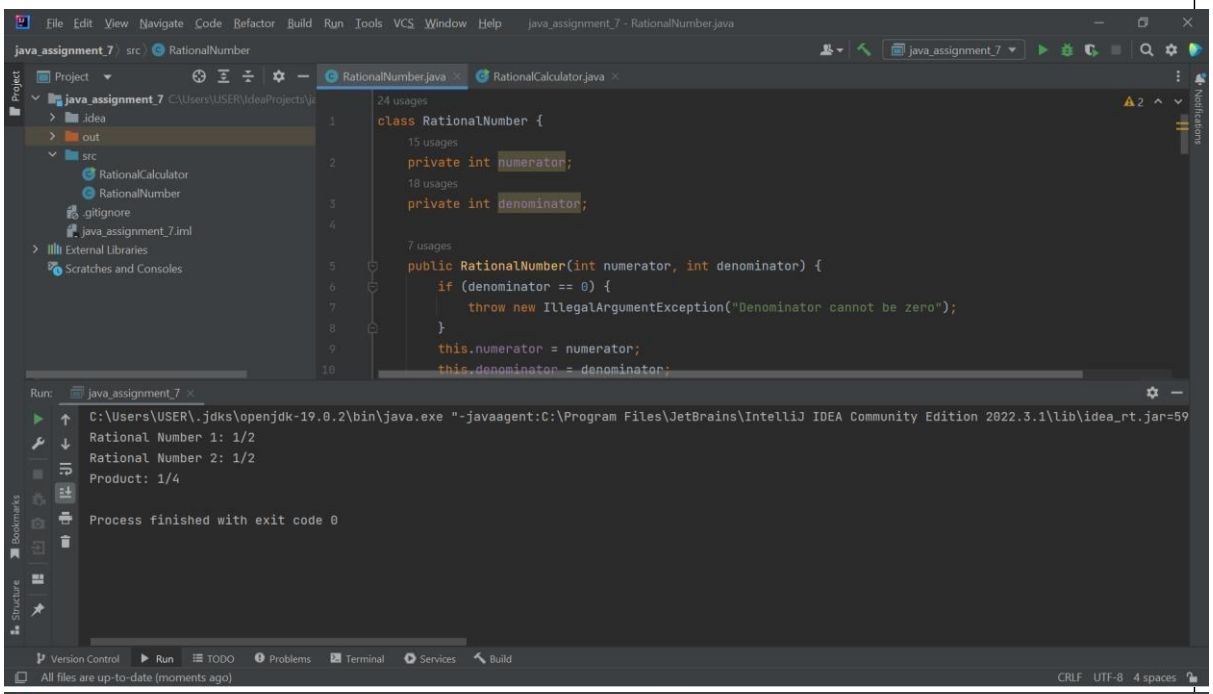
# INPUT:

