

ESIAL 1<sup>ère</sup> année  
 Epreuve de Structures de Données (SD)  
 Date : vendredi 25 mai 2012  
 Horaire : 10h à 12h

Durée du sujet : 1h50

Une feuille A4 autorisée  
 Calculatrices non autorisées

**Consignes :** les trois exercices seront rédigés sur des feuilles séparées. La clarté de la rédaction et la justification des réponses sont des éléments essentiels de l'appréciation. Le barème est donné à titre indicatif. Toute tricherie ou tentative de tricherie sera sévèrement sanctionnée.

## Exercice 1 : Spécification algébrique et tests (7 points)

### Introduction

L'objectif de cet exercice est de spécifier, implanter et tester un type abstrait algébrique. On choisit ici les nombres rationnels, on rappelle que l'ensemble des nombres rationnels est  $\mathbb{Q} = \{\frac{p}{q}, p \in \mathbb{Z} \text{ et } q \in \mathbb{Z}^*\}$ , un nombre rationnel  $\frac{p}{q}$  est donc défini par un entier relatif  $p$ , le numérateur du nombre rationnel et par un entier relatif  $q$  différent de 0, son dénominateur.

On considère la spécification suivante des nombres rationnels sous la forme du type abstrait `Rat`, cette spécification ne prend en compte que quelques opérations des nombres rationnels.

<u>Type Rat</u>	
<u>Opérations</u>	
<code>creerRat</code>	: Integer × Integer → Rat    -- crée un nombre rationnel
<code>num</code>	: Rat → Integer    -- calcule le numérateur d'un nombre rationnel
<code>den</code>	: Rat → Integer    -- calcule le dénominateur d'un nombre rationnel
<code>plus</code>	: Rat × Rat → Rat    -- calcule la somme de deux nombres rationnels
<code>fois</code>	: Rat × Rat → Rat    -- calcule le produit de deux nombres rationnels
<code>reduire</code>	: Rat → Rat    -- rend irréductible un nombre rationnel

L'opération `creerRat` permet de créer un nombre rationnel à partir de deux entiers relatifs, le premier argument de `creerRat` est le numérateur du rationnel créé, le second argument étant le dénominateur.

L'opération `plus` calcule la somme de deux nombres rationnels en utilisant la formule  $plus(\frac{p1}{q1}, \frac{p2}{q2}) = \frac{p1*q2 + p2*q1}{q1*q2}$ .

L'opération `fois` calcule le produit de deux nombres rationnels, en utilisant la formule  $fois(\frac{p1}{q1}, \frac{p2}{q2}) = \frac{p1*p2}{q1*q2}$ .

L'opération `reduire` permet de mettre un nombre rationnel sous forme irréductible en utilisant la formule  $reduire(\frac{p}{q}) = \frac{(p/gcd(p,q))*signe(q)}{(q/gcd(p,q))*signe(q)}$ , où  $gcd$  est une fonction qui calcule le plus grand commun diviseur de deux entiers relatifs et  $signe$  est une fonction qui calcule le signe d'un entier relatif, c'est-à-dire 1, -1 ou 0 selon la positivité du nombre considéré.

### Questions

- Déterminer les opérations internes et les observateurs du type abstrait `Rat` et donner si nécessaire les préconditions des opérations.
- En tenant compte des définitions des opérations, compléter les seconds membres des axiomes suivants :

`num(creerRat(p,q)) = ...`

`den(creerRat(p,q)) = ...`

```

num(plus(r1,r2)) = ...
den(plus(r1,r2)) = ...

num(fois(r1,r2)) = ...
den(fois(r1,r2)) = ...

num(reduire(r)) = ...
den(reduire(r)) = ...

```

Ces axiomes garantissent-ils la complétude suffisante du type abstrait `Rat`? Justifier votre réponse.

- En Java, on choisit d'implanter un nombre rationnel en utilisant deux entiers, le numérateur et le dénominateur. Compléter la classe `Rat` ci-jointe, en écrivant le corps du constructeur et les méthodes `plus`, `fois` et `reduire` qui **modifieront le receveur**.

**Indication** : on suppose que l'on dispose des fonctions `gcd` et `signe` calculant respectivement le pgcd et le signe.

```

package rat ;

public class Rat {
    // les attributs de la classe
    protected int num ;
    protected int den ;

    // Le constructeur de la classe Rat
    public Rat(int n, int d) {
        // A completer
    }

    // Ecrire les methodes plus, fois et reduire
}

```

- Dans le but d'implanter une classe de tests basée sur les axiomes du type abstrait `Rat`, écrire le code Java permettant de tester l'axiome dont la partie gauche est `num(plus(r1,r2))`.
- Les méthodes `plus` et `fois` précédemment écrites ne sont pas satisfaisantes car leur résultat n'est pas sous forme irréductible. Sans réécrire de nouvelles définitions d'opérations et de nouveaux axiomes, écrire deux méthodes Java `plusIrreductible` et `foisIrreductible` implantant respectivement la somme et le produit de deux rationnels et fournissant un rationnel irréductible.

**Indication** : on pourra utiliser les méthodes précédemment écrites.

## Exercice 2 : Implantation de type abstrait (7 points)

### Introduction

On étudie l'implantation des polynômes à coefficients entiers, dont voici une partie de la spécification sous la forme du type abstrait `Polynome` :

<u>Type Polynome</u>		
<u>Opérations</u>		
<code>nul</code>	: $\rightarrow$ Polynome	-- crée le polynôme nul
<code>degré</code>	: Polynome $\rightarrow$ Integer	-- degré
<code>fixerCoefficient</code>	: Polynome $\times$ Integer $\times$ Integer $\rightarrow$ Polynome	-- fixe le coefficient -- d'une puissance donnée
<code>coefficient</code>	: Polynome $\times$ Integer $\rightarrow$ Integer	-- coefficient d'une -- puissance donnée
<code>plus</code>	: Polynome $\times$ Polynome $\rightarrow$ Polynome	-- somme de deux polynômes

On envisage de construire une bibliothèque proposant diverses implantations du type `Polynome`.

## Questions

1. Proposer un diagramme de classes mettant en évidence l'architecture de cette bibliothèque si on envisage deux implantations différentes, l'une utilisant une structure contigüe pour mémoriser les coefficients et l'autre une structure chaînée. Justifiez votre proposition.
2. Ecrire le texte complet des classes que vous avez introduites.
3. Pour les deux implantations proposées, indiquer l'efficacité des opérations (place mémoire et temps d'exécution).

## Exercice 3 : Arbres AVL et équilibrage (6 points)

### Questions

1. Quelles sont les propriétés d'un arbre binaire de recherche ? et celles d'un arbre AVL ?
2. Rappeler le principe d'une rotation gauche (ou respectivement droite) dans un tel arbre ?
3. Construire un arbre AVL correspondant à la suite d'éléments 10, 3, 6, 12, 15 et 8, en procédant à des adjonctions successives aux feuilles et en opérant des rééquilibrages de l'arbre lorsque cela est nécessaire. Vous présenterez les différentes étapes de construction de l'arbre sous la forme de schémas, en précisant les rotations utilisées.