

```
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()
    {
        if (denominator != 0)
            return (double) numerator / denominator;
        else
            return Double.NaN;
    }

    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;
    }

    /**
     * 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();
    }
}
```

```

public class MainFrame extends javax.swing.JFrame
{
    /*
     * Version 1: fraction initiale vaut directement 0/1
     */

    // Fraction avec la valeur 0 (valeur initiale)
    private Fraction fraction = new Fraction(0, 1);

    public MainFrame()
    {
        initComponents();
        updateView();
    }
// Skipped: ... initComponents { ... }
    private Fraction getFractionFromView()
    {
        int n = Integer.valueOf(numeratorTextField.getText());
        int d = Integer.valueOf(denominatorTextField.getText());
        return new Fraction(n, d);
    }

    private void updateView()
    {
        numeratorTextField.setText(String.valueOf(fraction.getNumerator()));
        denominatorTextField.setText(String.valueOf(fraction.getDenominator()));
        decimalLabel.setText(String.valueOf(fraction.getDecimal()));
    }

    private void reduceButtonActionPerformed(java.awt.event.ActionEvent evt) {GEN-FIRST:event_reduceButtonActionPerformed
        fraction.simplify();
        updateView();
    }//GEN-LAST:event_reduceButtonActionPerformed

    private void addButtonActionPerformed(java.awt.event.ActionEvent evt){GEN-FIRST:event_addButtonActionPerformed
    {GEN-HEADEREND:event_addButtonActionPerformed
        fraction.add(getFractionFromView());
        updateView();
    }//GEN-LAST:event_addButtonActionPerformed

    private void multiplyButtonActionPerformed(java.awt.event.ActionEvent evt){GEN-FIRST:event_multiplyButtonActionPerformed
    {GEN-HEADEREND:event_multiplyButtonActionPerformed
        fraction.multiply(getFractionFromView());
        updateView();
    }//GEN-LAST:event_multiplyButtonActionPerformed

    private void subtractButtonActionPerformed(java.awt.event.ActionEvent evt){GEN-FIRST:event_subtractButtonActionPerformed
    {GEN-HEADEREND:event_subtractButtonActionPerformed
        fraction.subtract(getFractionFromView());
        updateView();
    }//GEN-LAST:event_subtractButtonActionPerformed

    private void divideButtonActionPerformed(java.awt.event.ActionEvent evt){GEN-FIRST:event_divideButtonActionPerformed
    {GEN-HEADEREND:event_divideButtonActionPerformed
        fraction.divide(getFractionFromView());
        updateView();
    }//GEN-LAST:event_divideButtonActionPerformed

    private void enterButtonActionPerformed(java.awt.event.ActionEvent evt) {GEN-FIRST:event_enterButtonActionPerformed
        fraction = getFractionFromView();
        updateView();
    }//GEN-LAST:event_enterButtonActionPerformed

// Skipped: ... main routine
// Skipped: ... graphic attributes
}

```