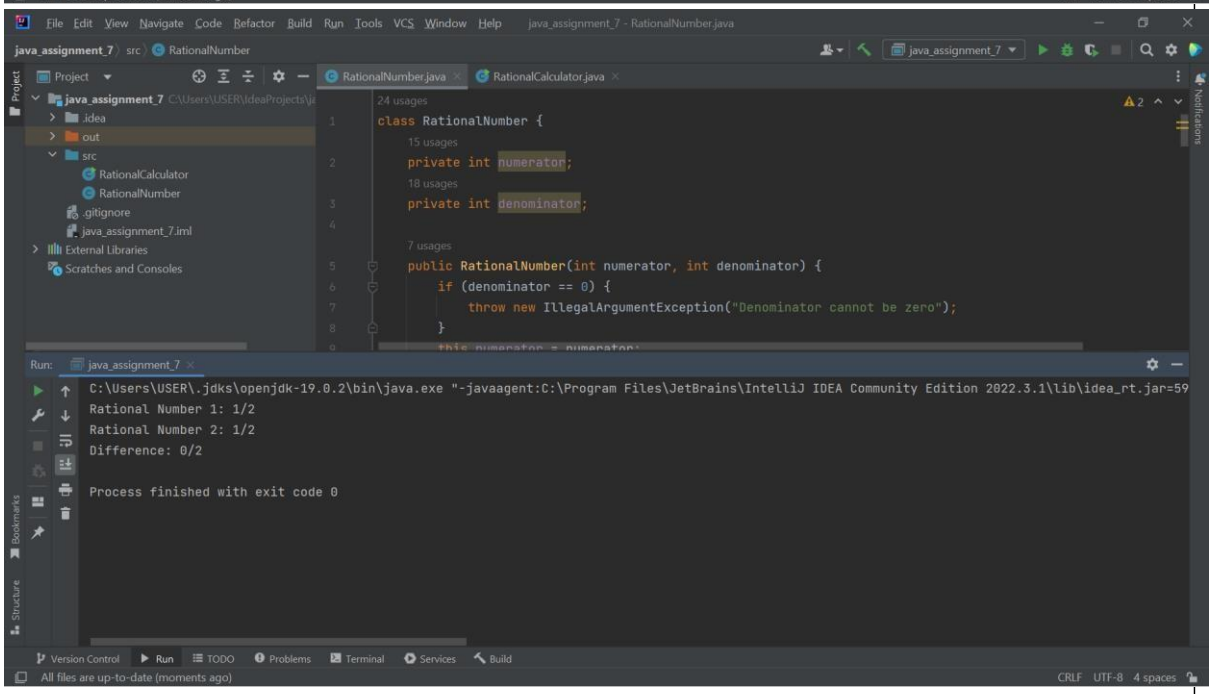
```
class RationalNumber {
    private int numerator;
    private int denominator;

    public RationalNumber(int numerator, int denominator) {
        if (denominator == 0) {
            throw new IllegalArgumentException("Denominator cannot be
zero");
        }
        this.numerator = numerator;
        this.denominator = denominator;
    }

    public RationalNumber add(RationalNumber other) {
        int lcm = getLeastCommonMultiple(this.denominator,
other.denominator);
        int numeratorSum = this.numerator * (lcm / this.denominator) +
other.numerator * (lcm / other.denominator);
        return new RationalNumber(numeratorSum, lcm);
    }

    public RationalNumber subtract(RationalNumber other) {
        int lcm = getLeastCommonMultiple(this.denominator,
other.denominator);
        int numeratorDiff = this.numerator * (lcm / this.denominator) -
other.numerator * (lcm / other.denominator);
        return new RationalNumber(numeratorDiff, lcm);
    }

    public RationalNumber multiply(RationalNumber other) {
        int numeratorProd = this.numerator * other.numerator;
        int denominatorProd = this.denominator * other.denominator;
        return new RationalNumber(numeratorProd, denominatorProd);
    }

    public RationalNumber divide(RationalNumber other) {
        if (other.numerator == 0) {
            throw new IllegalArgumentException("Cannot divide by zero");
        }
        int numeratorQuot = this.numerator * other.denominator;
        int denominatorQuot = this.denominator * other.numerator;
        return new RationalNumber(numeratorQuot, denominatorQuot);
    }

    public boolean isEqualTo(RationalNumber other) {
        return this.numerator * other.denominator == this.denominator *
other.numerator;
    }

    public double toFloatingPoint() {
        return (double) this.numerator / this.denominator;
    }

    public RationalNumber getAbsoluteValue() {
        int absNumerator = Math.abs(this.numerator);
        return new RationalNumber(absNumerator, this.denominator);
    }
```

```
    private int getLeastCommonMultiple(int a, int b) {
        return (a * b) / getGreatestCommonDivisor(a, b);
    }

    private int getGreatestCommonDivisor(int a, int b) {
        if (b == 0) {
            return a;
        } else {
            return getGreatestCommonDivisor(b, a % b);
        }
    }

    @Override
    public String toString() {
        return numerator + "/" + denominator;
    }
}

public class RationalCalculator {
    public static void main(String[] args) {
        try {
            if (args.length != 5) {
                throw new IllegalArgumentException("Please provide four
numbers as command line arguments and specify the operator");
            }

            int num1 = Integer.parseInt(args[0]);
            int den1 = Integer.parseInt(args[1]);
            int num2 = Integer.parseInt(args[2]);
            int den2 = Integer.parseInt(args[3]);
            int operator = Integer.parseInt(args[4]);

            RationalNumber rational1 = new RationalNumber(num1, den1);

            RationalNumber rational2 = new RationalNumber(num2, den2); //
Example second rational number

            System.out.println("Rational Number 1: " +
rational1.toString());
            System.out.println("Rational Number 2: " +
rational2.toString());
            switch(operator){
                case 1:
                    RationalNumber sum = rational1.add(rational2);
                    System.out.println("Sum: " + sum.toString());
                    break;
                case 2:
                    RationalNumber difference =
rational1.subtract(rational2);
                    System.out.println("Difference: " +
difference.toString());
                    break;
                case 3:
                    RationalNumber product = rational1.multiply(rational2);
                    System.out.println("Product: " + product.toString());
                    break;
                case 4:
                    RationalNumber quotient = rational1.divide(rational2);
```

