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

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2021 г.

**РАЗРАБОТКА ИНФОРМАЦИОННО-МОДЕЛИРУЮЩЕЙ   
СИСТЕМЫ**

**«РАСЧЁТ ЭЛЕКТРИЧЕСКОГО ФИЛЬТРА»**

**КУРСОВАЯ РАБОТА**

**«Технологии программирования»**

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

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

Руководитель

доцент, к.т.н. А.С. Истомин

должность, звание подпись расшифровка подписи

Студент:

НМТ-293907 Е.Ю.Мазлова

номер группы подпись расшифровка подписи

Екатеринбург

2021

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

Институт новых материалов и металлургии

Кафедра «Теплофизика и информатика в металлургии»

УТВЕРЖДАЮ:

Зав. кафедрой \_\_\_\_\_\_\_\_\_\_\_\_ (Н.А. Спирин)

«08» апреля 2021 г.

**Задание**

**на выполнение курсовой работы**

**«Технологии программирования»**

Студент группы НМТ-283907. Образовательная программа: 09.03.02/03.01 – «Информационные системы и технологии в металлургии».

Фамилия: **Мазлова.** Имя: **Елена.** Отчество: **Юрьевна.**

Руководитель: ст. преподаватель, к.т.н. Истомин А.С. ([as.istomin@live.ru](mailto:as.istomin@live.ru) )

Срок выполнения работы: с «08» апреля 2021 г. по «29» мая 2021 г.

1. Тема проекта:

**Разработка информационно-моделирующей системы «Расчёт электрического фильтра».**

2. Содержание курсовой работы

2.1. Пояснительная записка:

* титульный лист;
* бланк задания преподавателя на выполнение проекта;
* оглавление;
* постановка задачи, проверка корректности алгоритма расчета и организация работы над проектом в системе YouTrack:
  + физическая постановка задачи;
  + математическая модель (подробно в цифрах);
  + создание тестового варианта расчета в Microsoft Excel;
  + постановки задач для выполнения отдельных этапов проекта;
* проектирование и реализация программного обеспечения:
  + создание архитектуры системы;
  + разработка блок-схемы работы пользователя с программой;
  + создание программного обеспечения в системе управления версиями GitLab;
  + разработка математической библиотеки;
  + разработка пользовательского интерфейса;
  + обработка исключительных ситуаций;
  + разработка контекстно-зависимой справки;
  + создание дистрибутива;
* разработка системы автоматизированного тестирования математической библиотеки;
* описание программного обеспечения:
  + установка и настройка программного продукта;
  + функциональные возможности программного продукта;
  + технология выполнения расчетов на конкретном примере;
* заключение
* библиографический список;
* приложение:
  + фрагменты листинга программного обеспечения.

2.2. Компьютерные версии

* исходный код проекта расположен в системе удаленного контроля версий GitLab;
* реализация алгоритма расчета в электронных таблицах Microsoft Excel;
* архитектура программного обеспечения в пакете в пакете Microsoft Visio;
* файл справки в пакете Help&Manual v5;
* руководство пользователя в форматах \*.hlp, \*.chm, \*.pdf.

3. Особые дополнительные требования

* программная платформа – операционная система Microsoft Windows;
* среда программирования – Microsoft Visual Studio. Язык программирования C#;
* объектно-ориентированная методология разработки;
* расчетный модуль в виде библиотеки dll, которая подключена к клиентскому модулю;
* обеспечение функциональности программного обеспечения: ввод/корректировка/сохранение варианта исходных данных во внешнем файле; расчет; отображение на форме результатов расчета в численном и графическом виде; формирование отчета с возможностью настройки показателей; экспорт отчета в форматы Microsoft Word, Microsoft Excel, HTML и др. популярные форматы; файл справочной помощи;
* надежность работы, защита от некорректно вводимых данных;
* установка программного средства с помощью инсталлятора;
* нумерацию версий программы проводить в формате 1.YY.MM.NN, где YY – две последние цифры года, MM – номер месяца, NN – номер версии в текущем месяце.

4. К защите предоставляются следующие материалы

* программная реализация (дистрибутив и исходники программного продукта);
* размещение исходного кода проекта в системе GitLab; проверка рабочей версии программного продукта путем компиляции из системы удаленного контроля версий GitLab в присутствии Заказчика;
* наличие выполненных (закрытых) задач по проекту в системе управления проектами и задачами YouTrack;
* файл в электронных таблицах с тестовой версией алгоритма расчета;
* help-файл в программе Help&Manual v5 (исходники и откомпилированный);
* презентация работы;
* пояснительная записка (ПЗ), выполненная в соответствии с требованиями по оформлению курсовых и дипломных работ.

Допуск к защите проекта по модулю осуществляется только после предварительной сдачи преподавателю всего набора содержимого электронных папок и утвержденной пояснительной записки.

5. План выполнения проекта по модулю

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование элементов проектной работы | Сроки | Примечания | Отметки о  выполнении |
| Анализ предметной области. Проверка корректности алгоритма расчета. Выполнение тестовых расчетов в виде xls-файла | 08.04.2021 –  17.04.2021 гг. | Алгоритм расчета в виде Microsoft Excel |  |
| Создание архитектуры. Проектирование структуры класса математической библиотеки Реализация алгоритмического обеспечения. Кодирование модуля dll и клиентского модуля. | 17.04.2021 –  24.04.2021 гг. | Программное обеспечение модуля dll и клиентского модуля |  |
| Подготовка программных тестов для тестирования математической библиотеки | 24.04.2021 –  01.05.2021 гг. | Средства тестирования в Microsoft Visual Studio  NUnit + Resharper (опционально) |  |
| Реализация пользовательского интерфейса.  Отображение в программе результатов расчета в численном виде.  Отображение в программе результатов расчета в графическом виде | 01.05.2021 –  08.05.2021 гг. | Visual Studio |  |
| Формирование отчета | 08.05.2021 –  15.05.2021 гг. | Visual Studio |  |
| Создание справки и установщика приложения | 15.05.2021 –  22.05.2021 гг. | InnoSetup (или аналог)  Help&Manual (или аналог) |  |
| Формирование документации. | 22.05.2021 –  25.05.2021 гг. | Пакет Microsoft Visio  Пакет Ramus Educational |  |
| Оформление пояснительной записки. | 25.05.2021 –  29.05.2021 гг. | Сдача всего содержимого электронных папок и ПЗ на проверку |  |

6. Выполнение проекта по модулю закончено «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_2021 г.

Руководитель работы, ст. преподаватель, к.т.н. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.С. Истомин

Задание к исполнению принял(а) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Е.Ю.Мазлова

**РЕФЕРАТ**

Пояснительная записка изложена на 59 листах, содержит 2 таблицы, 15 рисунков и 5 приложений.

РАСЧЁТ НАГРЕВА ЦИЛИНРА ПРИ ГУ III РОДА, ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ, АРХИТЕКТУРА, ИНТЕРФЕЙС, ИНСТАЛЛЯЦИЯ, VISUAL STUDIO C#.

Курсовая работа посвящена разработке программного обеспечения, представляющего собой информационно-моделирующую систему расчёта нагрева цилиндра при граничных условиях III рода.

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. Размещение исходного программного кода выполнено в системе удаленного контроля версий GitHub.

Основными функциями программного обеспечения являются:

* Выбор материала цилиндра, ввод исходных данных и расчет температуры и времени нагрева;
* Отображение результатов в численном виде;
* Отображение результатов расчета в графическом виде;
* Формирование отчёта исходя из полученных данных.

Основные конечные пользователи программного обеспечения – студенты вузов.

