

TP - Templates et surcharges de C++ n° 9

Exercice 1 :

On souhaite écrire une classe **Fraction** qui représente les nombres en fractions entières. L'un des constructeurs de cette classe prendra en entrée deux entiers x et y , et stockera les deux entiers correspondant à la fraction réduite irréductible (c.à.d. réduite du pgcd de x et de y).

1. définissez cette classe et écrivez la méthode statique privée qui calcule le pgcd
2. redéfinissez l'opérateur de sortie « pour permettre un affichage rapide
3. surchargez les opérateurs $+$, $-$ sur les fractions
4. Vérifiez qu'une opération $\frac{1}{2} + 1$ ne compile pas. Ajoutez simplement un constructeur de fractions à une seul argument, le second valant 1. Vérifiez que l'expression précédente est à présent évaluée. Quel est le mécanisme qui a été mis en oeuvre ?
5. complétez afin de pouvoir évaluer également $1 + \frac{2}{4}$.
6. relisez vos déclarations de fonctions pour utiliser au maximum le mot clé **const** et des variables références au lieu de variables copiées lorsqu'elles sont passées en argument.

Exercice 2 :

On utilisera comme point de départ pour implémenter une pile générique le code suivant :¹

```
template <class T> class Stack
{
private:
    int size;
    int top;
    T* stacktab;
    ....
};
```

1. Ecrivez un constructeur vide et un autre avec comme argument la taille, des méthodes pour tester si la pile est vide, si elle est pleine, une méthode pour empiler (seulement lorsqu'il y a de la place, ne rien faire sinon) et une pour dépiler (on suppose qu'il y a bien des éléments).
2. Vous redéfinirez également l'opérateur << pour permettre un affichage rapide.
3. Testez avec **Stack<int>** et **Stack<char>**.
4. On veut maintenant empiler des **Fractions**, mais **Fraction** n'a pas de constructeur vide. Que peut-on empiler ? Faites le nécessaire pour que l'affichage d'une pile de **Fractions** soit convenable.

Exercice 3 :

1. Ecrivez une fonction qui prend un tableau d'un type quelconque, son nombre d'éléments, et qui les affiche.
2. Testez avec des tableaux de type **int**, **char**, **Fraction**.
3. Ecrivez une fonction qui prend un tableau d'un type quelconque, son nombre d'éléments, et qui le trie en utilisant l'ordre induit par l'opérateur <.
4. Vérifiez pour les trois types précédent, et faites le nécessaire si besoin.

1. Pourquoi mes templates ne sont ils pas reconnus à l'édition des liens? la réponse sur : <http://cpp.developpez.com/faq/cpp/?page=Les-templates>