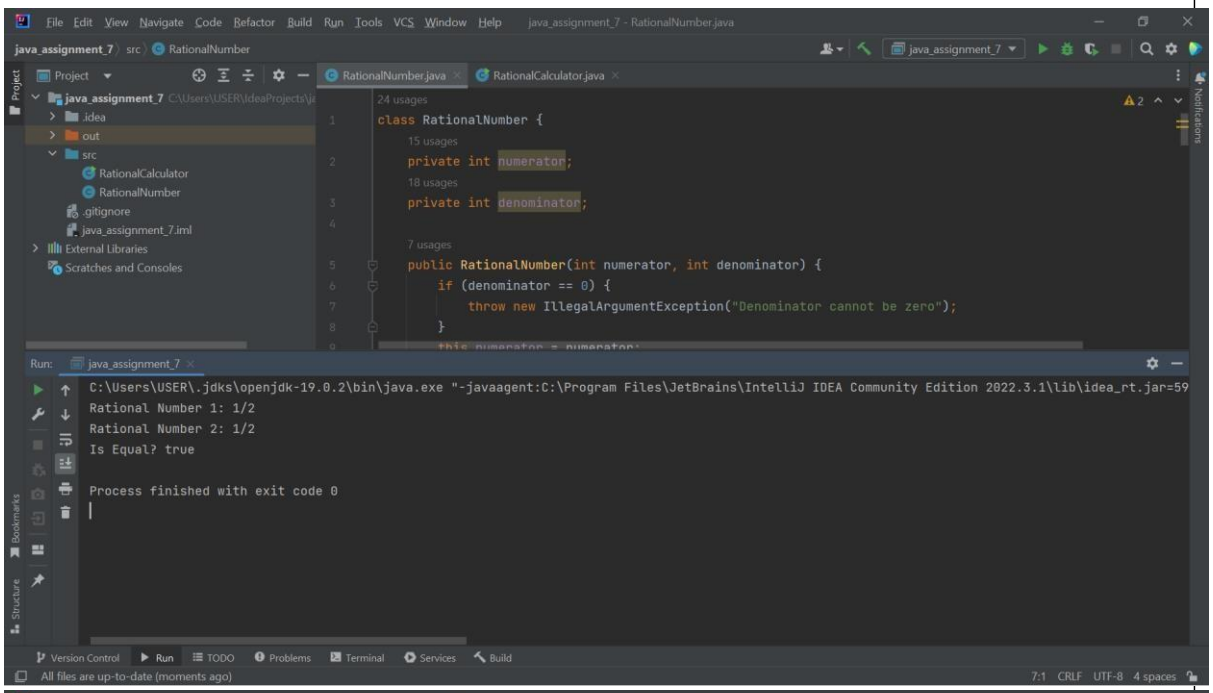
```
                    System.out.println("Quotient: " + quotient.toString());
                    break;
                case 5:
                    boolean isEqual = rational1.isEqualTo(rational2);
                    System.out.println("Is Equal? " + isEqual);
                    break;
                case 6:
                    double floatingPointValue =
rational1.toFloatingPoint();
                    System.out.println("Floating Point Value: " +
floatingPointValue);
                    break;
                case 7:
                    RationalNumber absoluteValue =
rational1.getAbsoluteValue();
                    System.out.println("Absolute Value: " +
absoluteValue.toString());
                    break;
                default:
                    System.out.println("Wrong choice");
                    break;
            }
        } catch (NumberFormatException e) {
            System.err.println("Error: Invalid input. Please provide valid
integers as command line arguments.");
        } catch (IllegalArgumentException e) {
            System.err.println("Error: " + e.getMessage());
        } catch (Exception e) {
            System.err.println("Error: " + e.getMessage());
        }
    }
}
```