# **СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 8](#_Toc126277818)

[1 ПОСТАНОВКА ЗАДАЧИ, ПРОВЕРКА КОРРЕКТНОСТИ АЛГОРИТМА РАСЧЕТА И ОРГАНИЗАЦИЯ РАБОТЫ НАД ПРОЕКТОМ В СИСТЕМЕ ATLASSIAN JIRA 9](#_Toc126277819)

[1.1 Физическая постановка задачи 9](#_Toc126277820)

[1.2 Математическая модель задачи 10](#_Toc126277821)

[1.3 Создание тестового варианта расчета в электронных таблицах Microsoft Excel 12](#_Toc126277822)

[2 ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ 14](#_Toc126277823)

[2.1 Разработка архитектуры системы 14](#_Toc126277824)

[2.2 Разработка блок-схемы работы пользователя с программой 15](#_Toc126277825)

[2.3 Создание программного обеспечения в системе управления версиями GitHub 15](#_Toc126277826)

[2.4 Разработка математической библиотеки 16](#_Toc126277827)

[2.5 Реализация пользовательского интерфейса 17](#_Toc126277828)

[2.6 Обработка исключительных ситуаций 20](#_Toc126277829)

[2.7 Создание справочной помощи 21](#_Toc126277830)

[2.8 Создание дистрибутива 22](#_Toc126277831)

[3.1 Проектирование системы 23](#_Toc126277832)

[3.2 Реализация системы 23](#_Toc126277833)

[4 ОПИСАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ 25](#_Toc126277834)

[4.1 Установка и настройка программного средства 25](#_Toc126277835)

[4.2 Функциональные возможности программного продукта 25](#_Toc126277836)

[4.3 Технология выполнения расчетов 26](#_Toc126277837)

[ЗАКЛЮЧЕНИЕ 27](#_Toc126277838)

[БИБЛИОГРАФИЧЕСКИЙ СПИСОК 28](#_Toc126277839)

[ПРИЛОЖЕНИЕ А СПРАВОЧНЫЕ ТАБЛИЦЫ 30](#_Toc126277840)

[ПРИЛОЖЕНИЕ Б ФРАГМЕНТ ЛИСТИНГА ПРОГРАММНОГО КОДА КЛАССА TEMPINITIAL.XAML.CS 31](#_Toc126277841)

[ПРИЛОЖЕНИЕ В ФРАГМЕНТ ЛИСТИНГА ПРОГРАММНОГО КОДА КЛАССА RESULTTIME.XAML.CS 36](#_Toc126277842)

[ПРИЛОЖЕНИЕ Г ФРАГМЕНТ ЛИСТИНГА ПРОГРАММНОГО КОДА КЛАССА UNITTEST1.CS 38](#_Toc126277843)

[ПРИЛОЖЕНИЕ Д ФРАГМЕНТ ЛИСТИНГА ПРОГРАММНОГО КОДА КЛАССА FORMULES.CS 47](#_Toc126277844)

# **ВВЕДЕНИЕ**

Современные тенденции развития металлургии характеризуются развитием, внедрением и широким использованием компьютерных систем моделирования и выполнения расчётных процессов в АСУП и АСУ ТП. Уже сегодня на крупнейших металлургических предприятиях Нашей страны разрабатываются и функционируют мощнейшие компьютерные системы, в основе которых лежат методы математического моделирования.

Теплотехнический расчет нагрева цилиндрических заготовок определенного радиуса при нестационарной теплопроводности является важной составляющей не только учебного курса теплотехники, но и реального технологического процесса, ведь правильно подобранный режим и время нагрева обеспечивают качественное изготовление готового продукта. Также, что немаловажно, неправильно подобранный режим термической обработки может повлечь за собой выход из строя важнейших промышленных агрегатов и, как следствие, принести колоссальный материальный ущерб предприятию.

Реализация такого расчета ручным способом нецелесообразна, так как не гарантирует правильность результатов и занимает большое количество времени. В то время как использование информационно-моделирующей системы для подобных расчётов значительно ускоряет процесс расчёта и повышает их качество и эффективность. Исходными данными для расчета являются радиус цилиндра, материал цилиндра, коэффициент теплопроводности материала цилиндра, теплоемкость материала цилиндра, плотность материала цилиндра, коэффициент теплоотдачи, время нагрева цилиндрам, температура печи, начальная температура цилиндра до нагрева, конечная температура нагрева цилиндра.

# **1 ПОСТАНОВКА ЗАДАЧИ, ПРОВЕРКА КОРРЕКТНОСТИ АЛГОРИТМА РАСЧЕТА И ОРГАНИЗАЦИЯ РАБОТЫ НАД ПРОЕКТОМ В СИСТЕМЕ ATLASSIAN JIRA**

## **1.1 Физическая постановка задачи**

Необходимо разработать программный продукт, позволяющий выполнять расчет нагрева неограниченного цилиндра при граничных условиях III рода.

Нагрев или охлаждение любого твердого тела сопровождается изменением его температурного поля в течение времени воздействия на тело теплового потока. Это явление носит название нестационарной теплопроводности. Решение инженерных задач нестационарной теплопроводности связано с определением температурного поля тела при заданном времени воздействия теплового потока или нахождением времени тепловой обработки тела при достижении температурные полем заданного по технологии значения. Такие задачи решают с помощью дифференциального уравнения теплопроводности Фурье, вид которого зависит от формы тела и применяемой системы координат.

Для тела типа неограниченного цилиндра дифференциальное уравнение теплопроводности в системе цилиндрических координат записывают так:

Решение приведенных уравнений может осуществиться только при конкретизации математического описания связи между временными и пространственными изменениями температуры в любой точке тела. Такая конкретизация происходит при введении краевых условий, когда к основному дифференциальному уравнению теплопроводности присоединяют дополнительные условия, включающие форму и размеры тела, его теплофизические характеристики (коэффициент теплопроводности, удельную теплоемкость, плотность), а также уравнения, описывающие начальное температурное поле тела в момент воздействия на него тепловым потоком и граничные условия. Последние отображают условия теплового взаимодействия поверхности тела с окружающей средой. Для решения дифференциального уравнения теплопроводности Фурье используют граничные условия I, II, III и IV рода. [2].

## **1.2 Математическая модель задачи**

Математическая модель решения поставленной задачи реализуется на основе методики расчета нагрева неограниченного цилиндра при граничных условиях III рода. Граничные условия III рода устанавливают зависимость плотности теплового потока вследствие теплопроводности тела от температуры поверхности тела и окружающей среды:

где αΣ - суммарный коэффициент теплообмена между средой и поверхностью тела;

tср - температура среды;

tпов - температура поверхности тела.

Нагреваемое (охлаждаемое) тело может быть принято, как неограниченный цилиндр, если его длина превосходит диаметр минимум в 3,5 раза. Совместное решение дифференциального уравнения теплопроводности для неограниченного цилиндра, плотности теплового потока при ГУ 3-го рода и уравнения начальных условий, при которых исходное поле температур тела равномерное и постоянное по всему его сечению, позволяет создать универсальное решение дифференциального уравнения теплопроводности в критериальной форме:

(1)

где 𝛩п, 𝛩м, 𝛩с – безразмерные (относительные) разности температур поверхности, массы и центра (оси) тела:

𝐹𝑜 – число (критерий) Фурье (), связывающее время τ, размер тела R и его коэффициент температуропроводности . Коэффициенты Pц, Мц, Nц, μц2 являются функциями числа БиО () и находятся по таблицам (см. прил. В). [2].

Для нахождения времени нагрева цилиндра необходимо преобразовать формулу расчета относительной разности температур на поверхности цилиндра:

где tпеч – температура в печи;

tнач – температура цилиндра в начале нагрева.

Затем, решая систему (1) относительно числа Фурье, определяется формула нахождения времени нагрева [1]:

Для лучшего понимания задачи была разработана блок-схема математической модели, представленная на рисунке 1.1.

