

programmation

TP N 2

Question 1.

Nous nous proposons d'écrire la définition d'une classe *Fraction* munie des méthodes *plus*, *moins*, *multiplie*, *divise*, *equals*, *compareTo*. Une fraction est un couple d'entiers (numérateur, dénominateur) habituellement notée : numérateur / dénominateur. Le dénominateur d'une fraction est toujours différent de 0.

Une fraction est dite *réduite* lorsque son numérateur et son dénominateur sont premiers entre eux. Pour *réduire* une fraction, on calcule le plus grand commun diviseur de son numérateur et de son dénominateur, puis on divise le numérateur et le dénominateur par le plus grand commun diviseur calculé précédemment.

Les opérateurs sont définis de la façon suivante :

1. $\frac{n1}{d1} + \frac{n2}{d2} = \frac{n1*p/d1 + n2*p/d2}{p}$ avec $p = \text{ppcm}(d1, d2)$.
2. $\frac{n1}{d1} - \frac{n2}{d2} = \frac{n1*p/d1 - n2*p/d2}{p}$ avec $p = \text{ppcm}(d1, d2)$.
3. $\frac{n1}{d1} * \frac{n2}{d2} = \frac{n1*n2}{d1*d2}$.
4. $\frac{n1}{d1} / \frac{n2}{d2} = \frac{n1*d2}{d1*n2}$.

Après toute opération, on calculera la fraction réduite ayant la même valeur que la fraction obtenue.

Le *plus petit commun multiple* de deux nombres *a* et *b* est obtenu par la formule : $\text{ppcm}(a,b) = a/\text{pgcd}(a,b)*b$.

Pour calculer le *plus grand commun diviseur*(*a*, *b*) on utilisera l'algorithme suivant :

1. On calcule le reste de la division de *a* par *b*.
2. Si ce reste est 0, le *pgcd* est *b*.
3. Si ce reste est différent de 0, le *pgcd*(*a*, *b*) est le même que le *pgcd*(*b*, *reste*);

Les *constructeurs* d'une instance de la classe *fraction* :

- *Fraction()* initialise par défaut la fraction avec la valeur (0,1),
- *Fraction(int n, int d)* initialise la fraction avec la valeur (n,d),
- *Fraction(int n)* initialise la fraction avec la valeur (n,1),
- *Fraction(String s)* initialise la fraction avec la fraction contenue sous forme de chaîne dans *s*.

On définira les méthodes :

- *String toString ()* qui permet de convertir une fraction en chaîne de caractères : le numérateur, ']' puis le dénominateur,
- *Fraction valueOf(String s)* qui permet d'obtenir une fraction à partir d'une chaîne de caractères contenant un entier, le caractère ']', puis un autre entier.

Question 2.

On utilisera la classe *Fraction* pour écrire un programme qui calcule des expressions arithmétiques de fractions toutes strictement positives : par exemple :

12|34+123|321*5|6=6695|10914

La première partie, jusqu'à = est tapée par l'utilisateur du programme, puis après la frappe de = le programme affiche le résultat.

Pour écrire ce programme, on fera les hypothèses suivantes :

1. Il n'y a jamais d'erreur dans les données tapées par l'utilisateur du programme.
2. Les opérateurs ont la même priorité, et l'opérateur le plus à gauche est évalué en premier. L'expression précédente correspond donc au parenthésage :

(12|34+123|321) * 5|6

Question 3.

Utiliser la classe *Fraction* pour écrire un programme qui traduit une fraction quelconque en sa " notation égyptienne " : toute fraction est notée sous la forme $1|d_1+1|d_2+1|d_3+\dots+1|d_n$. Cette décomposition n'est pas unique, la méthode est donc au choix du programmeur.

Exemple : $3|5=1|2+1|10$



On peut trouver de nombreuses explications sur les fractions égyptiennes ici : <http://www.mcs.surrey.ac.uk/Personal/R.Knott/Fractions/egyptian.html>