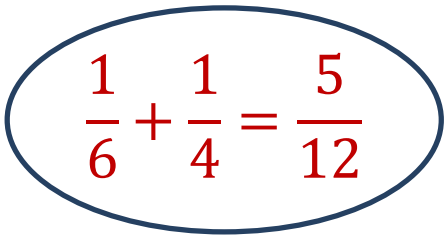
Problem Set 4 Exercise #09: Fraction



Reference: Lecture 10 Unit 2 notes

Learning objective: Object-oriented programming

Estimated completion time: 70 minutes

Problem statement:

[Modified from CS1020 assignment]

Write **PS4_Ex09_Fraction.java** to define a **`Fraction`** class which contains the following attributes:

- **`private int numer; // numerator`**
- **`private int denom; // denominator`**

You are to write the following constructor:

- **`Fraction(int numerator, int denominator)`** that creates a fraction according to the given parameters. You may assume that **`denominator`** is always positive.

The **`Fraction`** class should also provide the following member methods. Note that in each of the following method, the fraction object that invokes a method shouldn't be modified by the method call.

- **`Fraction simplify()`** to return a new **`Fraction`** object which is the simplified form of "this" (the calling) object.
    - Example: If the method is called by fraction 3/12, then it returns a new fraction 1/4. The calling fraction 3/12 remains unchanged after the method call.
- **`Fraction add(Fraction another)`** to add "this" fraction object with **`another`** and return the sum of the two fractions as a new **`Fraction`** object in simplified form.
- **`Fraction minus(Fraction another)`** to return "this" – "another" as a new **`Fraction`** object in simplified form.
- **`Fraction times(Fraction another)`** to return "this" * "another" as a new **`Fraction`** object in simplified form.
- **`Fraction divide(Fraction another)`** to return "this" / "another" as a new **`Fraction`** object in simplified form. You may assume that fraction **`another`** is not zero**.**
- **`equals()`** and **`toString()`** methods.

- **`boolean largerThan(Fraction another)`** methods that returns true if "this" **`Fraction`** object is larger than "another".

You may want to include a member method **`euclidGcd(int a, int b)`** to compute the greatest common divisor of **`a`** and **`b`**, and make this method private access. (Why?)

A client program **PS4_Ex09_TestFraction.java** is provided and should **not** be modified. It does the following:

- Read data to create 2 Fraction objects and display them (use **`toString()`** implicitly).
- Check if the 2 fractions are the same or if one is larger than another (use **`equals()`** and **`largerThan()`** methods).
- Perform addition, subtraction, multiplication and division operations on 2 fraction objects and display results.

You must write your **`Fraction`** class properly such that running **`TestFraction`** produces the same output as the sample runs shown below.

Sample run #1:

```
Enter 1st fraction: 2 20
Enter 2nd fraction: 3 30
1st fraction is 2/20
2nd fraction is 3/30
The two fractions are the same.
2/20 + 3/30 = 1/5
2/20 - 3/30 = 0/1
2/20 * 3/30 = 1/100
2/20 / 3/30 = 1/1
```

Sample run #2:

```
Enter 1st fraction: 0 8
Enter 2nd fraction: 6 14
1st fraction is 0/8
2nd fraction is 6/14
6/14 is larger than 0/8
0/8 + 6/14 = 3/7
0/8 - 6/14 = -3/7
0/8 * 6/14 = 0/1
0/8 / 6/14 = 0/1
```

Sample run #3:

```
Enter 1st fraction: 1 10
Enter 2nd fraction: -5 10
1st fraction is 1/10
2nd fraction is -5/10
1/10 is larger than -5/10
1/10 + -5/10 = -2/5
1/10 - -5/10 = 3/5
1/10 * -5/10 = -1/20
1/10 / -5/10 = -1/5
```

Sample run #4:

```
Enter 1st fraction: -2 5
Enter 2nd fraction: 14 6
1st fraction is -2/5
2nd fraction is 14/6
14/6 is larger than -2/5
-2/5 + 14/6 = 29/15
-2/5 - 14/6 = -41/15
-2/5 * 14/6 = -14/15
-2/5 / 14/6 = -6/35
```

Sample run #5:

```
Enter 1st fraction: 14 6
Enter 2nd fraction: -2 5
1st fraction is 14/6
2nd fraction is -2/5
14/6 is larger than -2/5
14/6 + -2/5 = 29/15
14/6 - -2/5 = 41/15
14/6 * -2/5 = -14/15
14/6 / -2/5 = -35/6
```