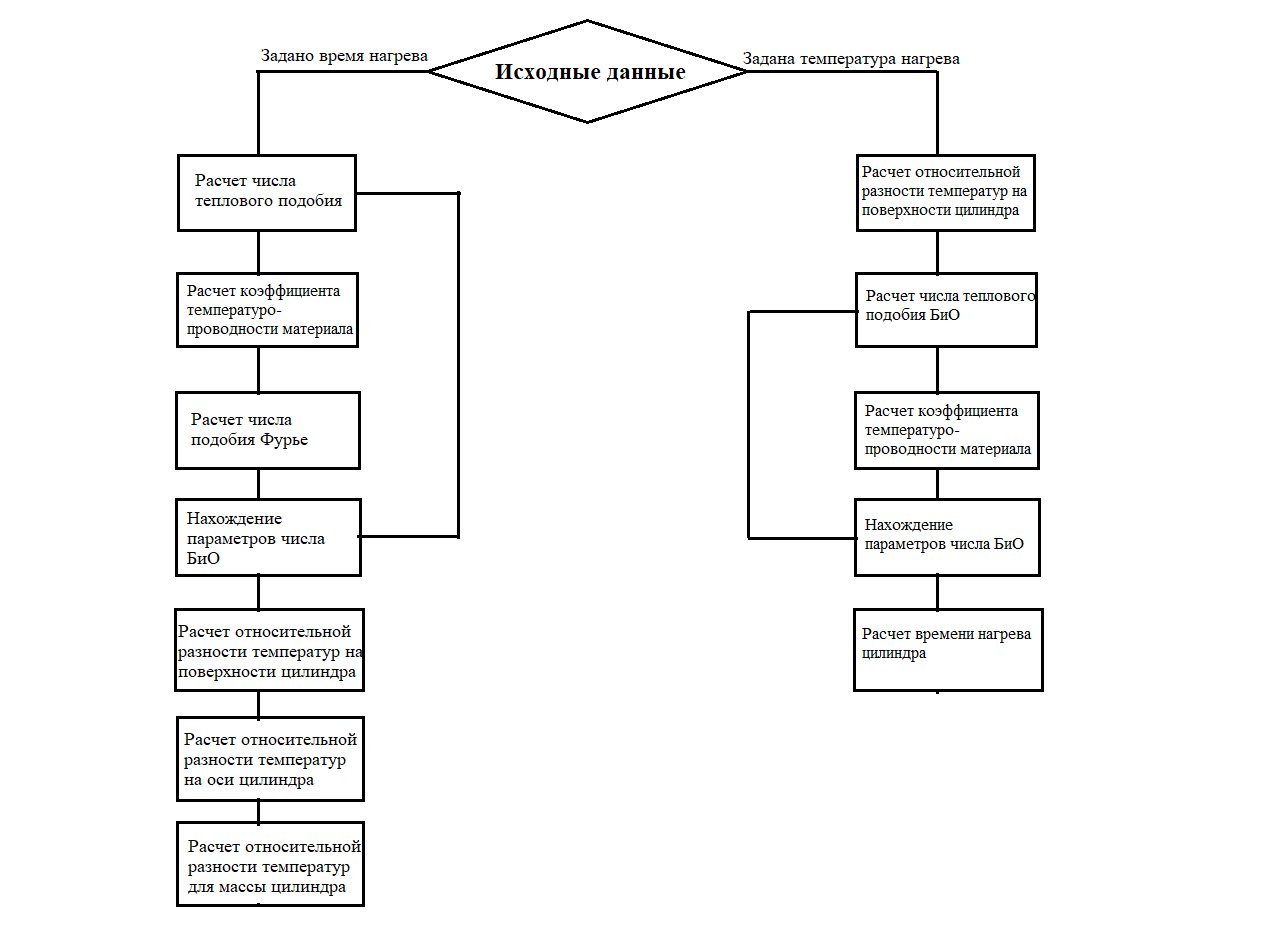


Рисунок 1.1 – Блок-схема математической модели

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

Для разработки расчета и проверки его корректности необходимы методика расчета и пример расчетной задачи. В пакете MS Excel были разработаны таблицы «Исходные данные», «Результаты расчета» и «Справочная информация» (рисунок 1.2) и внесены данные с примера расчетной задачи с соответствующими расчетными формулами. После выполнения расчета встроенным пакетом Excel, сверим цифры на экране с цифрами из расчетной задачи. Если они совпадают – расчет верный.

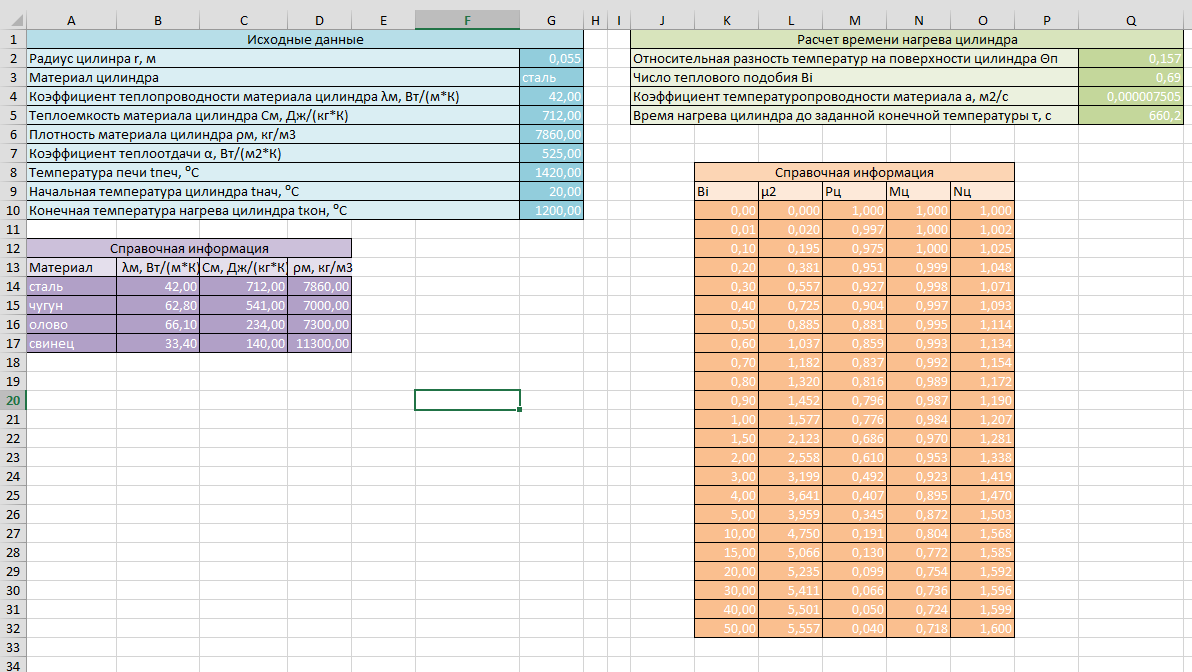


Рисунок 1.2 – Расчетные таблицы в пакете Microsoft Excel

Для решения поставленных задач необходимо было разработать в пакете так же возможность выбора материала изделия и сопутствующих параметров (плотность, коэффициент теплопроводности, теплоемкость). Для этого была создана вспомогательная таблица «Справочная информация», содержащая эти данные. Реализация осуществлялась по средствам раскрывающихся списков и применением функции ВПР. Благодаря функционалу пакета, с изменением чисел в исходных ячейках, автоматически изменяется и результат расчета. Результат ее выполнения представлен на рисунке 1.3. Аналогичным способом функция ВПР была применена в формулах расчета относительных разностей температур, для выбора параметров числа BiO (Рц, Мц, Nц, μц).

