

Construction Objets avancée

Giuseppe Lipari

January 29, 2019

1 Instructions

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

- Vos noms ;
- Pour chaque question :
 - Si vous avez réussi à coder les fonctionnalités demandées
 - La liste de tests de régression correspondants à la question

2 TP 2: Opérations sur les stacks

2.1 Exercice 1: opérateurs sur le stack

Vous devez étendre la classe `Stack` développée dans le TP1 pour supporter les opérateurs suivants.

2.1.1 Q1: Opérateur d'affectation

Coder l'opérateur membre:

```
Stack &operator=(const Stack &other);
```

Tester l'équivalence avec le constructeur par copie.

2.1.2 Q2: Opérateur de comparaison

Coder l'opérateur membre:

```
bool operator==(const Stack &other) const;
```

avec la sémantique la plus intuitive (le contenu des deux piles doit être égal).

Tester que deux piles sont égales après une opération d'affectation.

2.1.3 Q3: Opérateur d'addition

Coder l'opérateur membre:

```
Stack & operator+=(const Stack &other);
```

Le résultat final est une pile qui contient tous les éléments des deux piles d'origine. L'ordre des éléments est le même que celui qu'on obtient avec l'opération suivante :

```
Stack s1;
Stack s2;

s1 += s2;
/* equivalent to
Stack s_copy(s2);
while (!s_copy.isEmpty()) {
    s1.push(s_copy.top());
    s_copy.pop();
} */
```

Tester cette équivalence.

2.1.4 Q4: Opérateur globale d'addition

Coder l'opérateur global :

```
Stack operator+(const Stack &s1, const Stack &s2);
```

à partir de l'opérateur membre. Tester l'équivalence entre les deux.

2.1.5 Q5: Opérateur d'affichage

Coder l'opérateur global

```
std::ostream& operator<<(std::ostream &os, const Stack &s);
```

qui permet d'imprimer le contenu de la pile sur un sortie.

Tester l'opérateur sur le `stringstream` (help).

2.2 Exercice 2: Les tours de Hanoï

Utiliser la classe `Stack` pour coder l'algorithme de résolution des tours de Hanoï avec 8 disques.

- Voir la description du problème sur Wikipedia.
- Chaque disque est représenté par un entier qui donne sa dimension : le disque plus petit est représenté par le nombre 1, le disque plus grand par le nombre 8.
- Chaque tour est représentée par un `Stack`.

2.2.1 Q6: Safety push

Ajouter à la classe Stack l'opérateur membre suivant :

```
Stack &operator+=(int elem);
```

qui vérifie si le push est correct en comparant le paramètre `elem` avec le top du `Stack` : si oui, il fait le `push` ; si non, il lève une exception de type `IncorrectPush`.

Utiliser l'opérateur ci-dessus pour insérer un disque sur la tour.

2.2.2 Q7: Programme

Coder le programme qui prend en entrée le nombre de disques, et montre la solution sur le terminal, en utilisant l'algorithme récursif décrit dans Wikipedia.