Министерство образования и науки РФ

ФГАОУ ВО «Уральский федеральный университет   
имени первого Президента России Б.Н. Ельцина»

Институт новых материалов и металлургии

Кафедра «Теплофизика и информатика в металлургии»

Оценка работы: \_\_\_\_\_\_\_\_\_\_\_\_\_

Члены комиссии:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Подпись расшифровка подписи

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Подпись расшифровка подписи

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2023 г.

**Разработка программного обеспечения**

**«Расчет центробежной форсунки для аппаратов мокрой очистки газов»**

**«Введение в технологию разработки программного обеспечения»**

Направление 09.03.02 «Информационные системы и технологии»   
(уровень бакалавриата)

Образовательная программа 09.03.02/03.01   
«Информационные системы и технологии в металлургии»

Руководитель

ст. преподаватель, к.т.н. А.С. Истомин

должность, звание подпись расшифровка подписи

Нормоконтролер

ст. преподаватель, к.т.н. А.С. Истомин

должность, звание подпись расшифровка подписи

Студенты:

НМТ-283907 И.В. Машуков

номер группы подпись расшифровка подписи

Екатеринбург

2023

**РЕФЕРАТ**

Пояснительная записка изложена на 25 листах и содержит 10 рисунков и 1 приложение.

РАСЧЕТ ЦЕНТРОБЕЖНОЙ ФОРСУНКИ ДЛЯ АППАРАТОВ МОКРОЙ ОЧИСТКИ ГАЗОВ, ПРОГРАММИРОВАНИЕ, ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ, АРХИТЕКТУРА, ИНТЕРФЕЙС, VISUAL STUDIO C#, ИНСТАЛЛЯЦИЯ.

Проект по модулю посвящен разработке программного обеспечения упрощения расчета центробежной форсунки для аппаратов мокрой очистки газов.

Отражены основные этапы разработки программного обеспечения: постановка задачи, реализация тестового варианта расчета в электронных таблицах Microsoft Excel; проектирование и реализация программного средства – математической библиотеки и пользовательского интерфейса; разработка системы автоматизированного тестирования математической библиотеки; создание справочной документации; подготовка дистрибутива. Размещение исходного программного кода выполнено в системе удаленного контроля версий Bitbucket.

Основными функциями программного обеспечения является расчет центробежной форсунки для аппаратов мокрой очистки газов.

Основные конечные пользователи программного обеспечения – студенты вузов.

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 4](#_Toc509503109)

[1 ПОСТАНОВКА ЗАДАЧИ, ПРОВЕРКА КОРРЕКТНОСТИ АЛГОРИТМА РАСЧЕТА И ОРГАНИЗАЦИЯ РАБОТЫ НАД ПРОЕКТОМ   
В СИСТЕМЕ ATLASSIAN JIRA 5](#_Toc509503110)

[1.1 Физическая постановка задачи 5](#_Toc509503111)

[1.2 Математическая модель задачи 5](#_Toc509503112)

[1.3 Создание тестового варианта расчета в электронных таблицах Microsoft Excel 7](#_Toc509503113)

[2 ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ 8](#_Toc509503115)

[2.1 Разработка архитектуры системы 8](#_Toc509503116)

[2.2 Разработка блок-схемы работы пользователя с программой 8](#_Toc509503117)

[2.3 Создание программного обеспечения в системе управления версиями GitHub 8](#_Toc509503118)

[2.4 Разработка математической библиотеки 9](#_Toc509503119)

[2.5 Реализация пользовательского интерфейса 10](#_Toc509503120)

[2.6 Обработка исключительных ситуаций 10](#_Toc509503121)

[2.7 Создание справочной помощи 11](#_Toc509503122)

[2.8 Создание дистрибутива 12](#_Toc509503123)

[3 РАЗРАБОТКА СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ТЕСТИРОВАНИЯ МАТЕМАТИЧЕСКОЙ БИБЛИОТЕКИ 13](#_Toc509503124)

[3.1 Проектирование системы 13](#_Toc509503125)

