

Fiche C++ n° 2 LE DESTRUCTEUR

Nous avons déjà vu le CONSTRUCTEUR. Il s'agit d'une méthode appelée automatiquement lors de la création (instanciation) d'un objet. Cette méthode porte le même nom que la classe.

Il existe aussi une méthode appelée `~Classe()` qui s'appelle le DESTRUCTEUR et qui est appelée automatiquement quand l'objet est détruit.

Il est important de créer un destructeur si des éléments sont alloués dynamiquement (opérateur new) au sein de la classe. Dans ce cas, il faut les enlever de la mémoire dans le destructeur (opérateur delete).

```
class TableEntiers
{
private:
    int *tab;
    int nombreElementsMax;
    int nombreElementsCourants;

public:
    TableEntiers(int n);
    bool insert(int valeur, int position);
    int getValeur(int position);
    int size() { return nombreElementsCourants; }
    int sizeMax() { return nombreElementsMax; }
    ~TableEntiers();
};
```

```
#include "TableEntiers.h"

TableEntiers::TableEntiers(int n)
{
    nombreElementsMax = n;
    tab = new int[nombreElementsMax]; // allocation mémoire
    nombreElementsCourants = 0;
}

TableEntiers::~~TableEntiers()
{
    delete[] tab; // Libération de la mémoire
}

bool TableEntiers::insert(int valeur, int position)
{
    int i;
    ...
}

int TableEntiers::getValeur(int position)
{
    return tab[position];
}
```

L'opérateur **delete** ne doit être utilisé que sur des pointeurs créés avec new. Il existe sous deux formes, **delete** et **delete[]**.

delete Adresse obtenue avec new est utilisé si l'espace alloué contient une valeur unique (objet ou valeur d'un type prédéfini).

delete [] adresse obtenue avec new est utilisé si l'espace alloué représente un tableau.

En résumé, il faut créer un destructeur :

- si une allocation dynamique de mémoire a été réalisée par l'une des méthodes de la classe.
- si un port ou un fichier a été ouvert et qu'il faut le fermer quand l'objet est détruit
- ...