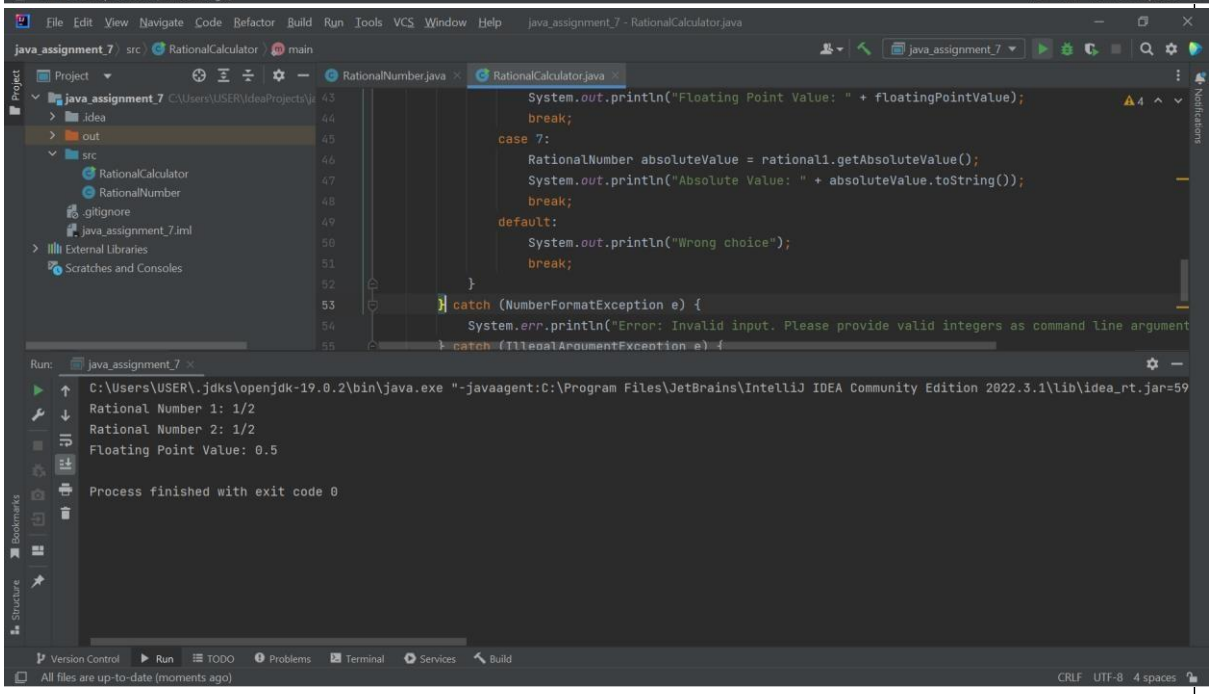
# OUTPUT:


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
java_assignment_7 - RationalNumber.java
RationalNumber.java
RationalCalculator.java
class RationalNumber {
private int numerator;
private int denominator;
public RationalNumber(int numerator, int denominator) {
if (denominator == 0) {
throw new IllegalArgumentException("Denominator cannot be zero");
}
this.numerator = numerator;
C:\Users\USER\.jdks\openjdk-19.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2022.3.1\lib\idea_rt.jar=59
Rational Number 1: 1/2
Rational Number 2: 1/2
Product: 1/4
Process finished with exit code 0


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
java_assignment_7 - RationalNumber.java
RationalNumber.java
RationalCalculator.java
class RationalNumber {
private int numerator;
private int denominator;
public RationalNumber(int numerator, int denominator) {
if (denominator == 0) {
throw new IllegalArgumentException("Denominator cannot be zero");
}
C:\Users\USER\.jdks\openjdk-19.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2022.3.1\lib\idea_rt.jar=59
Rational Number 1: 1/2
Rational Number 2: 1/2
Difference: 0/2
Process finished with exit code 0

File - C:\Users\USER\IdeaProjects\learningjava\src\Calculator.java



**GITHUB LINK: https://github.com/adityasable22/pij-assignment-7.git**