

```

/*Kymberlee Sables (n01364866)
This program performs arithmetic for fractions through object-oriented programming. Specifically, this is done
through classes, objects, constructors, methods, etc., and with the guidance of the UML diagram.*/

import java.util.Scanner;

//TestFraction Class
public class TestFractionSables {
    //main method
    public static void main(String[] args) {
        //variables
        int n1;
        int n2;
        int d1;
        int d2;

        //statements
        Scanner input = new Scanner (System.in);

        //Stage 1
        Fraction myFraction1 = new Fraction();

        //Stage 2
        System.out.printf("Enter a numerator for myFraction1: ");
        n1 = input.nextInt();

        myFraction1.setNum(n1);

        System.out.printf("Enter a denominator for myFraction1: ");
        d1 = input.nextInt();

        myFraction1.setDen(d1);

        //Stage 3
        System.out.printf("\nEnter a numerator for myFraction2: ");
        n2 = input.nextInt();

        System.out.printf("Enter a denominator for myFraction2: ");
        d2 = input.nextInt();

        Fraction myFraction2 = new Fraction(n2, d2);

        //Stage 4
        System.out.print("\nmyFraction1 = ");
        myFraction1.printFraction();

        System.out.print("\nmyFraction2 = ");
        myFraction2.printFraction();

        //Stage 5
        System.out.print("\n\nThe numerator of myFraction1 is: ");
        System.out.print(myFraction1.getNum());

        System.out.print("\n\nThe denominator of myFraction1 is: ");
        System.out.print(myFraction1.getDen());

        System.out.print("\n\nThe numerator of myFraction2 is: ");
        System.out.print(myFraction2.getNum());

        System.out.print("\n\nThe denominator of myFraction2 is: ");
        System.out.print(myFraction2.getDen());

        //Stage 6 (lines 63-69) & Stage 7

        //add
        System.out.print("\n\n");
        myFraction1.printFraction();
        System.out.print(" + ");
        myFraction2.printFraction();
        System.out.print(" = ");
        System.out.print(myFraction1.addFraction(myFraction2));    //adding the 2 fractions

        //subtract
        System.out.print("\n\n");
        myFraction1.printFraction();
        System.out.print(" - ");
        myFraction2.printFraction();
        System.out.print(" = ");
        System.out.print(myFraction1.subtractFraction(myFraction2));    //subtracting the 2 fractions
    }
}

```

```

        //multiply
        System.out.print("\n");
        myFraction1.printFraction();
        System.out.print(" * ");
        myFraction2.printFraction();
        System.out.print(" = ");
        System.out.print(myFraction1.multiplyFraction(myFraction2));    //multiplying the 2 fractions

        //divide
        System.out.print("\n");
        myFraction1.printFraction();
        System.out.print(" / ");
        myFraction2.printFraction();
        System.out.print(" = ");
        System.out.print(myFraction1.divideFraction(myFraction2));    //dividing the 2 fractions

        input.close();
    }
}

//Fraction Class
class Fraction {
    //variables (data field)
    private int num;
    private int den;

    /*no-arguments constructor - default values (myFraction1)
    This is a special method that sets the default values for the data-field when no arguments are taken as input.*/
    Fraction() {
        num = 0;
        den = 1;
    }

    /*constructor - specified values (myFraction2)
    *this is another special method that takes and store the specified values of the data-field.*/
    Fraction(int n, int d) {
        num = n;
        den = d;
    }

    /*method(s) - accessors and mutators
    Accessor - allows access to private data-fields.
    Mutator - allows the data-field to be redefined while being private.*/

    /*accessor for data field "num"
    gives access to "num".*/
    public int getNum() {
        return num;
    }

    /*accessor for data field "den"
    gives access to "den".*/
    public int getDen() {
        return den;
    }

    /*mutator for data field "num"
    can redefine "num".*/
    public void setNum(int num) {
        this.num = num;
    }

    /*mutator for data field "den"
    can redefine "den".*/
    public void setDen(int den) {
        this.den = den;
    }

    /*method - String
    enables Java to display the object's value instead of their address.*/
    public String toString() {
        return (num + " / " + den);
    }

    /*method - printFraction
    class method that prints the objects in fraction format.*/
    public void printFraction() {
        System.out.print(num + " / " + den);
    }
}

```

```

    }

    /*method - reduceFraction
    class method that simplifies the fraction based on the gcd.*/
    private Fraction reduceFraction(int x, int y){
        int g = gcd(x, y);

        x /= g;
        y /= g;

        return new Fraction(x, y);
    }

    /*method - gcd
    class method that calculates the greatest common divisor in order to simplify the fractions.*/
    private static int gcd(int p, int q) {
        if(q == 0)
            return p;
        else
            return gcd(q, p % q);
    }

    /*method - addFraction
    class method that adds the 2 fractions according to the formula and sends it to the reduceFraction method for
    simplification.*/
    public Fraction addFraction(Fraction F2) {
        int nx = (this.num * F2.den) + (F2.num * this.den);
        int dx = this.den * F2.den;

        return reduceFraction(nx, dx);
    }

    /*method - subtractFraction
    class method that subtracts the 2 fractions according to the formula and sends it to the reduceFraction method for
    simplification.*/
    public Fraction subtractFraction(Fraction F2) {
        int nx = (this.num * F2.den) - (F2.num * this.den);
        int dx = this.den * F2.den;

        return reduceFraction(nx, dx);
    }

    /*method - multiplyFraction
    class method that multiplies the 2 fractions according to the formula and sends it to the reduceFraction method for
    simplification.*/
    public Fraction multiplyFraction(Fraction F2) {
        int nx = this.num * F2.num;
        int dx = this.den * F2.den;

        return reduceFraction(nx, dx);
    }

    /*method - divideFraction
    class method that divides the 2 fractions according to the formula and sends it to the reduceFraction method for
    simplification.*/
    public Fraction divideFraction(Fraction F2) {
        int nx = this.num * F2.den;
        int dx = this.den * F2.num;

        return reduceFraction(nx, dx);
    }
}

```