[3.2 Реализация системы 13](#_Toc509503126)

[4 ОПИСАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ 15](#_Toc509503127)

[4.1 Установка и настройка программного средства 15](#_Toc509503128)

[4.2 Функциональные возможности программного продукта 16](#_Toc509503129)

[4.3 Технология выполнения расчетов 16](#_Toc509503130)

[ЗАКЛЮЧЕНИЕ 17](#_Toc509503131)

[БИБЛИОГРАФИЧЕСКИЙ СПИСОК 18](#_Toc509503132)

[ПРИЛОЖЕНИЕ 18](#_Toc509503133)

# ВВЕДЕНИЕ

Целью данного проекта является разработка приложения для расчета центробежной форсунки для аппаратов мокрой очистки газов, которое поможет пользователям рассчитывать эквивалентную скорость воды в сопле, диаметр выходного отверстия, коэффициент заполнения сопла, геометрическую характеристику форсунки, эксцентриситет между входным патрубком и соплом форсунки, внутренний диаметр камеры форсунки, высоту камеры закручивания, коэффициент расхода воды через сопло форсунки, расход воды на 1 форсунку, общее расчетное количество форсунок на аппарат и число форсунок, приходящихся на 1 ряд.

Для достижения этой цели необходимо решить следующие задачи:

1. Разработать интерфейс приложения, который будет интуитивно понятен и удобен для использования.

2. Реализовать алгоритмы расчета необходимых параметров.

3. Обработать возможные исключительные ситуации, которые могут возникнуть во время работы приложения.

Для реализации данного проекта будет использоваться язык программирования C# и среда разработки Visual Studio.

В результате выполнения проекта будет создано полноценное приложение, которое поможет пользователям рассчитать параметры для центробежной форсунки для аппаратов мокрой очистки газов.

# 1 ПОСТАНОВКА ЗАДАЧИ, ПРОВЕРКА КОРРЕКТНОСТИ АЛГОРИТМА РАСЧЕТА И ОРГАНИЗАЦИЯ РАБОТЫ НАД ПРОЕКТОМ

## 1.1 Физическая постановка задачи

Необходимо разработать программное обеспечение, которое будет представлять собой расчет центробежной форсунки для аппаратов мокрой очистки газов. Основной задачей программы будет расчет эквивалентной скорости воды в сопле, диаметра выходного отверстия, коэффициента заполнения сопла, геометрической характеристики форсунки, эксцентриситета между входным патрубком и соплом форсунки, внутреннего диаметра камеры форсунки, высоты камеры закручивания, коэффициента расхода воды через сопло форсунки, расхода воды на 1 форсунку, общего расчетного количества форсунок на аппарат и числа форсунок, приходящихся на 1 ряд.

Для расчета необходимых показателей программа должна запрашивать у пользователя общий расход воды на орошение, избыточное давление воды перед форсункой, плотность воды на орошение, коэффициент динамической вязкости воды, рекомендуемый оптимальный диаметр капель распыляемой воды, заданный угол раскрытия факела форсунки, заданное число входных каналов вихревой камеры, заданный диаметр входного канала вихревой камеры, заданное число рядов форсунок, а затем проводить необходимые вычисления.

## 1.2 Математическая модель задачи

Скорость истечения жидкости из сопла форсунки:

 *м/c*, (12.1)

Условная скорость истечения

, (12.2)

где Vж – расход воды на форсунку, м3/с; Pж – давление воды, Па;

ρж – плотность воды, кг/м3; do – диаметр сопла, м; φ – коэффициент заполнения сопла.

Расход воды на форсунку:

, (12.3)

где Кф – коэффициент расхода воды через сопло, определяемый по формуле

, (12.4)

Геометрическую характеристику форсунки **Аф** находят из выражения

, (12.5)

где R – эксцентриситет форсунки, *м*;

nф – число входных каналов вихревой камеры;

Fвx = *πdвх2/4* - площадь сечения входного канала с заданным диаметром *dвx, м2*.