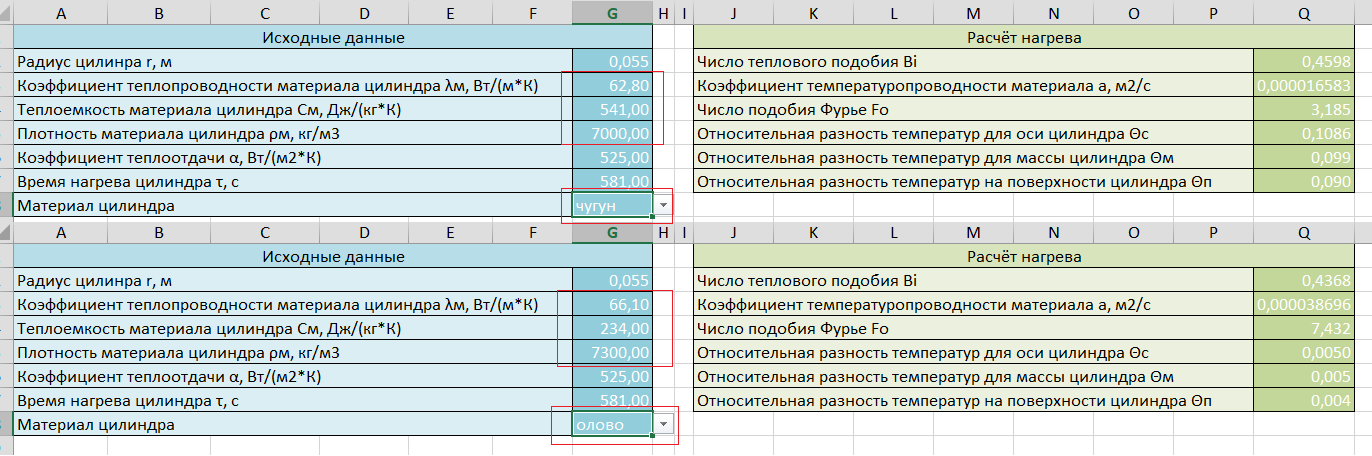


Рисунок 1.3 – Результат выполнения функции ВПР

# **2 ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

## **2.1 Разработка архитектуры системы**

Разработка архитектуры системы – один из важнейших начальных этапов работы, ведь грамотно разработанная архитектура (Рисунок 2.1) исключает возникновение множества ошибок в процессе работы, даёт правильное понимание внутренних процессов системы и взаимодействия с ней конечного пользователя.

Взаимодействие конечного пользователя с программой осуществляется посредствам графического интерфейса. В соответствии с требованиями, в нём предусмотрена возможность ввода, корректировки и удаления исходных данных, а также формирование отчёта по расчётным значениям. На интерфейсе отображаются результаты расчётов как в численном, так и в отчётном виде.

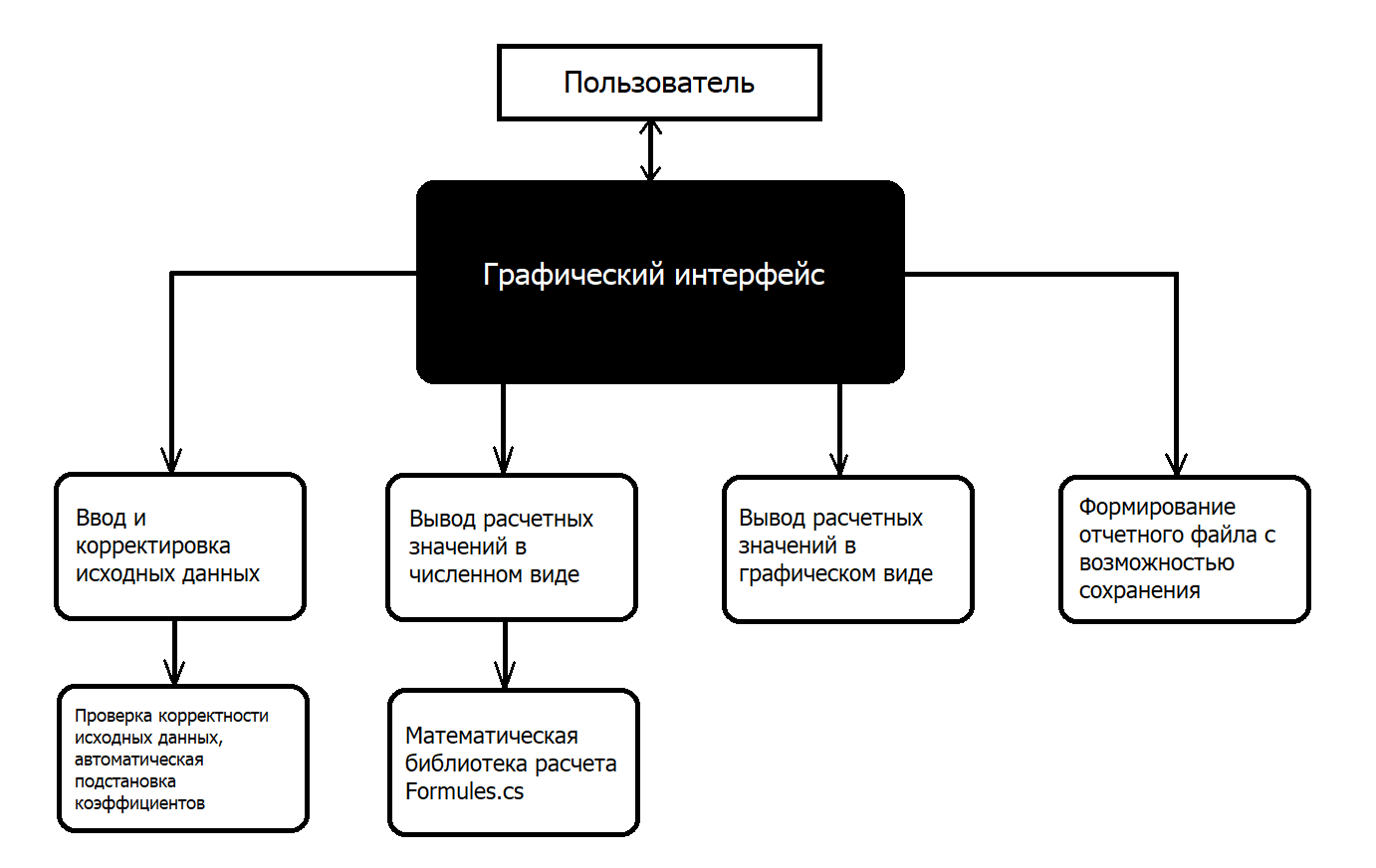


Рисунок 2.1 – Архитектура системы

## **2.2 Разработка блок-схемы работы пользователя с программой**

Блок схема работы пользователя с программой реализована в графическом редакторе Microsoft Visio в формате DFD (диаграмма потоковых данных), представленная на рисунке 2.2.

Взаимодействие пользователя с программой начинается с ввода исходных данных. Затем происходит обработка и расчёт значений посредствам обращения к математической библиотеке, после чего расчётные значения выводятся на пользовательский интерфейс в виде числовых данных, графика или же в форме отчёта. При повторном введении данных все вышеперечисленные действия повторяются.

