

# TP5: Initiation à la programmation en C++ : Exceptions

K.Zine Elabidine

January 12, 2011

## 1 Mise en bouche

### 1.1 Exceptions : erreur de calcul

Coder les fonctions suivantes en gérant les exceptions possibles :

- addition()
- division()
- multiplication()

Créer un objet Calculatrice qui utilise ces fonctions.

### 1.2 Exceptions : erreur d'allocation de l'espace mémoire

Ecrire une classe test qui déclare deux tableaux d'entier A et B en utilisant la fonction new .

- Penser à gérer l'exception ' bad\_alloc'
- Tester votre programme en essayant d'allouer beaucoup d'espace à votre tableau

### 1.3 Constructeur par copie

Ecrire une class Table ayant pour attribut une tableaux de taille N.

- Créer un constructeur par copie pour cette classe.
- Penser créer un destructeur pour cette classe .

## 2 POO: Arbre binaire de recherche

### 2.1 Définition

Dans cette partie on cherche à modéliser un arbre binaire de recherche . On rappelle qu'un arbre est défini par ses noeuds, chaque noeud possède une clé, telle que chaque noeud du sous arbre gauche ait une clé inférieure ou égale à celle du noeud considéré.

La recherche dans un arbre binaire d'un noeud ayant une clé particulière est un procédé récursif. On commence par examiner la racine. Si sa clé est la clé recherchée, l'algorithme termine et renvoie la racine. Si elle est strictement inférieure, alors elle est dans le sous-arbre gauche, sur lequel on effectue alors récursivement la recherche. De même si la clé de la racine est strictement supérieure à la clé recherchée la recherche continue sur le sous-arbre droit. Si on atteint une feuille dont la clé n'est pas celle recherchée, on sait alors que la clé recherchée n'appartient à aucun noeud.

### 2.2 Travail demandé

Modéliser cette arbre en c++ et implémenter les fonctions suivantes:

- Suppression d'une feuille : Il suffit de l'enlever de l'arbre vu qu'elle n'a pas de fils.
- Suppression d'un noeud avec un enfant : Il faut l'enlever de l'arbre en le remplaçant par son fils.
- Suppression d'un noeud avec deux enfants : Supposons que le noeud à supprimer soit appelé N (le noeud de valeur 7 dans le graphique ci-dessous). On le remplace alors par son successeur le plus proche (le noeud le plus à gauche du sous-arbre droit - ci-dessous, le noeud de valeur 9) ou son plus proche prédécesseur (le noeud le plus à droite du sous-arbre gauche - ci-dessous, le noeud de valeur 6). Cela permet de garder une structure d'arbre binaire de recherche. Puis on applique à nouveau la procédure de suppression à N, qui est maintenant une feuille ou un noeud avec un seul fils.
- Parcours de l'arbre .

Penser à gérer les exceptions et à utiliser un constructeur par copie pour les noeuds ainsi qu'un destructeur.

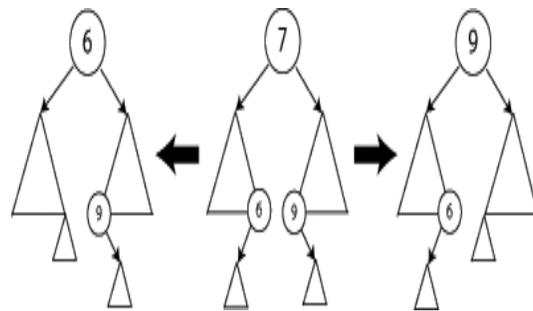


Figure 1: Exemple d'un arbre binaire