

```
/**
 * Manipuler et calculer avec des fractions.
 *
 * @author biech153 (Biersbach Chris) / gamca174 (Gamboa Carlos) / olial319 (Olinger Alex)
 * @version 27/06/2019 7:00:50
 * Classe: 3GIG
 */
public class Fraction
{
    private int numerator;
    private int denominator;

    public Fraction(int pNumerator, int pDenominator)
    {
        setFraction(pNumerator, pDenominator);
    }

    public void setFraction(int pNumerator, int pDenominator)
    {
        numerator = pNumerator;
        denominator = pDenominator;

        simplify();
    }

    public int getNumerator()
    {
        return numerator;
    }

    public int getDenominator()
    {
        return denominator;
    }

    public double getDecimal()
    {
        return numerator / (double) denominator;
    }

    public String toString()
    {
        String res = "["+numerator;
        if (denominator != 1)
            res = res + "/" + denominator;
        res = res + " (" + getDecimal() + ")"]";
        return res;
    }

    public int gcd(int a, int b)
    {
        int h;
        while (b != 0)
        {
            h = a % b;
            a = b;
            b = h;
        }
        return a;
    }

    public int lcm(int a, int b)
    {
        return (a*b) / gcd(a, b);
    }

    public void simplify()
    {
        int g = gcd(numerator, denominator);
        numerator = numerator / g;
        denominator = denominator / g;
    }

    // ... suite page suivante ...
}
```

```
/**
 * operation:
 *  $f = f + pF$ 
 * (f est l'objet courant)
 */
public void add(Fraction pF)
{
    int l = lcm(denominator, pF.getDenominator());
    int m1 = l / denominator;
    int m2 = l / pF.getDenominator();

    numerator = numerator * m1 + pF.getNumerator() * m2;
    denominator = l;

    simplify();
}

/**
 * operation:
 *  $f = f - pF$ 
 * (f est l'objet courant)
 */
public void subtract(Fraction pF)
{
    int l = lcm(denominator, pF.getDenominator());
    int m1 = l / denominator;
    int m2 = l / pF.getDenominator();

    numerator = numerator * m1 - pF.getNumerator() * m2;
    denominator = l;

    simplify();
}

/**
 * operation:
 *  $f = f * pF$ 
 * (f est l'objet courant)
 */
public void multiply(Fraction pF)
{
    numerator = numerator * pF.getNumerator();
    denominator = denominator * pF.getDenominator();

    simplify();
}

/**
 * operation:
 *  $f = f / pF$ 
 * (f est l'objet courant)
 */
public void divide(Fraction pF)
{
    numerator = numerator * pF.getDenominator();
    denominator = denominator * pF.getNumerator();

    simplify();
}

// ... suite page suivante ...
```

```
/**
 * operation:
 *   fRes <- pF1 + pF2
 *   (l'objet courant n'est pas pris en compte)
 */
public Fraction add(Fraction pF1, Fraction pF2)
{
    Fraction res = new Fraction(pF1.getNumerator(), pF1.getDenominator());
    res.add(pF2);
    return res;
}

/**
 * operation:
 *   fRes <- pF1 - pF2
 *   (l'objet courant n'est pas pris en compte)
 */
public Fraction subtract(Fraction pF1, Fraction pF2)
{
    Fraction res = new Fraction(pF1.getNumerator(), pF1.getDenominator());
    res.subtract(pF2);
    return res;
}

/**
 * operation:
 *   fRes <- pF1 * pF2
 *   (l'objet courant n'est pas pris en compte)
 */
public Fraction multiply(Fraction pF1, Fraction pF2)
{
    Fraction res = new Fraction(pF1.getNumerator(), pF1.getDenominator());
    res.multiply(pF2);
    return res;
}

/**
 * operation:
 *   fRes <- pF1 / pF2
 *   (l'objet courant n'est pas pris en compte)
 */
public Fraction divide(Fraction pF1, Fraction pF2)
{
    Fraction res = new Fraction(pF1.getNumerator(), pF1.getDenominator());
    res.divide(pF2);
    return res;
}
}
```

```
/**
 * Tester la classe Fraction.
 *
 * @author      biech153 (Biersbach Chris) / gamca174 (Gamboa Carlos) / olial319 (Olinger Alex)
 * @version     27/06/2019 7:00:50
 * Classe:      3GIG
 */
public class FractionTest
{
    /**
     * Programme principal.
     */
    public static void main(String[] args)
    {
        Fraction r;
        Fraction f1 = new Fraction(3, 10);
        Fraction f2 = new Fraction(2, 4);

        System.out.print(f1+" + "+f2+" = ");
        f1.add(f2);
        System.out.println(f1);

        System.out.print(f1+" - "+f2+" = ");
        f1.subtract(f2);
        System.out.println(f1);

        System.out.print(f1+" * "+f2+" = ");
        f1.multiply(f2);
        System.out.println(f1);

        System.out.print(f1+" / "+f2+" = ");
        f1.divide(f2);
        System.out.println(f1);

        f1 = new Fraction(3, 10);
        f2 = new Fraction(2, 4);

        System.out.print(f1+" + "+f2+" = ");
        r = f1.add(f1, f2);
        System.out.println(r);

        System.out.print(f1+" - "+f2+" = ");
        r = f1.subtract(f1, f2);
        System.out.println(r);

        System.out.print(f1+" * "+f2+" = ");
        r = f1.multiply(f1, f2);
        System.out.println(r);

        System.out.print(f1+" / "+f2+" = ");
        r = f1.divide(f1, f2);
        System.out.println(r);
    }
}
```