

DÉPARTEMENT DE MATHÉMATIQUES, D'INFORMATIQUE ET DE GÉNIE

Programmation Orientée Objet II
Devoir 3 — Énoncé

SIGLE : INF11207
TITRE : Programmation orientée objet II
GROUPE : 06
PROFESSEUR : Steven Pigeon
K-212
steven_pigeon@uqar.ca

DATE DE REMISE : 5 avril 2016, avant minuit

Les `floats` et des `doubles` sont la façon facile et évidente de faire des calculs, mais ils ne sont pas la seule façon ! On peut aussi assez facilement faire des calculs avec des « fractions vulgaires », les fractions communes qu'on a tous appris à la petite école. Ces fractions ont un entier $a \in \mathbb{Z}$ au *numérateur*, et un naturel ($b \in \mathbb{N}$) au *dénominateur*. La fraction elle-même est exprimée par a/b . Le fait que $a \in \mathbb{Z}$ permet d'avoir des fractions négatives (puisque $\mathbb{Z} = \{\dots, -3, -2, -1, 0, 1, 2, 3, \dots\}$) et $b \in \mathbb{N}$ d'éviter la division par zéro (puisque $\mathbb{N} = \{1, 2, 3, \dots\}$).

Les fractions vulgaires permettent les opérations habituelles, $+$, $-$, \times , \div , qui sont définies tant qu'on n'essaye pas de faire une division par zéro — une opération avec une fraction ayant un dénominateur à zéro est tout à fait invalide. Idéalement, bien que cela ne soit pas strictement nécessaire, le résultat d'une opération est une fraction simplifiée (on dit aussi *réduite*), dont on a éliminé tous les facteurs communs entre le numérateur et le dénominateur.

L'objectif de ce devoir est d'implémenter une classe `fraction` où le numérateur et le dénominateur sont des entiers et où on a procédé à la surcharge des opérateurs de façon à donner à la classe `fraction` toute l'utilité d'un type arithmétique de base.

1. **(15 pts)**. Implémentez une classe `fraction` qui, grâce à la surcharge d'opérateur, supporte les opérations de base :

- **3 pts**. Implémenter les opérateurs unaires : le moins unaire (`-x`), le plus unaire (`+x`), et l'inverse multiplicatif, noté `~x` (tilde).
- **12 pts**. Les opérateurs de base `+`, `-`, `*` et `/`. Chaque opérateur doit permettre de prendre un entier et une fraction, une fraction et un entier, et deux fractions et réaliser l'opération correspondante — il y a donc au moins trois variantes pour chaque opérateur !
- Plus évidemment les constructeurs nécessaires et la surcharge à `<<` pour l'affichage.

Indice : utilisez les constructeurs pour simplifier le code !

2. **(Bonus, 2 pts)**. Faites en sorte que les fractions soient *toujours* simplifiées. C'est-à-dire que la fraction $21/35$ doit être réduite automatiquement à $3/5$. *Indice : l'algorithme du sage géomètre qui vivait à Alexandrie vous aidera beaucoup.*

Faites que votre classe soit compatible (ce n'est pas optionnel !) avec le fichier `main.cpp` trouvé sur Moodle. En temps voulu (c'est-à-dire au moment de la correction), le corps de la fonction `main()` sera remplacé par un programme différent, et c'est ce programme test qui validera votre implémentation.

*
* *

Modalités. Quant à la remise du devoir, préparez une archive (.zip ou autre) qui contiendra votre solution Visual Studio 2010 (ou tout autre suite) *au complet*, et dont le nom sera *votre-nom.devoir3.zip*. Si vous faites le bonus, vous pouvez l'intégrer directement dans votre devoir.