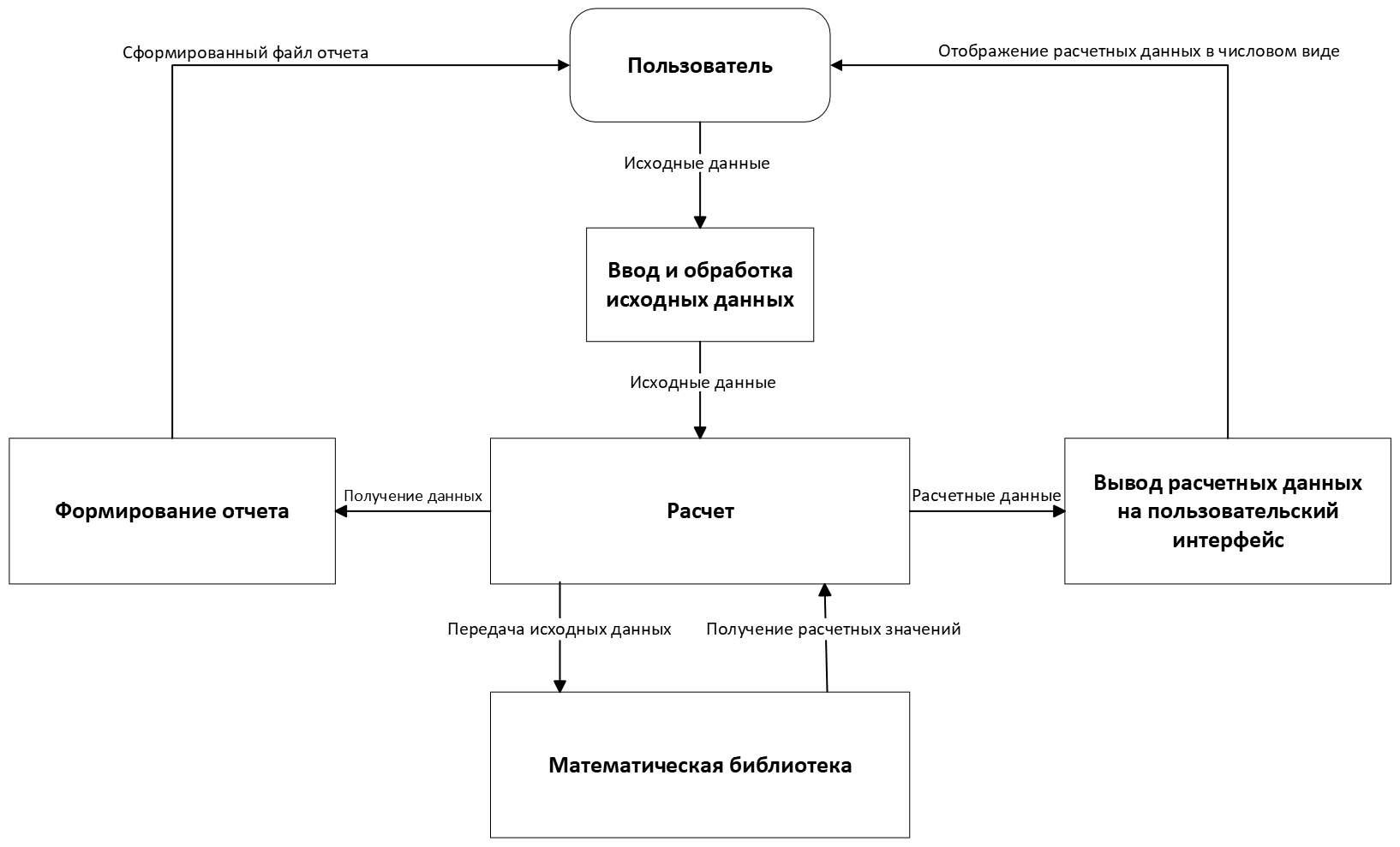


Рисунок 2.2 – Блок-схема работы пользователя с программой

## **2.3 Создание программного обеспечения в системе управления версиями GitHub**

Системы контроля версий позволяют разработчикам сохранять все изменения, внесённые в код. Поэтому в случае возникновения непоправимых ошибок или случайном удалении файла они могут просто откатить код до рабочего состояния вместо того, чтобы тратить часы на поиски маленькой ошибки или ошибок, ломающих весь код. Системы контроля версий также дают возможность нескольким разработчикам работать над одним проектом и сохранять внесённые изменения, чтобы убедиться, что все могут следить за тем, над чем они работают.

Данные возможности полностью предоставляет система управления версиями GitHub – сервис онлайн-хостинга репозиториев, обладающий всеми функциями распределённого контроля версий и функциональностью управления исходным кодом — всё, что поддерживает Git. Используется вместе с Git и даёт разработчикам возможность сохранять их код онлайн, а затем взаимодействовать с другими разработчиками в разных проектах. Работа с системой осуществлялась посредствам графического интерфейса настольного приложения GitHub Desktop (рисунок 2.3). [10].

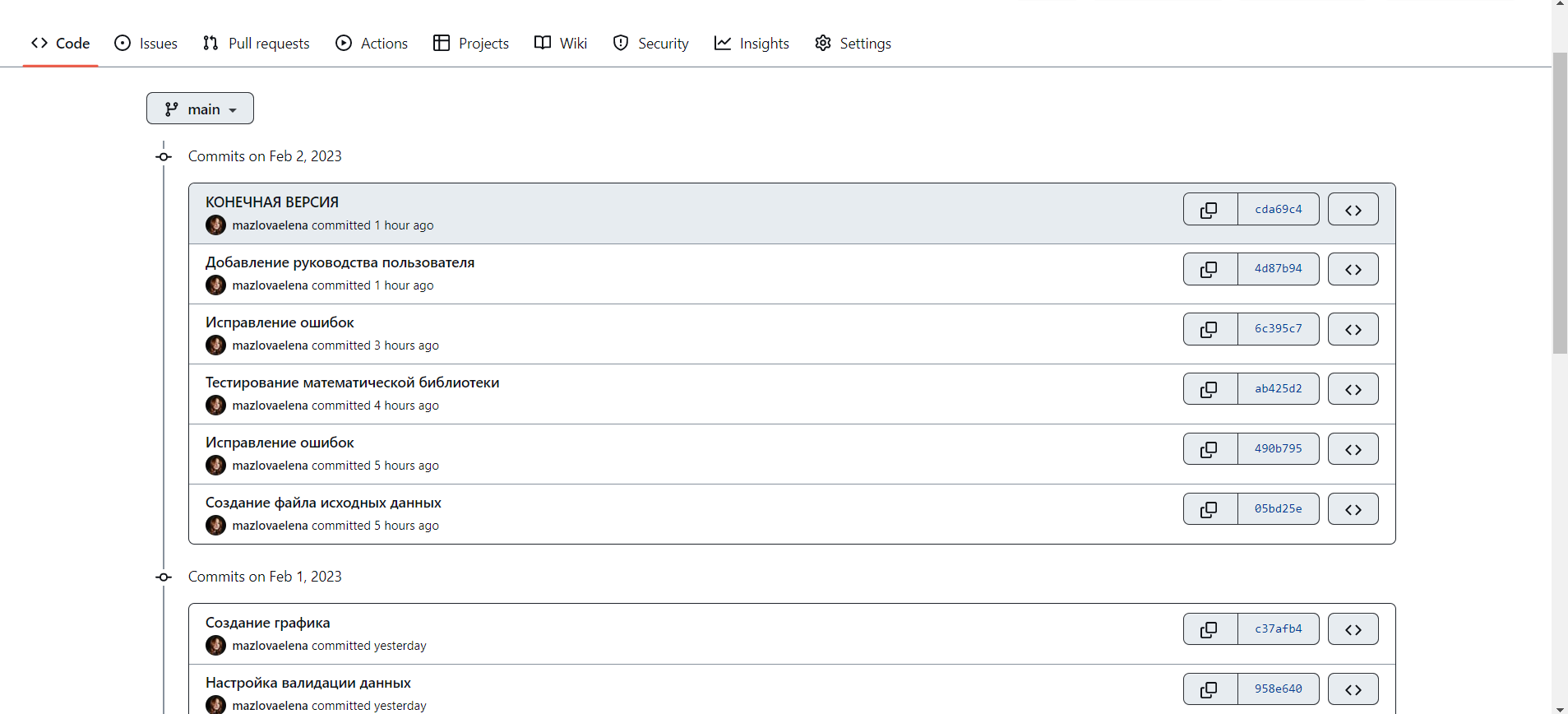


Рисунок 2.3 – Репозиторий проекта в сервисе GitLab

Ссылка на репозиторий в системе GitHub: <https://github.com/mazlovaelena/HeatingCilindr>.

