

Construction Objets avancée

Giuseppe Lipari

January 21, 2018

1 Instructions

Vous devez rendre sur github le code demandé avec un fichier README qui contient

- Vos noms
- Pour chaque question
 - Si vous êtes réussi à coder les fonctionnalité demandées
 - La liste de tests de régression correspondants à la question

2 TP 1: Stack class

Vous devez concevoir et coder une classe `Stack`. La classe c'est une pile qui contient des entiers. L'interface de la classe est la suivante:

```
class Stack {  
    ...  
public:  
    Stack(); // default constructor, empty stack  
    Stack(const Stack &s); // copy constructor  
    ~Stack(); // destructor  
  
    bool isEmpty() const; // returns true if empty  
    int top() throws EmptyExc const; // returns the element at the top  
    void pop(); // removes element from the top  
    void push(int elem) // puts an element on top  
    void clear(); // removes all elements  
    int size() const; // number of elements currently in the stack  
    int maxsize() const; // size of the internal representation  
};
```

Quand la pile est vide :

- si l'utilisateur essaie d'appeler `top()`, une exception est levée.
- si l'utilisateur essaie d'appeler `pop()`, il n'y a pas d'exception.

La classe n'a pas, à priori, une limite supérieure au nombre d'éléments qu'elle peut contenir. La représentation interne est un tableau d'entiers de taille fixe. Si la pile est pleine, et on essaie d'insérer un nouveau élément,

1. un nouveau tableau, plus grand, est créé ;
2. le contenu du vieux tableau est copié dans le nouveau ;
3. le vieux tableau est détruit.

2.1 Question 1.1

Coder la classe.

Écrire des tests de régression. En particulier, il faut tester

1. le copy-constructor
2. le destructeur (toute la mémoire est bien libérée)
3. le fait que la taille s'agrandit de manière automatique quand on pousse des nouveaux éléments.

Pour les tests, utilisez le framework `Catch`.

<https://github.com/catchorg/Catch2>

Un exemple est fourni avec le code du TP.

2.2 Question 1.2

Ajouter une fonction membre `void reduce()` qui réduit la taille du tableau pour être égal au nombre d'éléments contenus dans la pile.

Vérifiez le bon fonctionnement de cette fonction avec des tests de régression.

2.3 Question 1.3

Écrire une fonction globale `half()` qui prend un objet `Stack` comme paramètre et enlève la moitié des éléments. Quel est le prototype de cette fonction ?

Tester la fonction `half()`.

2.4 Question 1.4

Écrire une fonction globale `half_copy()` qui prend un objet `Stack` comme paramètre, et rend une copie de la pile avec la moitié des éléments.

Tester également la fonction `half_copy()`.