Rational Class Example

This lab exercise contains starter files with a skeleton for a class called **`Rational`**. Your job will be to implement this class**.** The idea is that a **`Rational`** object represents a rational number. Please recall that a rational number is a number that can be represented as the quotient of two integers. For example, all of the following are valid `Rational` numbers: `5/13, 20/17, 3/7, 7/3, 2/1, 1/2, 20/40`

1. Your first changes will be in the file `Rational.java`.
2. Define two private instance variables of type `int` – one called **`numer`** for the numerator, and one called **`denom`** for the denominator.
3. Provide a constructor that allows the user to specify the numerator and denominator in that order:
   ```
   public Rational(int numerIn, int denomIn)
   ```
4. Provide two "getters":
   ```
   public int getNumer()
   public int getDenom()
   ```
5. Provide a `toString` method. For example, if the rational number is 7/5, then the return value should be the `String` "7/5". (No spaces).
6. Provide an instance method called **`reciprocal`**, which returns the reciprocal of the rational number. For example, if the current object is 7/5, the return value will be a new rational number that represents 5/7.
7. Provide a **`static`** method called **`multiply`**, which takes **two** rational numbers as parameters and returns a new `Rational` number representing their product. **Do not reduce any fractions!**
8. Provide an **instance** method called **`divide`**, which takes **one** rational number as a parameter and returns a new `Rational` number representing the quotient obtained by dividing the current object by the parameter. For example, if `x` is `2/3` and `y` is `5/7` then `x.divide(y)` should yield `14/15`. **Do not reduce any fractions!**
9. Provide an instance method called **`add`**, which takes **one** rational number as a parameter and returns a new `Rational` number representing the sum of the current object plus the parameter. You must use the following formula for computing the sum. **Do not reduce any fractions!**

   The sum of a/b plus c/d is to be computed as:

   (ad + cb) / (bd)

I have provided you with the **`PublicTests.java`** file so that you can see how we are testing those for this lab exercise and so that you can run those tests locally for practice if you would like. You do not have to if you don't want to since the submit server will do the tests when you submit anyway.fs