

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение высшего
профессионального образования

«Вятский государственный университет»

(ФГБОУ ВО «ВятГУ»)

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

Разработка программных систем

Использование средств автоматической генерации документации и рефакторинга
программного кода

Вариант 4

Выполнил студент группы ИВТ-31 _____/Кудяшев Я.Ю./

Проверил преподаватель _____/Чистяков Г.А./

Киров 2022

1. Задание

В ходе выполнения лабораторной работы решить следующие задачи:

- Провести ряд преобразований программного кода, полученного в ходе выполнения предыдущей работы, с использованием встроенных средств рефакторинга;
- Сопроводить код комментариями с использованием Javadoc;
- Сгенерировать документацию к разработанным классам.

2. Листинг программы

Пример кода с комментариями приведен в приложении А.

3. Документация

Пример документации к разработанным классам приведен в приложении Б.

4. Вывод

В ходе выполнения лабораторной работы были изучены основы Javadoc – стандарта документирования классов Java, получены навыки использования встроенных средств рефакторинга кода и генерации HTML-документации из Javadoc.

Приложение А

(обязательное)

Пример кода с комментариями

BigFraction.java

```
package com.company;

import java.math.BigInteger;

/**
 * Class for working with fractional arithmetic
 *
 * @author Yaroslav Kudyashev
 *
 * @version 1.0
 */

public class BigFraction {

    /**
     * Variables for input and output values
     */

    BigInteger numerator = BigInteger.valueOf(788);

    BigInteger denominator = BigInteger.valueOf(788);

    public static BigFraction answer = new BigFraction(BigInteger.ONE, BigInteger.ONE);

    /**
     * Constructor for input values
     *
     * @param numerator Numerator of the first fraction
     *
     * @param denominator Denominator of the first fraction
     */

    public BigFraction(BigInteger numerator, BigInteger denominator) {

        this.numerator = numerator;

        this.denominator = denominator;

    }

    /**
```

** Method toString for correct representation of class instances*

** @return String with input values and result*

**/*

```
public String toString() {
```

```
    return "The first operator is " + Main.arg1.numerator + "/" + Main.arg1.denominator +
```

```
        "\nThe second operator is " + Main.arg2.numerator + "/" + Main.arg2.denominator +
```

```
        "\nThe answer is " + answer.numerator + "/" + answer.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 of the arg parameter*

**/*

```
public BigFraction Addition(BigFraction arg1) {
```

```
    answer.numerator =  
(arg1.numerator.multiply(Main.arg2.denominator)).add(Main.arg2.numerator.multiply(arg1.denominator));
```

```
    answer.denominator = arg1.denominator.multiply(Main.arg2.denominator);
```

```
    BigFraction last_answer = new BigFraction(answer.numerator, answer.denominator);
```

```
    last_answer = Reduction(last_answer);
```

```
    return last_answer;
```

```
}
```

*/***

** Function for subtracting two fractions*

** @param arg1 Numerator and denominator of two fractions*

** @return Result of subtracting two fractions (numerator and denominator) + initial values of the arg parameter*

**/*

```
public BigFraction Subtraction(BigFraction arg1) {
```

```

        answer.numerator =
(arg1.numerator.multiply(Main.arg2.denominator)).subtract(Main.arg2.numerator.multiply(arg1.denominator));

        answer.denominator = arg1.denominator.multiply(Main.arg2.denominator);

        BigFraction last_answer = new BigFraction(answer.numerator, answer.denominator);

        last_answer = Reduction(last_answer);

        return last_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 of the arg parameter*

*/

```

public BigFraction Multiplication(BigFraction arg) {

    answer.numerator = arg.numerator.multiply(Main.arg2.numerator);

    answer.denominator = arg.denominator.multiply(Main.arg2.denominator);

    BigFraction last_answer = new BigFraction(answer.numerator, answer.numerator);

    last_answer = Reduction(last_answer);

    return last_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 of the arg parameter*

*/

```

public BigFraction Division(BigFraction arg) {

    answer.numerator = arg.numerator.multiply(Main.arg2.denominator);

    answer.denominator = arg.denominator.multiply(Main.arg2.numerator);

    BigFraction last_answer = new BigFraction(answer.numerator, answer.denominator);

```

```

        last_answer = Reduction(last_answer);

        return last_answer;

    }

/**

    * Finding the maximum common divider

    * @param numerator Numerator of the number

    * @param denominator Denominator of the number

    * @return Maximum common divider

    */

private BigInteger Checker(BigInteger numerator, BigInteger denominator) {

    /**

        * Variable for compare

        */

        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) {  
  
    /**  
  
        * Variable for additional calculations  
  
        */  
  
        BigInteger dop;  
  
  
        dop = Checker(answer.numerator, answer.denominator);  
  
        answer.numerator = answer.numerator.divide(dop);  
  
        answer.denominator = answer.denominator.divide(dop);  
  
  
        return answer;  
    }  
}
```

Приложение Б

(обязательно)

Пример документации

Method Summary	
Methods	
Modifier and Type	Method and Description
BigFraction	Addition (BigFraction arg1) Function for adding two fractions
BigFraction	Division (BigFraction arg) Function for division two fractions
BigFraction	Multiplication (BigFraction arg) Function for multiplication two fractions
BigFraction	Reduction (BigFraction arg) Function for reducing fractions
BigFraction	Subtraction (BigFraction arg1) Function for subtracting two fractions
java.lang.String	toString () Method toString for correct representation of class instances

Рисунок 1 – Документация для методов класса BigFraction(1)

Method Detail
toString
<pre>public java.lang.String toString()</pre> <p>Method toString for correct representation of class instances</p> <p>Overrides:</p> <p>toString in class java.lang.Object</p> <p>Returns:</p> <p>String with input values and result</p>
Addition
<pre>public BigFraction Addition(BigFraction arg1)</pre> <p>Function for adding two fractions</p> <p>Parameters:</p> <p>arg1 - Numerator and denominator of two fractions</p> <p>Returns:</p> <p>Result of adding two fractions (numerator and denominator) + initial values of the arg parameter</p>

Рисунок 2 – Документация для методов класса BigFraction(2)