

TD/TP 5

La classe TABLE et ses sous-classes

Récupérer les fichiers de la classe table fournis pour ce TP.
/home/shared/ELI5_C++/tp5/exo1
Commencer par regarder les fichiers fournis.

Exercice 1 Définition de la sous-classe LIFO

La classe LIFO est une TABLE mais ces accès sont contraints par le protocole suivant : "le dernier élément entré dans la table le premier élément sorti. C'est ce qu'on appelle plus généralement une pile.

Question 1

Définir les nouveaux attributs de la classe LIFO. (fichier lifo.h)

Question 2

Ecrire le constructeur et le destructeur pour la classe LIFO. On veut pouvoir déclarer un objet LIFO de la façon suivante :

```
LIFO l1;  
LIFO l2(32);  
LIFO * t3 = new LIFO(64);
```

Question 3

Dans un fichier main.cpp écrire une fonction main qui test la création et la destruction des objets.

Question 4

La classe LIFO diffère de la classe TABLE par son mode d'accès aux éléments qu'elle contient. La méthode d'insertion ne change pas on ajoute à la fin, par contre il faut définir la méthode double pop() qui permet d'enlever le dernier élément insérer.

Question 5

Tester votre fonction dans le main

Exercice 2 Définition de la sous-classe FIFO

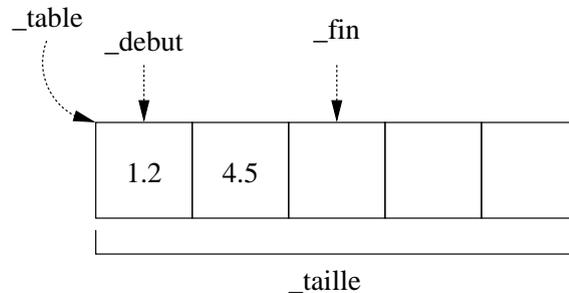
La classe FIFO est une TABLE mais ces accès sont contraints par le protocole suivant : "le premier élément entré dans la table le premier élément sorti. C'est ce qu'on appelle plus généralement une file.

Question 1

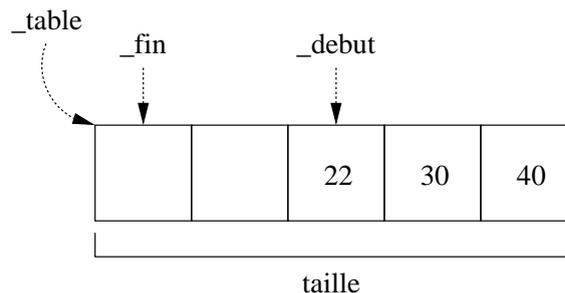
Définir les nouveaux attributs de la classe FIFO. (fichier fifo.h) On utilisera deux pointeurs supplémentaires

- pour savoir le plus vieux élément inséré (c'est celui qui sortira en premier).
- pour savoir où on peut ajouter des éléments.

Exemple :



après plusieurs insert et pop :



Question 2

Ecrire les méthode `end` et `rbegin` qui retournent un pointeur sur le début et la fin de la FIFO.

Question 3

Ecrire le constructeur et le destructeur pour la classe FIFO.

Question 4

Définir les méthodes `insert` et `pop` pour la classe FIFO.

Exercice 3 Template de class

Nous allons maintenant définir une classe générique tableau qui pourra contenir des éléments de types différents de `double`. Vous pouvez récupérer le fichier `template_tab.h` dans `/home/shared/ELI5_C++/tp5/exo3/`. L'implémentation des méthodes d'un template se fait dans le même fichier que ça déclaration sinon l'édition de lien ne pourra pas se faire correctement.

Question 1

Implémenter les méthodes et les opérateurs.

Question 2

Testez votre template dans un main avec les types suivant : `char`, `int`, `list<int>`.