Коэффициент заполнения сопла 0 < φ < 1 определяется по эмпирической формуле по заданному углу *α:*

φ=1–0,001134α1,322.

Установлено, что в диапазоне чисел Рейнольдса *2280 ≤ R ≤ 18280* между средним размером капель и диаметром сопла существует зависимость вида

dк/do=18,3/(Reэ)0,59 (12.7)

где Reэ = *ωэdэρэ/µж*;

µж – коэффициент динамической вязкости воды, *Па с*.

Определив по формуле (12.7) значение *do*, находят эксцентриситет форсунки

R=Aфnфdвх2/2do,*м,*(12.8)

внутренний диаметр вихревой камеры

D = 2К + dвх, *м,*(12.9)

высоту вихревой камеры

Н = 1,2dвх, *м*, (12.10)

и принимают угол конусности на входе потока в сопло *γ = 90 ÷ 120°*.

По заданному общему расходу воды на орошение (*Vж*) определяется необходимое количество форсунок

Nф=Vж/Vф (12.11)

и число форсунок, приходящихся на 1 ряд

Nф,р = Nф/np, (12.12)

где np – число заданных рядов форсунок.

## 1.3 Создание тестового варианта расчета в электронных таблицах Microsoft Excel

С помощью электронной таблицы Microsoft Excel была воссоздана система расчета необходимых данных для центробежной форсунки для аппаратов мокрой очистки газов (Рисунок 2).

Изображение выглядит как текст, снимок экрана, число, программное обеспечение

Автоматически созданное описание

Рисунок 1 – Расчеты в Microsoft Excel

# 2 ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

## 2.1 Разработка архитектуры системы

Разработанная система достаточна примитивна и состоит из 3 проектов:  
 1. Библиотека, содержащая в себе класс, используемый для хранения и вычисления данных отображаемых пользователю.

2. Проект тестов, предназначенный для проверки выходных данных и облегчения дальнейшей работы при изменении входных данных (Рисунок 3).

3. Проект Windows Forms, предназначенный для предоставления пользователю возможности использования библиотеки в отдельной программе (Рисунок 4).

## 2.2 Разработка блок-схемы работы пользователя с программой

Пользователь взаимодействует с графическим интерфейсом, вводя исходные значения. Эти значения используется с библиотекой для расчетов, после чего выводятся полученные результаты на интерфейс. Блок-схема работы пользователя с программой представлена на рисунке 5.

Изображение выглядит как текст, Шрифт, диаграмма, линия

Автоматически созданное описание

Рисунок 2 – Блок схема взаимодействия пользователя с программой

## 2.3 Создание программного обеспечения в системе управления версиями GitHub

Создан локальный репозиторий Git и внешний репозиторий GitHub.

https://github.com/MashukovI/TRPOPract

## 2.4 Разработка математической библиотеки

Математическая библиотека будет использоваться для выполнения математических операций в программе и содержать в себе все функции необходимые для расчетов Код класса математической библиотеки представлен на рисунке 3.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 3 – Код библиотеки

## 2.5 Реализация пользовательского интерфейса

В решение добавляем проект Windows Forms на .NET 7.0, добавляем зависимость от библиотеки и создаём графический интерфейс для пользователя

Изображение выглядит как текст, бутылка, снимок экрана

Автоматически созданное описание

Рисунок 4 – Пользовательский интерфейс

## 2.6 Обработка исключительных ситуаций

Обработка исключительных ситуаций, по типу некорректного ввода пользователя, происходит благодаря блоку try-catch. В блоке try – написан код, который выполняется в штатной ситуации, в catch – выводится окно, сигнализирующий об ошибке.

## 2.7 Создание справочной помощи

При помощи методический материалов было создано небольшое руководство по использованию программы(Рисунок 5).