## **2.4 Разработка математической библиотеки**

Ввиду небольшого объема расчетных формул математическая библиотека содержит всего 1 класс – Formules. Класс содержит в себе переменные исходных данных, расчетные формулы и достаточно объемную часть – switch-case – содержащую в себе справочные данные.

Для удобного ориентирования в классе библиотеки он был условно поделен на 6 частей (рисунок 2.4). Первые две части – исходные данные для обеих задач, затем справочный блок выбора параметров числа БиО, расчет промежуточных параметров и расчеты температуры и времени нагрева. Отдельно выделен экземпляр класса библиотеки, это необходимо для дальнейшего удобства вызова методов расчета.



Рисунок 2.4 – Разделы математической библиотеки

## **2.5 Реализация пользовательского интерфейса**

Реализация пользовательского интерфейса осуществлялась с использованием инструментов создания WPF-приложений (язык разметки xaml) и языка программирования C# для написания функционала элементов пользовательских форм.

Пользовательский интерфейс представляет собой окно с навигационной панелью кнопочного типа. Изменяющиеся вкладки реализованы с помощью компонента «UserControl» и помещены в родительский контейнер «Window». Первая вкладка «Информация» (рисунок 2.5) содержит общие сведения о принципах расчета нагрева цилиндрических заготовок и назначении программного обеспечения. Также страница содержит кнопку «Руководство пользователя», позволяющую посмотреть руководство по использованию системы.

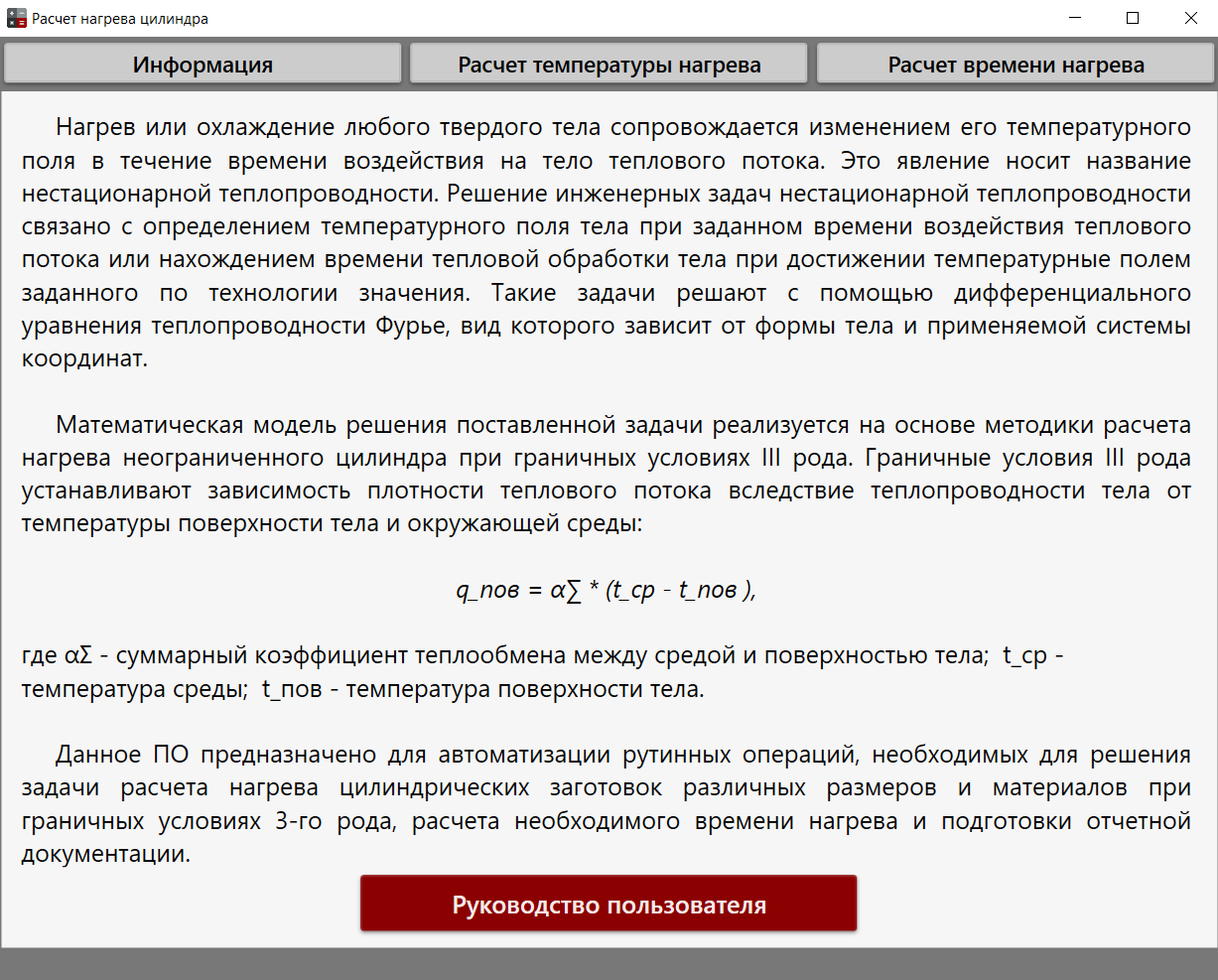


Рисунок 2.5 – Вкладка «Информация»

Далее пользователь выбирает тип решаемой задачи. Вкладки «Расчет температуры нагрева» (рисунок 2.6) и «Расчет времени наргева» содержат поля для возможности ввода исходных данных пользователя и кнопку «Расчёт».

