CS 101, Spring 2020

Homework Assignment 2

Due: 23:55, May 15, 2020

<u>Instructions:</u> *Submit your homework solutions to Moodle as a single .zip file using*

the following naming convention.

> **SS_HW02_Surname_FirstName.zip**
> where SS is the section number 01 or 02,
> and Surname is your family name, & FirstName is your first name.

Follow the CS101 Java coding style guidelines provided on Moodle (see FAQ).

Important note: All of your methods must be preceded by a JavaDoc comment that gives a brief description of the method's purpose and includes @param tag to list parameters and @return tag to describe what it returns, where appropriate.

Question

Design and implement a class `RationalNumber` with the following properties. A rational number is the ratio of two integers: a numerator and a denominator. The class should supply a set of necessary operations on rational numbers

- addition
- subtraction
- multiplication
- division
- reciprocal

and should hide implementation details of data representation and internal manipulations.

The operations on rational numbers listed above should implement binary operations. In other words you are going to define these methods such that a `RationalNumber` can be added to, subtracted from, multiplied with, or divided by another `RationalNumber`. The result should be a `RationalNumber`.

You should also define the following instance mehods:

- `boolean isInteger( )`
- `boolean isZero( )`
- `boolean isLessThan( RationalNumber other )`
- `boolean equals( RationalNumber other )`
- `String toString( )`

Here are some (internal) representation considerations to guide your design:

- A rational number is a pair of integers called numerator and denominator.
- The numerator carries the sign of the rational number and can be positive, negative, or zero.
- The denominator is kept positive and can never be zero.
- Rational numbers with the same value may have different representations (for example $^{1}/3$, $^{2}/6$, $^{3}/9$). In your representation, equal rational numbers must have the same numerator and the same denominator. In other words, rational numbers must be reduced to their lowest terms. The representation of 0 (zero) as a rational number is $^{0}/1$.

This representation should be enforced by the `RationalNumber` constructor. The constructor should take the two integer parameters n and d and construct a rational number $^{n}/d$. The denominator of the resulting `RationalNumber` should not be zero or negative. The rational number should be reduced by removing the `gcd` (greatest common divisor) between the numerator and the denominator. You should define a helper method `gcd` that computes the greatest common divisor of two numbers and have the constructor use it to create the internal representation. Note that `gcd` should be implemented as a private method because it should not be part of the public interface of the RationalNumber class.

Implement another class, say `TestRationalNumber`, to test your `RationalNumber` class by creating several instances of `RationalNumber` class and performing operations on them and displaying the results on the console.

Sample usage:

```
public class TestRationalNumber
{
   public static void main( String[] args )
   {
      RationalNumber x;
      RationalNumber y;
      RationalNumber z;

      x = new RationalNumber( 2, 20 );
      y = new RationalNumber( -3, 30 );

      z = x.add( y );
      System.out.println( x + " + " + y " = " + z );

      z = x.multiply( y );
      System.out.println( x + " x " + y " = " + z );

   }
}
```