Изображение выглядит как текст, диаграмма, дизайн

Автоматически созданное описание

Рисунок 5 – Справочные материалы

## 2.8 Создание дистрибутива

Для создания данного проекта необходимо скачать расширение Microsoft Visual studio installer projects 2022 (Рисунок 6)

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

Рисунок 6 – Расширение installer

Далее создаем отдельный проект Setup в нем задаем все необходимые параметры.

# 3 РАЗРАБОТКА СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ТЕСТИРОВАНИЯ МАТЕМАТИЧЕСКОЙ БИБЛИОТЕКИ

## 3.1 Проектирование системы

Cпроектированы модульные тесты для математической библиотеки, предусматривающие возможность проверить работоспособность и корректность ее работы.

## 3.2 Реализация системы

Для отлаживания библиотеки, описанной ранее (пункт 2.4), добавляем проект с Unit тестами (NUnit). В него так же добавляем ссылку на проект библиотеки. Пример теста представлен на рисунке 7. Пример ошибки на рисунке 8.

Изображение выглядит как текст, снимок экрана, программное обеспечение, дисплей

Автоматически созданное описание

Рисунок 7 – Код проекта тестов

Изображение выглядит как текст, бутылка, снимок экрана

Автоматически созданное описание

Рисунок 8 – Вывод ошибки

Листинг кода всех вышеперечисленных проектов представлен в Приложении 1.

# 4 ОПИСАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

## 4.1 Установка и настройка программного средства

Для установки программного продукта достаточно следовать инструкциям мастера-установщика(Рисунок 9). Все необходимые для работы файлы будут помещены в папку с приложением.

Изображение выглядит как текст, электроника, снимок экрана, дисплей

Автоматически созданное описание

Рисунок *9* – Мастер-установщик

## 4.2 Функциональные возможности программного продукта

Функциональными возможностями программного продукта являются расчет и структурированный вывод на экран информации (Рис. 10).

Изображение выглядит как текст, бутылка, снимок экрана

Автоматически созданное описание

Рисунок 10 – вывод информации при успешном расчете

## 4.3 Технология выполнения расчетов

Для выполнения расчетов не требовалось сложных подходов, по типу асинхронности или многопоточности, так как они представляют из себя простейшие cpu bound операции. Попытка разделения вычислений на несколько потоков приведет лишь к их замедлению в данном случае, т.к. будет потрачено больше времени на создание и обслуживание фоновых потоков. Асинхронность так же не даст какого-либо эффекта, т.к. не выполняется никаких запросов.

# ЗАКЛЮЧЕНИЕ

В ходе лабораторной работы была разработана программа с использованием Windows Forms в среде разработки Visual Studio. Был создан графический интерфейс, включающий в себя различные элементы управления, такие как кнопки, текстовые поля. Была создана библиотека, позволяющая рассчитать необходимые для произведения расчетов данные, а так же написаны тесты, позволяющие обезопасить дальнейшую работу от ошибок и производить диагностику при изменении входных данных.

Разработка программы в процессе выполнения лабораторной работы позволила закрепить теоретические знания практическими навыками, а также получить ценный опыт работы с инструментами разработки приложений.

# БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Коцюба И.Ю., Чунаев А.В., Шиков А.Н. Основы проектирования информационных систем. Учебное пособие. – СПб: Университет ИТМО, 2015. – 206 с.
2. Горлушкина Н.Н. Системный анализ и моделирование информационных процессов и систем. Учебное пособие. – СПб: Университет ИТМО, 2016. – 120 с.
3. Грекул В.И., Денищенко Г.Н., Коровкина Н.Л. – Проектирование информационных систем: учебное пособие / 2-е изд., испр. – М.: Интернет-Университет информационных технологий (ИНТУИТ.РУ): БИНОМ. Лаборатория знаний, 2010. – 299 c.
4. Троелсен Э. Язык программирования C# 5.0 и платформа .NET 4.5. 6-е изд. – М.: ООО «И.Д. Вильямс», 2015. – 1312 с.
5. Флёнов М.Е. Библия С#. 2-е изд. – СПб.: БХВ-Петербург, 2011. – 560 с.
6. Рихтер Дж. CLR via C#. Программирование на платформе Microsoft .NET Framework 4.0 на языке C#. Практическое руководство - Санкт-Петербург, 2012. – 928 с.
7. Шилдт Г. Полное руководство С# 4.0 [Текст]: учебное пособие / Г. Шилдт — пер. с англ. Берштейн И. В. — Москва: Вильямс, 2012.— 1051 с.
8. Хайруллин Р.С. Программирование на C#: учебное пособие / Р.С. Хайруллин. – Казань: Изд-во Казанск. гос. архитект.-строит. ун-та, 2017. – 153 с.
9. Фаронов В. В. Создание приложений с помощью C#: Руководство программиста [Текст]: учебник / В. В. Фаронов — Москва: Эксмо, 2008. — 576 с.
10. Лавров В.В. Методические указания к выполнению курсовой работы для студентов бакалавриата, обучающихся по направлению «Информационные системы и технологии» и студентов магистратуры, обучающихся по программе «Информационные системы в металлургии» / В.В. Лавров, К.А. Щипанов, А.А. Бурыкин – Екатеринбург: УрФУ, 2014. – 49 с.
11. Лошкарев Н.Б. Указания к оформлению дипломных и курсовых проектов и работ: методические указания / Н.Б. Лошкарев, А.Н. Лошкарев, Л.А. Зайнуллин. – Екатеринбург: УГТУ–УПИ, 2007. – 49 с.

# ПРИЛОЖЕНИЕ

Листинг библиотеки классов

namespace TRPOPractic.Lib

{

public class RasLib

{

public RasLib(double Vj, double Pj, double a, double dvh, double Muj, byte nf, double np, double pj, double dk)

{

if (Vj < 0 || Pj < 0 || a < 0 || dvh < 0 || Muj < 0 || nf < 0 || np < 0 || pj < 0 || dk < 0)

{

throw new Exception("Параметры не должны быть меньше 0");

}

if (f < 0 || f > 1)

{

throw new Exception("[0-1]");

}

\_Vj = Vj;

\_Pj = Pj;

\_a = a;

\_dvh = dvh;

\_Muj = Muj;

\_nf = nf;

\_np = np;

\_pj = pj;

\_dk = dk;

}

private readonly double \_Vj;

private readonly double \_Pj;

private readonly double \_a;

private readonly double \_dvh;

private readonly double \_Muj;

private readonly byte \_nf;

private readonly double \_np;

private readonly double \_pj;

private readonly double \_dk;

public double Omega => 4d \* \_Vj / (Math.PI \* d0 \* d0 \* f);

public double OmegaE => Math.Sqrt(2d \* \_Pj / \_pj); //6,330255116

public double Vf => Kf \* Math.PI \* d0 \* d0 \* OmegaE / 4d; //4,2082E-06

public double Kf => 1d / Math.Sqrt(Af \* Af / (1d - f) + (1d / Math.Pow(f, 2d)));

public double Af => (1d - f) \* Math.Sqrt(2d) / f / Math.Sqrt(f);

public double R => Af \* \_nf \* \_dvh \* \_dvh / (2d \* d0);

public double D => (2d \* Kf) + \_dvh;

public double d0 => \_dk / (18.3d / Math.Pow(ReE, 0.59d));

public double ReE => OmegaE \* \_dk \* \_pj / \_Muj;

public double Fvh => Math.PI \* \_dvh \* \_dvh / 4d;

public double H => 1.2d \* \_dvh;

public double Nf => \_Vj / Vf;

public double Nfreal => Nfrreal\*\_np;

public double Nfr => Nf / \_np;

public double Nfrreal => Math.Truncate(Nfr);

public double f => 1d - (0.001134d \* Math.Pow(\_a, 1.322d));

public double Vfreal => Nfrreal \* Vf;

