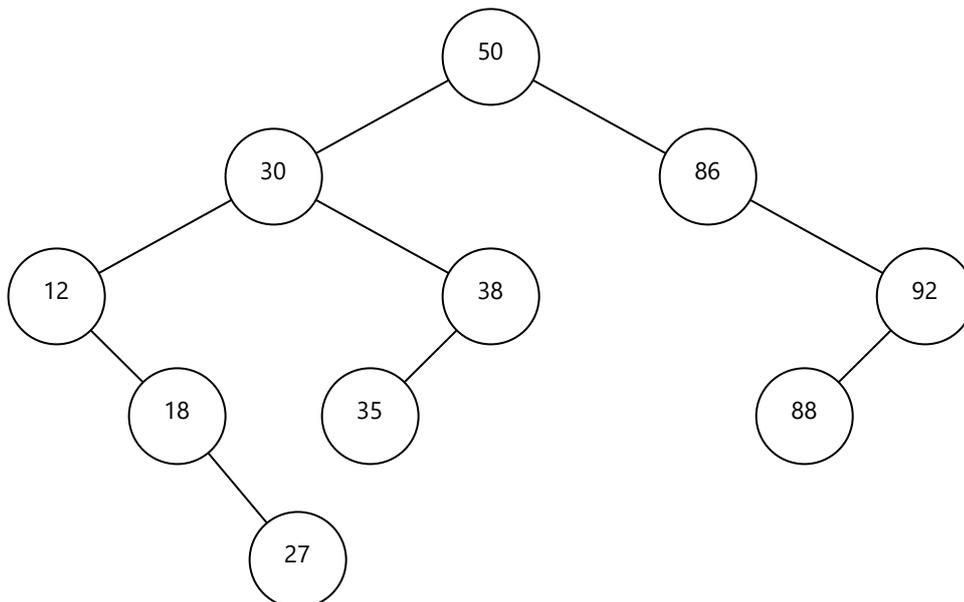


Figure 1 : Arbre construit à partir de l'ensemble d'éléments E

### Représentation informatique d'un arbre binaire

Chaque nœud est généralement représenté par un objet composé de sa valeur (dans notre cas un entier) ainsi que des références vers les nœuds qui correspondent aux fils gauche et droit. Si un nœud n'a pas de fils gauche ou droit, la référence correspondante pointe alors vers null.

## Construction d'un arbre à partir d'une liste d'éléments

La construction de l'arbre s'effectue de la manière suivante :

- Le premier élément de cet ensemble (50) servira de racine à notre arbre,
- Les éléments suivants sont ensuite ajoutés par ordre d'apparition dans l'ensemble.

Pour ajouter une nouvelle valeur, on part de la racine de l'arbre, en considérant la racine comme le nœud courant. On peut ajouter un élément au nœud courant, si

- l'élément à ajouter est inférieur à l'élément du nœud courant
  - si ce dernier n'a pas de fils gauche, on crée un nouveau nœud avec cette valeur et on l'associe au fils gauche ;
  - si le nœud courant a déjà un fils gauche, on essaye alors d'insérer cette nouvelle valeur à partir du fils gauche ;
- l'élément à ajouter est supérieur à l'élément du nœud courant
  - si ce dernier n'a pas de fils droit, on crée un nouveau nœud avec cette valeur et on l'associe au fils droit ;
  - si le nœud courant a déjà un fils droit, on essaye alors d'insérer cette nouvelle valeur à partir du fils droit ;

1. Ecrire une classe **NœudEntier** qui représente un nœud de l'arbre.

Cette classe contiendra un entier correspondant à la valeur du nœud ainsi que deux références de type NœudEntier vers les fils gauche et droit du nœud.

2. Créer les méthodes (récursives)

- *void ajouterValeur(int n)* qui ajoute une valeur n à partir de ce nœud
- *boolean rechercherValeur(int n)* qui recherche une valeur à partir de ce nœud et renvoie un booléen qui vaut vrai si la valeur est trouvée et faux sinon
- *toString* qui permet d'afficher toutes les valeurs à partir de ce nœud par ordre croissant.

## Exercice 2 – Arbre binaire générique

Ecrire une classe **Nœud** qui représente un nœud de l'arbre. Nous souhaitons cette fois-ci que la valeur du nœud soit générique : c'est-à-dire que cette classe pourra fonctionner pour tout type d'objets à partir du moment où nous pouvons les comparer deux à deux.

Dans la méthode main, faire une instance de cet arbre avec :

- des entiers,
- des chaînes de caractères,
- des personnes. Les objets Personne seront des instances d'une classe que vous devrez vous-même créer. Les personnes seront représentées par un nom et un prénom. La comparaison entre deux personnes se fera d'abord sur le nom, puis en cas de nom identique, sur le prénom.