TD POO C++ n°2

© *tv <tvaira@free.fr> v.1.0*

Ce sujet comprend 15 questions pour un total de 20 points.

Remarque : considérer que les questions font partie d'un même programme C++.

On désire développer une classe C++ nommée `Ratio` permettant de traiter des nombres rationnels quelconques sans passer par des `float` ou des `double` qui obligeraient à commettre des erreurs d'arrondi.

La classe `Ratio` code un nombre rationnel de la forme $\frac{n}{d}$ en stockant indépendamment son numérateur (`n`) et son dénominateur (`d`) qui sont tous les deux des entiers. Le numérateur servira à coder le signe du rationnel. Le dénominateur devra donc toujours rester positif.

L'accès aux membres numérateur et dénominateur ne doit pas être permis autrement que par des fonctions accesseurs et mutateurs : des accesseurs permettant d'extraire la valeur du numérateur (ou du dénominateur), et des mutateurs permettant la modification des membres (numérateur ou du dénominateur).

Soit la classe `Ratio` suivante :

```
class Ratio
{
   private:
      int numerateur;
      int denominateur;

      int  pgcd() const;
      void reduire();

   public:
      // ...

      int   getNumerateur() const;
      int   getDenominateur() const;
      void  setNumerateur(int);
      void  setDenominateur(int);
};
```

Une méthode `reduire()` est intégrée à la classe. Elle permet de réduire la fraction : c'est-à-dire de trouver la fraction équivalente ayant les numérateurs et dénominateurs les plus petits possibles en valeur absolue. On rappelle que pour réduire la fraction, il faut diviser le numérateur et le dénominateur par leurs le PGCD. Par exemple, la fraction $\frac{45}{30}$

n'est pas réduite car $45 = 5 \times 3 \times 3$ et $30 = 5 \times 3 \times 2$, on peut donc diviser le numérateur et le dénominateur par le PGCD(45,30)=15, et ainsi obtenir la fraction réduite $\frac{3}{2}$.

La modification par un des mutateurs doit déclencher systématiquement une simplification de la fraction, ainsi la fraction sera toujours réduite.

Il doit être possible de définir une instance de `Ratio` en précisant la valeur du numérateur et du dénominateur. Il doit également être possible de créer un rationnel à partir d'un nombre entier relatif quelconque ; on posera alors que le dénominateur vaut `1`. Enfin, on doit pouvoir créer un rationnel nul lorsqu'on défini un `Ratio` sans préciser la valeur de son numérateur et de son dénominateur. Comme pour toute classe que l'on doit écrire, on devra définir le constructeur de copie pour un `Ratio`. Vous devez utiliser la liste d'initialisation dans les différents constructeurs.

Question 1 (1 point)
Doit-il exister un constructeur par défaut pour cette classe ? Donner sa définiton.

Question 2 (2 points)
Donner la définition du constructeur utilisé lorsque l'on précise la valeur du numérateur et du dénominateur.

Question 3 (1 point)
Donner la déclaration du constructeur de copie.

Question 4 (2 points)
Donner la définition du constructeur de copie. Doit-on appeler la méthode `reduire()` ?

Question 5 (1 point)
Donner la déclaration de l'opérateur d'affectation pour la classe `Ratio`.

Question 6 (2 points)
Donner la définition de l'opérateur d'affectation pour la classe `Ratio`. Peut-on utiliser la liste d'initialisation ?

Question 7 (1 point)
Instancer un objet `r1` de type `Ratio` pourqu'il soit initialisé par défaut.

Question 8 (1 point)
Instancer un objet `r2` de type `Ratio` avec un numérateur 6 et un dénominateur 4.

Question 9 (2 points)
Quelles seront les valeurs affichées par les instructions suivante :

```
cout << "r2.numerateur = " << r2.getNumerateur() << endl;
cout << "r2.denominateur = " << r2.getDenominateur() << endl;
```

Question 10 (1 point)
Donner la déclaration de l'opérateur `+` qui permettra d'écrire : `r1 + r2;`.

Question 11 (2 points)
Donner la définition de l'opérateur `+`.

Question 12 (1 point)
Donner la déclaration de l'opérateur `+=` pour la classe `Ratio` qui permettra d'écrire : `r1 += r2;`.

Question 13 (2 points)
Donner la définition de l'opérateur `+=` pour la classe `Ratio`.

Question 14 (1 point)
Définir l'opérateur de flux `«` pour obtenir un affichage du type `n/d` ou simplement `n` quand le dénominateur est égal à `1` lorsqu'on exécute l'instruction suivante : `cout « r1 « endl;`.

Question 15 (0 points)
On désire ajouter un service permettant la conversion d'un réel `double` en un objet `Ratio` (`https://fr.wikipedia.org/wiki/Fraction_continue`). La fonction devra retourner un rationnel approchant `valeur` avec une erreur au plus égale à `precision`. `valeur` et `precision` seront deux arguments de la fonction. Cette fonction devra-t-elle accéder à des attributs de la classe ? Donner sa déclaration.