}

}

Листинг кода тестов

**using TRPOPractic.Lib;**

**namespace TRPO\_Practic\_Tests**

**{**

**public class Tests**

**{**

**[SetUp]**

**public void Setup()**

**{**

**}**

**[Test]**

**public void OmegaE\_Test()**

**{**

**var Vj = 0.034d;**

**var Pj = 20000d;**

**var a = 120;**

**var dvh = 0.02d;**

**var Muj = 0.001004d;**

**var nf = (byte)2;**

**var np = (byte)4;**

**var pj = 998.2d;**

**var dk = 0.0004d;**

**var raslib = new RasLib(Vj, Pj, a, dvh, Muj, nf, np, pj, dk);**

**Assert.That(raslib.OmegaE, Is.EqualTo(6.330255116d).Within(0.000000001));**

**}**

**[Test]**

**public void Vf\_Test()**

**{**

**var Vj = 0.034d;**

**var Pj = 20000d;**

**var a = 120;**

**var dvh = 0.02d;**

**var Muj = 0.001004d;**

**var nf = (byte)2;**

**var np = (byte)4;**

**var pj = 998.2d;**

**var dk = 0.0004d;**

**var raslib = new RasLib(Vj, Pj, a, dvh, Muj, nf, np, pj, dk);**

**Assert.That(raslib.Vf, Is.EqualTo(0.0000042082d).Within(0.0000001));**

**}**

**[Test]**

**public void Kf\_Test()**

**{**

**var Vj = 0.034d;**

**var Pj = 20000d;**

**var a = 120;**

**var dvh = 0.02d;**

**var Muj = 0.001004d;**

**var nf = (byte)2;**

**var np = (byte)4;**

**var pj = 998.2d;**

**var dk = 0.0004d;**

**var raslib = new RasLib(Vj, Pj, a, dvh, Muj, nf, np, pj, dk);**

**Assert.That(raslib.Kf, Is.EqualTo(0.171881368d).Within(0.000000001));**

**}**

**[Test]**

**public void Af\_Test()**

**{**

**var Vj = 0.034d;**

**var Pj = 20000d;**

**var a = 120;**

**var dvh = 0.02d;**

**var Muj = 0.001004d;**

**var nf = (byte)2;**

**var np = (byte)4;**

**var pj = 998.2d;**

**var dk = 0.0004d;**

**var raslib = new RasLib(Vj, Pj, a, dvh, Muj, nf, np, pj, dk);**

**Assert.That(raslib.Af, Is.EqualTo(4.089945853d).Within(0.00000001));//4,08932715**

**}**

**[Test]**

**public void R\_Test()**

**{**

**var Vj = 0.034d;**

**var Pj = 20000d;**

**var a = 120;**

**var dvh = 0.02d;**

**var Muj = 0.001004d;**

**var nf = (byte)2;**

**var np = (byte)4;**

**var pj = 998.2d;**

**var dk = 0.0004d;**

**var raslib = new RasLib(Vj, Pj, a, dvh, Muj, nf, np, pj, dk);**

**Assert.That(raslib.R, Is.EqualTo(0.73722366d).Within(0.000000001));**

**}**

**[Test]**

**public void D\_Test()**

**{**

**var Vj = 0.034d;**

**var Pj = 20000d;**

**var a = 120;**

**var dvh = 0.02d;**

**var Muj = 0.001004d;**

**var nf = (byte)2;**

**var np = (byte)4;**

**var pj = 998.2d;**

**var dk = 0.0004d;**

**var raslib = new RasLib(Vj, Pj, a, dvh, Muj, nf, np, pj, dk);**

**Assert.That(raslib.D, Is.EqualTo(0.363762737d).Within(0.00000001));**

**}**

**[Test]**

**public void d0\_Test()**

**{**

**var Vj = 0.034d;**

**var Pj = 20000d;**

**var a = 120;**

**var dvh = 0.02d;**

**var Muj = 0.001004d;**

**var nf = (byte)2;**

**var np = (byte)4;**

**var pj = 998.2d;**

**var dk = 0.0004d;**

**var raslib = new RasLib(Vj, Pj, a, dvh, Muj, nf, np, pj, dk);**

**Assert.That(raslib.d0, Is.EqualTo(0.002219107d).Within(0.000000001));**

**}**

**[Test]**

**public void ReE\_Test()**

**{**

**var Vj = 0.034d;**

**var Pj = 20000d;**

**var a = 120;**

**var dvh = 0.02d;**

**var Muj = 0.001004d;**

**var nf = (byte)2;**

**var np = (byte)4;**

**var pj = 998.2d;**

**var dk = 0.0004d;**

**var raslib = new RasLib(Vj, Pj, a, dvh, Muj, nf, np, pj, dk);**

**Assert.That(raslib.ReE, Is.EqualTo(2517.47436d).Within(0.0001));**

**}**

**[Test]**

**public void Fvh\_Test()**

**{**

**var Vj = 0.034d;**

**var Pj = 20000d;**

**var a = 120;**

**var dvh = 0.02d;**

**var Muj = 0.001004d;**

**var nf = (byte)2;**

**var np = (byte)4;**

**var pj = 998.2d;**

**var dk = 0.0004d;**

**var raslib = new RasLib(Vj, Pj, a, dvh, Muj, nf, np, pj, dk);**

**Assert.That(raslib.Fvh, Is.EqualTo(0.000314159d).Within(0.000000001));**

**}**

**[Test]**

**public void H\_Test()**

**{**

**var Vj = 0.034d;**

**var Pj = 20000d;**

**var a = 120;**

**var dvh = 0.02d;**

**var Muj = 0.001004d;**

**var nf = (byte)2;**

**var np = (byte)4;**

**var pj = 998.2d;**

**var dk = 0.0004d;**

**var raslib = new RasLib(Vj, Pj, a, dvh, Muj, nf, np, pj, dk);**

**Assert.That(raslib.H, Is.EqualTo(0.024d).Within(0.000000001));**

**}**

**[Test]**

**public void Nf\_Test()**

**{**

**var Vj = 0.034d;**

**var Pj = 20000d;**

**var a = 120;**

**var dvh = 0.02d;**

**var Muj = 0.001004d;**

**var nf = (byte)2;**

**var np = (byte)4;**

**var pj = 998.2d;**

**var dk = 0.0004d;**

**var raslib = new RasLib(Vj, Pj, a, dvh, Muj, nf, np, pj, dk);**

**Assert.That(raslib.Nf, Is.EqualTo(8079.462134d).Within(0.000001));**

**}**

**[Test]**

**public void Nfr\_Test()**

**{**

**var Vj = 0.034d;**

**var Pj = 20000d;**

**var a = 120;**

**var dvh = 0.02d;**

**var Muj = 0.001004d;**

**var nf = (byte)2;**

**var np = (byte)4;**

**var pj = 998.2d;**

**var dk = 0.0004d;**

**var raslib = new RasLib(Vj, Pj, a, dvh, Muj, nf, np, pj, dk);**

**Assert.That(raslib.Nfr, Is.EqualTo(2019.865533d).Within(0.000001));**

**}**

**[Test]**

**public void f\_Test()**

**{**

**var Vj = 0.034d;**

**var Pj = 20000d;**

**var a = 120;**

**var dvh = 0.02d;**

**var Muj = 0.001004d;**

**var nf = (byte)2;**

**var np = (byte)4;**

**var pj = 998.2d;**

**var dk = 0.0004d;**

**var raslib = new RasLib(Vj, Pj, a, dvh, Muj, nf, np, pj, dk);**

**Assert.That(raslib.f, Is.EqualTo(0.364243716d).Within(0.000000001));**

**}**

**}**

**}**

Листинг WinForms

using System.Configuration;

using TRPOPractic.Lib;

private void button1\_Click(object sender, EventArgs e)

