

Manipulating Rational Numbers

Introduction

A number n is called a *rational number* if there are two integers a and b with $b \neq 0$ such that n can be expressed as $n = \frac{a}{b}$. The goal of this assignment is to develop an implementation for performing simple arithmetic operations on rational numbers.

Problem Statement

Develop a **RationalNumber.java** class and write a test-driver program (**RationalNumberDriver.java**) to test the class. Your implementation must meet the following design requirements.

Design Requirements

- The class uses integer fields to represent the numerator and the denominator of a rational number.
- The class stores each rational number in reduced form. That is, the rational number $2/4$, for example, would be stored in the object as 1 in the numerator and 2 in the denominator. This should always be the case whether the rational number is read from the input file or is the result of some arithmetic operation.
- The denominator of a rational number must be stored positive. For example, rational numbers $\frac{3}{-4}$ and $\frac{-3}{-4}$ would be stored in the object as $\frac{-3}{4}$ and $\frac{3}{4}$ respectively). This should always be the case whether the rational number is read from the input file or is the result of some arithmetic operation.
- The class has a default constructor to set the numerator to 0 and the denominator to 1.
- The class contains a `toString` method to display rational numbers in the form a/b . The class contains an additional method for printing rational numbers in floating point format.
- The class provides four public member methods to perform addition, subtraction, multiplication, and division of two fractions.
- The test-driver program must verify that the denominator of any input rational number is not zero. If it is zero, the input is rejected and a message is displayed on the screen.
- The test-driver program performs the four operations: addition, subtraction, multiplication, and division on the two rational numbers. It displays the two fractions involved in the arithmetic operation and the fraction resulting from applying the operation in the form: $a/b \text{ operation } c/d = e/f$. The resulting fraction (e/f) is also displayed in floating point format.
- The test driver obtains its input from a data file (**fractions.txt**).

Program Input

The test-drive program obtains two rational numbers from a data file (**fractions.txt**) with data organized as follows. Each line in the input file represents a fraction. Therefore, each line consists of two integer values separated by a single space. You may assume that the first integer always represents the numerator and the second integer always represents the denominator of a rational number. It is possible that one or both values on a line be zeros. Here are the first few lines in a sample input file:

```
-4 -16
4 0
0 0
1 -8
```

Program Output

Below is a sample output based on the sample input data provided above:

Program is reading input file to validate two rational numbers...

The first valid fraction obtained is: $1/4$ 0.250

Invalid Fraction: denominator cannot be zero.

Invalid Fraction: denominator cannot be zero.

The second valid fraction obtained is: $-1/8$ -0.125

Program is now performing arithmetic operations on the two fractions...

$1/4 + -1/8 = 1/8 = 0.125$

$1/4 - -1/8 = 3/8 = 0.375$

$1/4 * -1/8 = -1/32 = -0.03125$

$1/4 / -1/8 = -2/1 = -2.0$

Program is done.

Bye!

Submission Instructions

When you are satisfied with your implementation, and after testing it thoroughly, submit the **RationalNumber.java** and the **RationalNumberDriver.java** files on Blackboard.