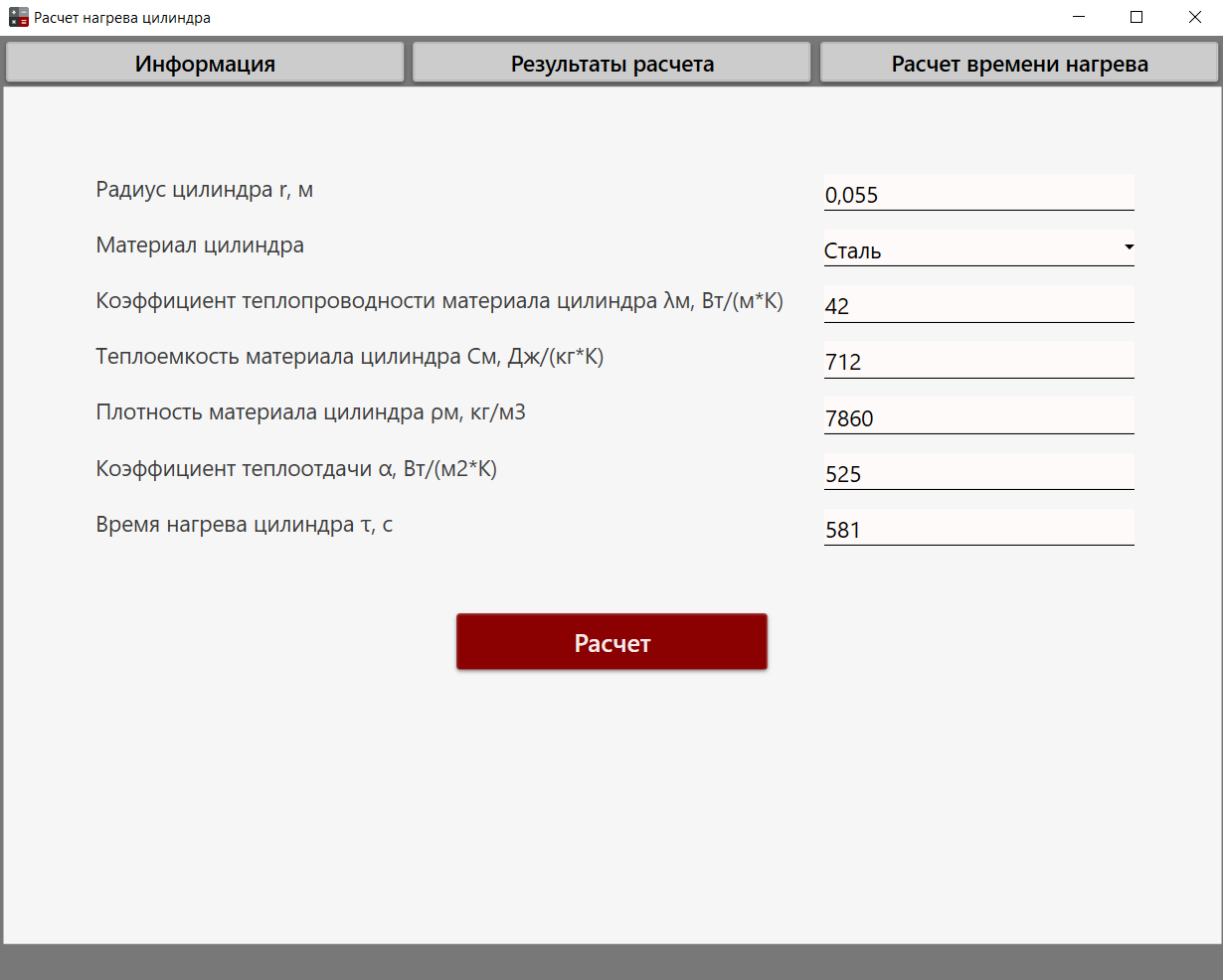


Рисунок 2.6 – Вкладка «Расчет температуры нагрева»

Вкладки исходных данных содержат раскрывающиеся списки для возможности выбора материала цилиндрической заготовки. При выборе определенного материала в поля ввода коэффициентов (теплопроводность, теплоемкость, плотность) происходит автоматическая подстановка соответствующих значений. Данный функционал реализован в классах .xaml.cs. Фрагмент кода реализации данного функционала представлен в приложении Б.

При переходе на вкладку с исходными данными задачи, кнопки навигационной панели автоматически изменяются на вывод результатов расчета (см. рисунок 2.6). При переходе к результатам расчета открывается вкладка, содержащая числовой вывод расчетных показателей, а также для задачи расчета температуры нагрева рассчитанные данные выводятся в графическом виде (рисунок 2.7). Вкладки дополнительно содержат кнопку сохранения отчетного файла. Код формирования отчетного файла представлен в приложении В.

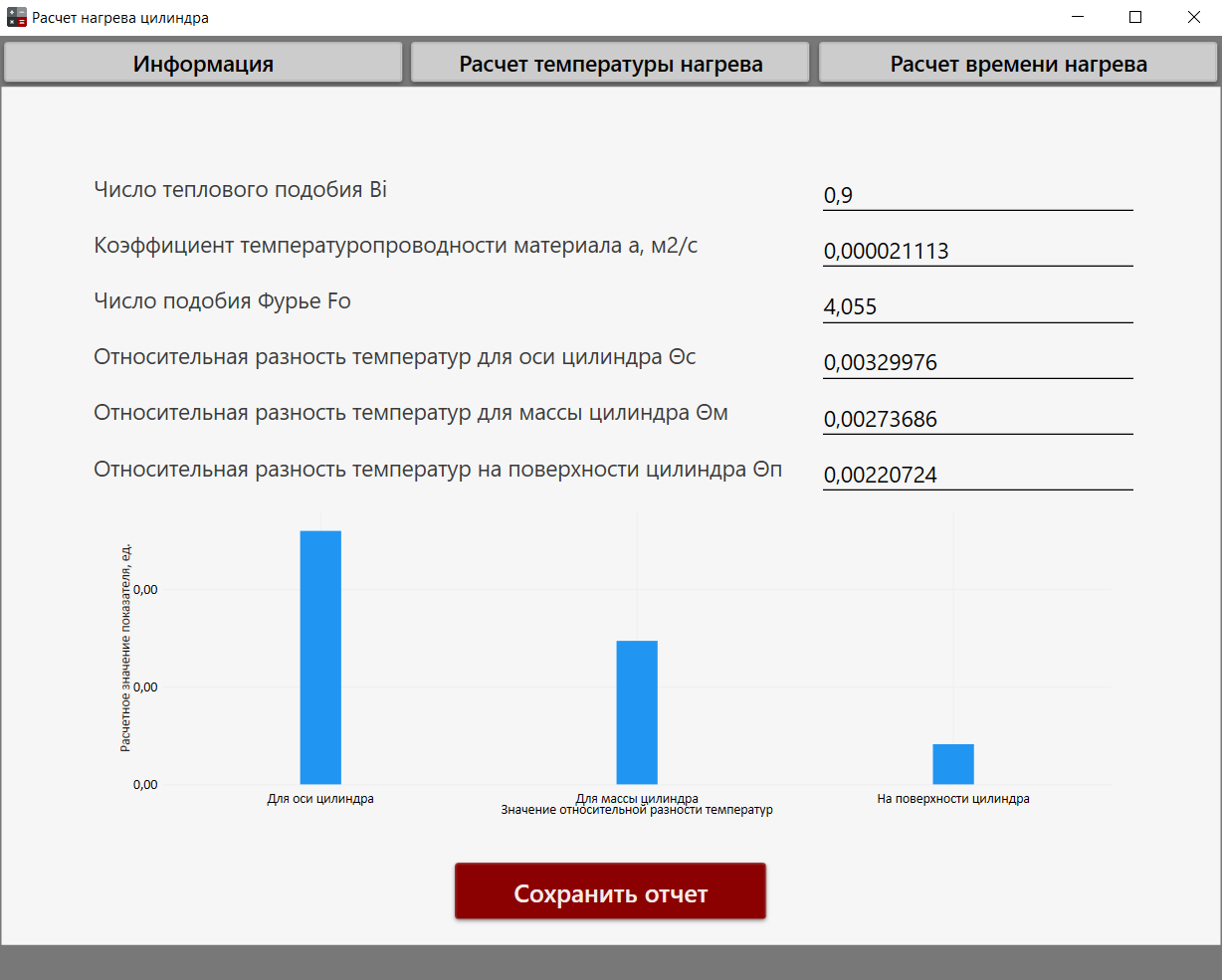


Рисунок 2.7 – Вкладка «Результаты расчёта»

## **2.6 Обработка исключительных ситуаций**

Обработка исключительных ситуаций играет важную роль для конечного использования приложения. Важно предусмотреть и исключить все ситуации, при которых возможно аварийное завершение работы приложения. В данном случае были реализованы следующие обработки исключительных ситуаций:

* Исключена возможность ввода букв, знаков или символов не числового формата;
* Исключена возможность ввода пустого значения (рисунок 2.8);
* Исключена возможность ошибочного внесения пробелов.

Фрагмент кода реализации обработки исключительных ситуаций представлен в приложении Б.

