

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Вятский государственный университет»
(ФГБОУ ВО «ВятГУ»)
Институт математики и информационных систем
Факультет автоматики и вычислительной техники
Кафедра электронных вычислительных машин

Отчёт
Лабораторная работа № 1 по дисциплине
«Разработка программных систем»

Вариант 4

Выполнил студент группы ИВТ6-3301-04 _____/Бикметов.И.Р
Проверил преподаватель _____/Чистяков Г.А

Киров 2025

1 Задание

Разработать класс `BigFraction` для работы с дробной длинной арифметикой. Класс должен содержать следующие публичные методы: сложение, вычитание, умножение, деление, сокращение дроби.

Сигнатура методов должна иметь вид «`public BigFraction operation(BigFraction arg)`». Представление дроби должно инкапсулироваться посредством двух экземпляров классов `BigInteger`. Класс должен иметь не менее двух конструкторов. Для корректного представления экземпляров класса при их выводе на экран требуется переопределить метод `toString()`.

2 Листинг программы

```
package main;

import java.math.BigInteger;

/**
 * Class for working with fractional arithmetic
 * @author OOO Shizotekhnika
 * @version 1.0
 */
public class BigFraction {

    /**
     * Variables to represent fractions
     */
    private BigInteger numerator = BigInteger.ONE;
    private BigInteger denominator = BigInteger.ONE;

    /**
     * Object for storing the result of arithmetic operations
     */
    private static BigFraction answer = new BigFraction(BigInteger.ONE, BigInteger.ONE);

    /**
     * Constructor for input values
     * @param numerator of the fraction
     * @param denominator of the fraction
     */
    public BigFraction(BigInteger numerator, BigInteger denominator) {
        this.numerator = numerator;
        this.denominator = denominator;
    }

    /**
     * Constructor for input values
     * @param denominator of the fraction

```

```

    */
    public BigFraction(BigInteger denominator) {
        this.numerator = BigInteger.ONE;
        this.denominator = denominator;
    }

    /**
     * Method to get the numerator
     * @return numerator of the fraction
     */
    public BigInteger GetNumerator() {
        return this.numerator;
    }

    /**
     * Method to get the denominator
     * @return denominator of the fraction
     */
    public BigInteger GetDenominator() {
        return this.denominator;
    }

    /**
     * Method to redefine the numerator
     * @param numerator of the fraction
     */
    public void SetNumerator(BigInteger numerator) {
        this.numerator = numerator;
    }

    /**
     * Method to redefine the denominator
     * @param denominator of the fraction
     */
    public void SetDenominator(BigInteger denominator) {
        this.denominator = denominator;
    }

    /**
     * Method toString for correct representation of class instances
     * @return String with input values and result
     */
    public String toString() {
        return this.numerator + "/" + this.denominator;
    }

    /**
     * Function for adding two fractions
     * @param arg1 Numerator and denominator of two fractions

```

```

    * @return Result of adding two fractions (numerator and denominator) + initial values
    */
    public BigFraction Addition(BigFraction arg) {
        answer.numerator = (arg.numerator.multiply(this.denominator)).add(this.numerator.multiply(this.denominator));
        answer.denominator = arg.denominator.multiply(this.denominator);
        return Reduction(new BigFraction(answer.numerator, answer.denominator));
    }

    /**
     * Function for subtracting two fractions
     * @param arg1 Numerator and denominator of two fractions
     * @return Result of subtracting two fractions (numerator and denominator) + initial values
     */
    public BigFraction Subtraction(BigFraction arg) {
        answer.numerator = (this.numerator.multiply(arg.denominator)).subtract(arg.numerator.multiply(this.denominator));
        answer.denominator = this.denominator.multiply(arg.denominator);
        return Reduction(answer);
    }

    /**
     * Function for multiplication two fractions
     * @param arg Numerator and denominator of two fractions
     * @return Result of multiplication two fractions (numerator and denominator) + initial values
     */
    public BigFraction Multiplication(BigFraction arg) {
        answer.numerator = arg.numerator.multiply(this.numerator);
        answer.denominator = arg.denominator.multiply(this.denominator);
        return Reduction(answer);
    }

    /**
     * Function for division two fractions
     * @param arg Numerator and denominator of two fractions
     * @return Result of division two fractions (numerator and denominator) + initial values
     */
    public BigFraction Division(BigFraction arg) {
        answer.numerator = this.numerator.multiply(arg.denominator);
        answer.denominator = this.denominator.multiply(arg.numerator);
        return Reduction(answer);
    }

    /**
     * Finding the maximum common divider
     * @param numerator Numerator of the number
     * @param denominator Denominator of the number
     * @return Maximum common divider
     */
    private BigInteger SearchCommonDivider(BigInteger numerator, BigInteger denominator) {
        numerator = numerator.abs();

```

```

        denominator = denominator.abs();
        int compare = denominator.compareTo(numerator);

        while (compare !=0) {
            compare = denominator.compareTo(numerator);
            if (compare==1) denominator = denominator.subtract(numerator);
            else numerator = numerator.subtract(denominator);
        }
        return denominator;
    }

    /**
     * Function for reducing fractions
     * @param arg Numerator and denominator of two fractions
     * @return Reduced numerators and denominators of operands and result
     */
    public BigFraction Reduction(BigFraction arg) {
        if(arg.numerator == BigInteger.ZERO) {
            return arg;
        }
        BigInteger commonDiv = SearchCommonDivider(arg.numerator, arg.denominator);
        arg.numerator = arg.numerator.divide(commonDiv);
        arg.denominator = arg.denominator.divide(commonDiv);
        return arg;
    }
}

```

3 Экранные формы

```

Введите номер команды:
3
Введите 1 число
Числитель: 4444
Знаменатель: 555434
Введите 2 число
Числитель: 21212
Знаменатель: 3232332
java.lang.IndexOutOfBoundsException: Очень большой знаменатель!
Знаменатель: 3232
4444/555434 * 21212/3232 = 5303/100988
Используй 'help' для справки.
Введите номер команды:

```

4 Выводы по работе

В ходе выполнения лабораторной работы были изучены основные конструкции языка программирования Java, структура программы, стандартные средства ввода/вывода; изучен основной функционал интегрированной среды разработки Eclipse; написана программа для работы в дробной длинной арифметикой.