{

try

{

RasLib cn = new(

double.Parse(Vj\_textBox.Text),

double.Parse(Pj\_bol\_textBox.Text),

double.Parse(a\_textBox.Text),

double.Parse(dvh\_textBox.Text),

double.Parse(Muj\_textBox.Text),

byte.Parse(nf\_textBox.Text),

double.Parse(np\_textBox.Text),

double.Parse(pj\_mal\_textBox.Text),

double.Parse(dk\_textBox.Text));

textBoxOmegae.Text = cn.OmegaE.ToString();

textBoxd0.Text = cn.d0.ToString();

textBoxf.Text = cn.f.ToString();

textBoxAf.Text = cn.Af.ToString();

textBoxR.Text = cn.R.ToString();

textBoxD.Text = cn.D.ToString();

textBoxH.Text = cn.H.ToString();

textBoxKf.Text = cn.Kf.ToString();

textBoxVf.Text = cn.Vf.ToString();

textBoxNf.Text = cn.Nf.ToString();;

textBoxNfr.Text = cn.Nfr.ToString();

textBoxNfreal.Text = cn.Nfreal.ToString();

textBoxNfrreal.Text = cn.Nfrreal.ToString();

textBoxVfreal.Text = cn.Vfreal.ToString();

Configuration config = ConfigurationManager.OpenExeConfiguration(ConfigurationUserLevel.None);

if (config.AppSettings.Settings["\_textBox\_Vj"] != null)

{

config.AppSettings.Settings["\_textBox\_Vj"].Value = Vj\_textBox.Text;

}

else

{

config.AppSettings.Settings.Add("\_textBox\_Vj", Vj\_textBox.Text);

}

if (config.AppSettings.Settings["\_Pj\_bol\_textBox"] != null)

{

config.AppSettings.Settings["\_Pj\_bol\_textBox"].Value = Pj\_bol\_textBox.Text;

}

else

{

config.AppSettings.Settings.Add("\_Pj\_bol\_textBox", Pj\_bol\_textBox.Text);

}

if (config.AppSettings.Settings["\_a\_textBox"] != null)

{

config.AppSettings.Settings["\_a\_textBox"].Value = a\_textBox.Text;

}

else

{

config.AppSettings.Settings.Add("\_a\_textBox", a\_textBox.Text);

}

if (config.AppSettings.Settings["\_dvh\_textBox"] != null)

{

config.AppSettings.Settings["\_dvh\_textBox"].Value = dvh\_textBox.Text;

}

else

{

config.AppSettings.Settings.Add("\_dvh\_textBox", dvh\_textBox.Text);

}

if (config.AppSettings.Settings["\_Muj\_textBox"] != null)

{

config.AppSettings.Settings["\_Muj\_textBox"].Value = Muj\_textBox.Text;

}

else

{

config.AppSettings.Settings.Add("\_Muj\_textBox", Muj\_textBox.Text);

}

if (config.AppSettings.Settings["\_nf\_textBox"] != null)

{

config.AppSettings.Settings["\_nf\_textBox"].Value = nf\_textBox.Text;

}

else

{

config.AppSettings.Settings.Add("\_nf\_textBox", nf\_textBox.Text);

}

if (config.AppSettings.Settings["\_np\_textBox"] != null)

{

config.AppSettings.Settings["\_np\_textBox"].Value = np\_textBox.Text;

}

else

{

config.AppSettings.Settings.Add("\_np\_textBox", np\_textBox.Text);

}

if (config.AppSettings.Settings["\_pj\_mal\_textBox"] != null)

{

config.AppSettings.Settings["\_pj\_mal\_textBox"].Value = pj\_mal\_textBox.Text;

}

else

{

config.AppSettings.Settings.Add("\_pj\_mal\_textBox", pj\_mal\_textBox.Text);

}

if (config.AppSettings.Settings["\_dk\_textBox"] != null)

{

config.AppSettings.Settings["\_dk\_textBox"].Value = dk\_textBox.Text;

}

else

{

config.AppSettings.Settings.Add("\_dk\_textBox", dk\_textBox.Text);

}

config.Save(ConfigurationSaveMode.Modified);

}

catch { MessageBox.Show("Все параметры должны быть больше 0"); }

}

}

}