

# Class Rational

**3.2.7** Implement a data type `Rational` for rational numbers that supports addition, subtraction, multiplication, and division.

```
public class Rational
```

---

```
    Rational(int numerator, int denominator)
```

```
    Rational plus(Rational b)           sum of this number and b
```

```
    Rational minus(Rational b)         difference of this number and b
```

```
    Rational times(Rational b)         product of this number and b
```

```
    Rational divides(Rational b)       quotient of this number and b
```

```
    String toString()                  string representation
```

Use `Euclid.gcd()` (PROGRAM 2.3.1) to ensure that the numerator and the denominator never have any common factors. Include a test client that exercises all of your methods. Do not worry about testing for integer overflow (see EXERCISE 3.3.17).

The output may be something like:

`1|2 + 1|3 = 5|6`

`1_1|6 / 2|3 = 1_3|4`

```
1  /*
2   * Max @ April, 2017
3   * Refactored on Dec 4, 2017
4   * Refactored on Nov 4, 2018
5   * Refactored on April 20, 2019
6   */
7  public class Rational {
8      private int num;    // numerator    分子
9      private int den;    // denominator  分母
10
11     public static void main (String[] args) {
12         Rational r1 = new Rational( 1, 2);
13         Rational r2 = new Rational( 1, 3);
14         System.out.printf( "%s + %s = %s\n", r1, r2, r1.plus( r2) );
15         System.out.printf( "%s - %s = %s\n", r2, r1, r2.minus( r1) );
16         System.out.printf( "%s * %s = %s\n", r1, r2, r1.times( r2) );
17         // call .toString(), One kind of Arguments Type Promotion.
18
19         Rational r3 = new Rational( 1, 1, 6);
20         Rational r4 = new Rational( 2, 3);
21         Rational r5 = r3.divides( r4);
22         System.out.printf( "%s / %s = %s\n", r3, r4, r5 );
23         System.out.printf( "%s = %s\n", r5, r5.floatString());
24         System.out.printf( "%s = %s\n", r5, r5.floatString(6));
25     }
26
27     public Rational () {
28         num = 0;
29         den = 1;
30     }
```

```

32 public Rational (int num, int den) {
33     this.num = num;
34     this.den = den;    // den > 0
35     normalize();
36 }
37
38 public Rational (int integral, int num, int den) {
39     this( integral * den + num, den );
40 }
41
42 private void normalize () {
43     int absNum = Math.abs( num );
44     int gcd = greatestCommonDivisor( absNum, den);
45     num /= gcd;
46     den /= gcd;
47 }
48
49 // jhtp8, p185, Exe 6.24, Euclid's Algorithm.
50 public static int greatestCommonDivisor (int a, int b) {
51     while (true) {    // 1/2+ Loop
52         int m = a % b;
53
54         if (m == 0) return b;
55
56         a = b;
57         b = m;
58     }
59 } // What about a < b ?

```

```
60
61 public Rational plus (Rational r) {
62     int gcd = greatestCommonDivisor( den, r.den);
63     int theNum = r.den / gcd * num + den / gcd * r.num;
64     int theDen = den / gcd * r.den;
65     return new Rational( theNum, theDen);
66 }
67
68 public Rational minus (Rational r) {
69     int gcd = greatestCommonDivisor( den, r.den);
70     int theNum = r.den / gcd * num - den / gcd * r.num;
71     int theDen = den / gcd * r.den;
72     return new Rational( theNum, theDen);
73 }
74
75 public Rational times (Rational r) {
76     int theNum = num * r.num;
77     int theDen = den * r.den;
78     return new Rational( theNum, theDen);
79 }
80
81 public Rational divides (Rational r) {
82     int theNum = num * r.den;
83     int theDen = den * r.num;
84     return new Rational( theNum, theDen);
85 }
```

```
86
87 public int    intValue ()      { return num / den; }
88 public double doubleValue ()   { return num / (double)den; }
89 public int    getNumerator ()  { return num; }
90 public int    getDenominator () { return den; }
91
92 public String toString () {
93     if (den == 1 || num == 0) return "" + num ;
94     String s = "";
95     int absNum = Math.abs( num );
96     int integral = absNum / den;
97     if (integral == 0) return num + "|" + den ;
98     if (num < 0) s = "-";
99     s = s + integral;
100    int module = absNum % den;
101    if (module == 0) return s;
102    return s + "_" + module + "|" + den ;
103 }
104
105 public String floatString (int... digits) {
106     return (digits.length == 0) ? "" + doubleValue() :
107         String.format( String.format("%%.%df", digits[0]), doubleValue());
108     // String.format("%%.%df", 6) ==> %.6f
109 }
```

```
H:\work>javac Rational.java
```

```
H:\work>java Rational
```

```
1|2 + 1|3 = 5|6
```

```
1|3 - 1|2 = -1|6
```

```
1|2 * 1|3 = 1|6
```

```
1_1|6 / 2|3 = 1_3|4
```

```
1_3|4 = 1.75
```

```
1_3|4 = 1.750000
```