Wentworth Institute of Technology
COMP1000 – Computer Science I
Fall 2017, Kreimendahl

Programming Assignment 9

In this assignment you are to write a program to solve the following problem. As with all assignments, remember the following submission steps:

- Make sure your code passes at least all the provided JUnit tests
- Create and test Javadoc code documentation
- Save, commit, and push all code changes
- Confirm the latest code is visible via the "Files" section of your repository website
- Confirm that the repository is private, and that the instructor has Developer access

Not that for this problem, some JUnit tests that will be used for grading have not been provided as a part of the starter code. It is your responsibility to thoroughly test your code.

Problem a (`PA9a.java`)

Write a Fraction class – that is, write your own class that will represent a fraction in your program. Each variable of type Fraction represents a single fraction. That means that the class should have at least two data fields that should be private: the numerator and the denominator of the fraction. The class also must have the following public class methods:

- **`Fraction(int n, int d); // constructor that defines a fraction n/d`**
- **`Fraction(); // constructor that defines a default fraction of 0/1`**
- **`Fraction(Scanner s); // constructor that defines a fraction via Scanner input`**
- **`double toDecimal(); // returns the decimal value of the fraction`**
- **`String toString(); // returns the string form of the fraction`**
    - `"numerator" if denominator is 1`
    - `"numerator/denominator (decimal, with three decimal places)" otherwise`
- **`int getNumerator(); // returns the numerator of the fraction`**
- **`int getDenominator(); // returns the denominator of the fraction`**
- **`Fraction plus(Fraction f); // returns fraction + parameter`**
- **`Fraction minus(Fraction f); // returns fraction - parameter`**
- **`Fraction times(Fraction f); // returns fraction * parameter`**
- **`Fraction divides(Fraction f); // returns fraction / parameter`**

You have been provided a private method:

- **`int[] simplifyFraction(int[] f); // returns simplified version of parameter`**

Using this method, all constructors should make sure that the fraction is in reduced form. You may not change the public API – that is, any methods not listed above must be made private.

You will also have to write a program to test your Fraction class. The following output is an example run:

```
== F1 ==
Enter numerator: 1
Enter denominator: 2
== F2 ==
Enter numerator: 3
Enter denominator: 5

F1: 1/2 (0.500)
F2: 3/5 (0.600)
F1+F1: 1
F2+F2: 6/5 (1.200)
F1+F2: 11/10 (1.100)
F2+F1: 11/10 (1.100)
F1-F1: 0
F2-F2: 0
F1-F2: -1/10 (-0.100)
F2-F1: 1/10 (0.100)
F1*F1: 1/4 (0.250)
F2*F2: 9/25 (0.360)
F1*F2: 3/10 (0.300)
F2*F1: 3/10 (0.300)
F1/F1: 1
F2/F2: 1
F1/F2: 5/6 (0.833)
F2/F1: 6/5 (1.200)
F1*F1-F2: -7/20 (-0.350)
```

The program should use the Fraction constructor that takes as an argument a Scanner, the plus/minus/times/divides methods (as opposed to directly computing the outputs), and implicitly the toString method.