

## TD 11-12 Arbres binaires

### Objectifs

- Manipuler des structures en C#
- Créer et gérer des arbres binaires et des arbres binaires de recherche en C# sans utiliser les classes « Stack », « Queue », « List »
- Suivre un chainage entre les maillons

### Exercice 1 : Premières méthodes

- 1- Récupérer l'archive se trouvant sur Devinci Learning contenant le code c# ; nécessaire pour la création.
- 2- Expliquer comment se fait la création d'un Arbre binaire dans ce programme fourni.
- 3- Ecrire une méthode bool EstFeuille() dans la classe « Nœud.cs » retournant true si le nœud courant est une feuille (si le nœud non NULL n'a pas de fils gauche ni de fils droit), false sinon.
- 4- Ecrire une méthode bool EstArbreVide() dans la classe « ArbreBinaire.cs » retournant true si l'arbre binaire courant est vide, false sinon.
- 5- Ecrire une méthode « NombreEnfants » qui retourne le nombre de fils d'un nœud, dans la classe « Nœud.cs ». Au maximum un nœud aura 2 enfants.

### Exercice 2 – - Affichage de l'Arbre binaire – Parcours en profondeur

Ecrire chacune des méthodes suivantes présentées en CMO :

- 1- void AffichagePrefixe(Nœud n) → Affichage des valeurs de l'arbre en commençant par le nœud courant, puis la partie gauche et enfin la partie droite (une fois la partie gauche totalement traitée, on passe au côté droit).
- 2- void AffichageInfixe(Nœud n) → Affichage des valeurs de l'arbre en commençant par la partie gauche, puis le nœud courant et enfin la partie droite.
- 3- void AffichagePostfixe(Nœud n) → Affichage des valeurs de l'arbre en commençant par la partie gauche puis la partie et enfin le nœud courant.

## Exercice 3 : Comptage des nœuds

Dans la classe « ArbreBinaire », écrire chacune des méthodes récursives suivantes :

- a. « NombreDescendants » : cette méthode calcule et retourne le nombre de descendants (c'est-à-dire : fils + petits-fils + petits petits fils + ....) à partir d'un nœud.
- b. « NombreElements » : calcule et retourne le nombre d'éléments (de nœuds) dans l'arbre y compris la racine.
- c. « NombreFeuilles » : cette méthode calcule et retourne le nombre de feuilles se trouvant dans l'arbre.

## Exercice 4 : Hauteur de l'arbre binaire

Dans la classe « ArbreBinaire », écrire la méthode récursive non terminale « Hauteur » qui retourne la hauteur de l'arbre, c'est-à-dire le nombre maximum de nœuds entre la racine et une feuille (racine et feuilles incluses)

## Exercice 5 : Suppression de l'arbre binaire

Dans la classe « ArbreBinaire », écrire une méthode récursive non terminale qui :

- Supprime l'arbre binaire en libérant l'espace mémoire occupé par chacun des nœuds de l'arbre.
- Met à null le nœud racine de l'arbre lorsque tous les nœuds auront été libérés.

## Exercice 6 : Création utilisateur

Adapter le programme fourni pour pouvoir créer un arbre binaire en dialoguant avec l'utilisateur. Ce dernier pourra saisir plusieurs nœuds pour la construction de son arbre binaire. Votre code devra s'assurer que chacun des nœuds a au maximum deux enfants.