

Fiche de TP : Prélude au Projet

Cette fiche porte sur la mise au point d’un outil indispensable à la réalisation efficace du projet.

Elle porte sur l’écriture d’une classe **Tableau** extensible, dont la vocation sera de stocker des données avec accès direct de manière contiguë. On pourra **sans perte de données** lui ajouter ou supprimer des éléments.

Besoin - 1 *Généricité*

La classe sera *générique*, avec comme argument *template* le type de données à stocker.

Besoin - 2 *Données membres*

Cette classe disposera de 3 données membres

m.valeurs un pointeur sur le type générique permettant de stocker les données du tableau,

m.capacite un entier qui contient la capacité actuelle du tableau,

m.taille un entier qui indique la taille occupée par les données.

Besoin - 3 *Fonctions membres*

Cette classe fournira :

Cycle de vie

- Un constructeur utilisant en argument optionnel la capacité du tableau (par défaut 0)
- Un constructeur par copie.
- Un destructeur.

Opérateurs

- Une surcharge de l’opérateur = qui permettra de recopier des Tableaux. Lors de la copie la capacité sera la capacité minimale pour contenir une copie.
- Deux fonctions d’accès (lecture seule et lecture/écriture) aux données par surcharge de l’opérateur [].

Opérations

- Une fonction **void reserve(const int& nouvelle_capacite)** qui permettra d’augmenter la capacité sans perte de données. Si la capacité proposée est plus petite que l’ancienne, aucune action n’est effectuée.
- Une fonction **void retaille(const int& nouvelle_taille)** qui change la taille des données utilisées. La capacité doit être augmentée au besoin.
- Une fonction **void ajoute(const T& t)** qui permettra d’ajouter une élément à la fin du tableau. La taille sera mise à jour et si la capacité du tableau n’est pas suffisante, elle sera augmentée (doubler la capacité est une bonne stratégie).
- Une fonction **void supprime(const int& i)** qui supprimera la case *i* du tableau. Pour cela, la dernière valeur contenue dans le tableau sera copiée en case *i* et la taille sera réduite de 1.

Accesseurs

- Une fonction retournant la taille des données stockées.
- Une fonction retournant la capacité