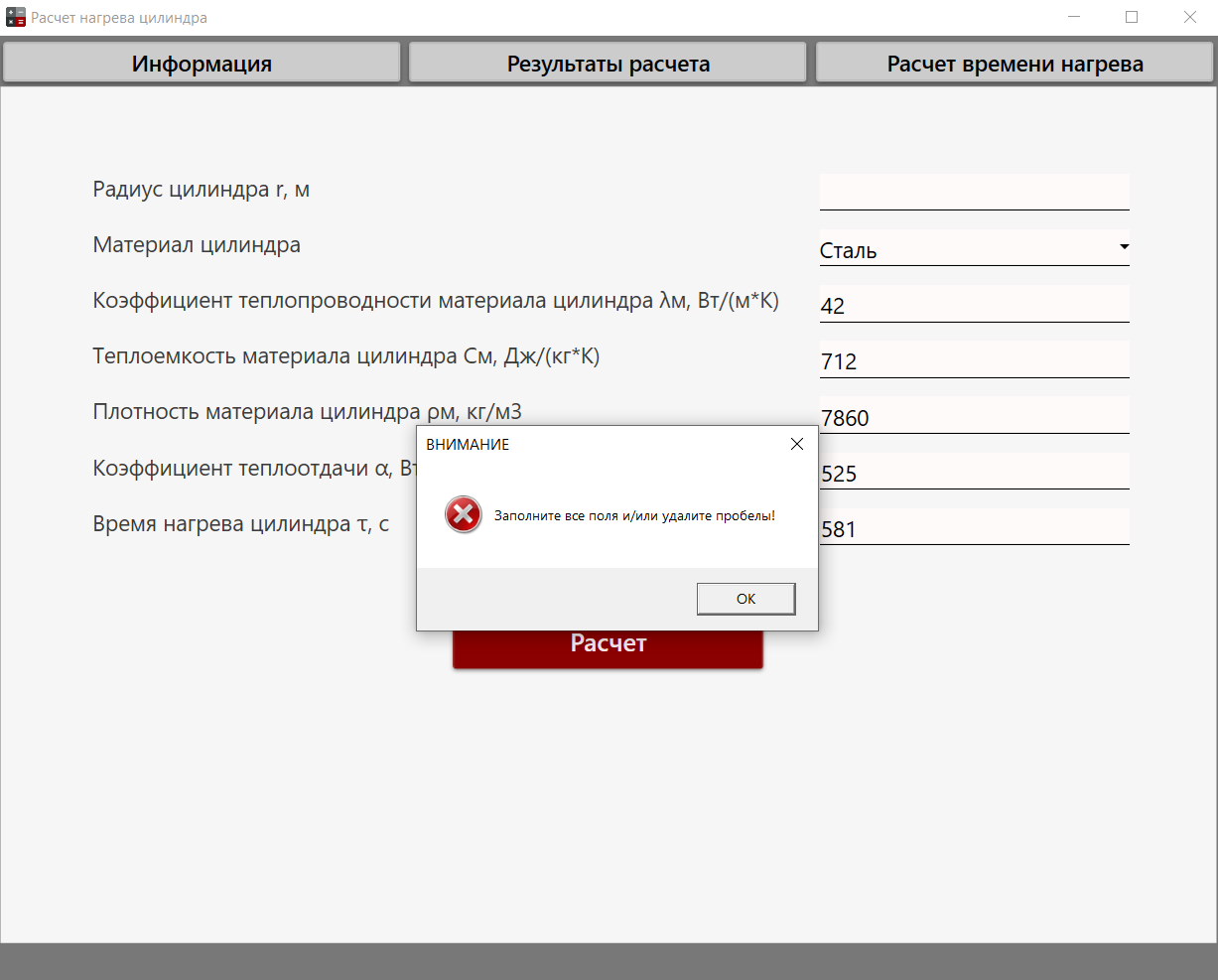


Рисунок 2.8 – Обработка исключения ввода пустого значения

## **2.7 Создание справочной помощи**

Для упрощения использования программы со стороны конечного пользователя, было разработано руководство пользователя, содержащее информацию об основном функционале, инструкцию по установке приложения и его дальнейшему использованию (рисунок 2.9). Реализация руководства пользователя была выполнена в онлайн-сервисе Canva, после чего файл был экспортирован в формат PDF.

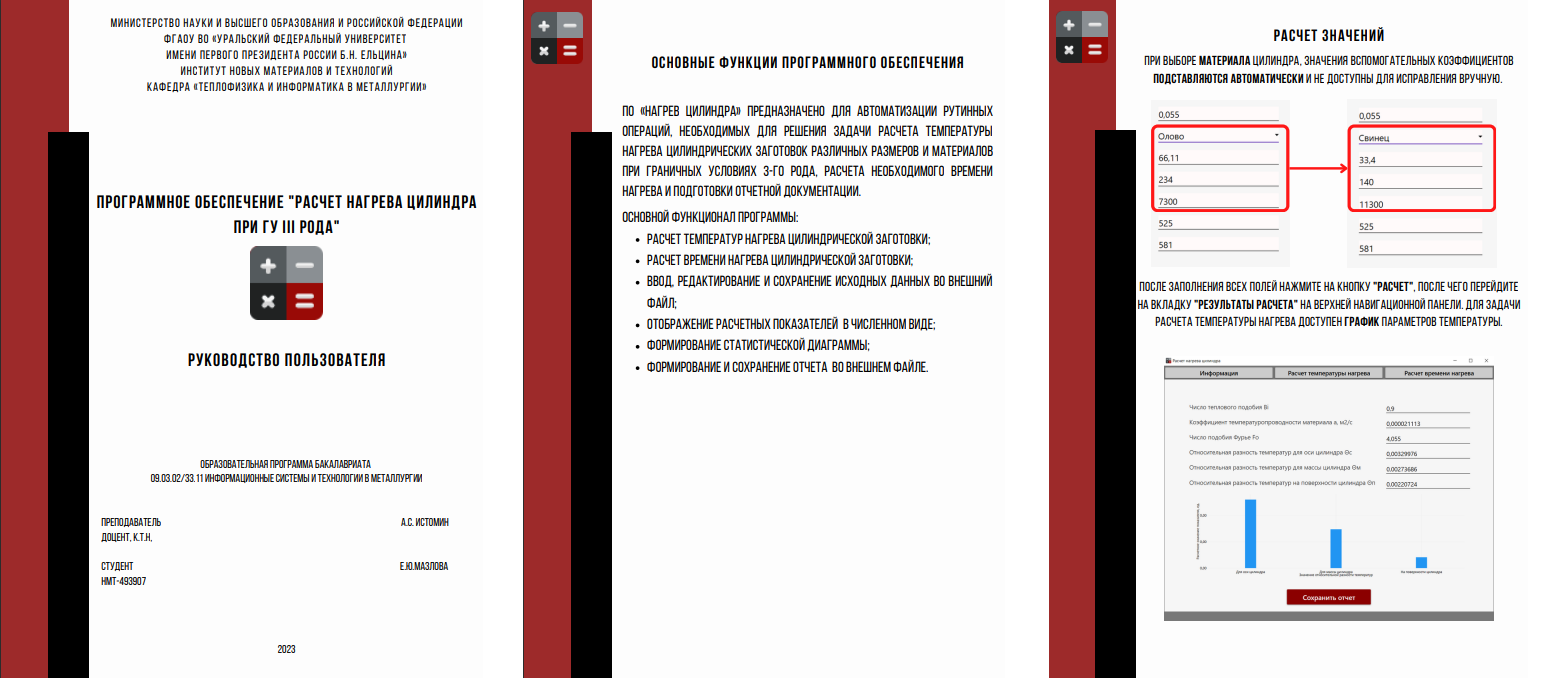


Рисунок 2.9 – Фрагменты руководства пользователя

## **2.8 Создание дистрибутива**

Для создания дистрибутива установки была использована утилита ActualInstaller. Данный продукт содержит обширный функционал даже в бесплатной версии и позволяет создавать простые, но в то же время функциональные файлы-установщики приложений. Пользователю достаточно указать название, версию и путь к папке exe-файла приложения и утилита скомпилирует дистрибутив. Этапы работы с данной утилитой представлены на рисунке 2.10.

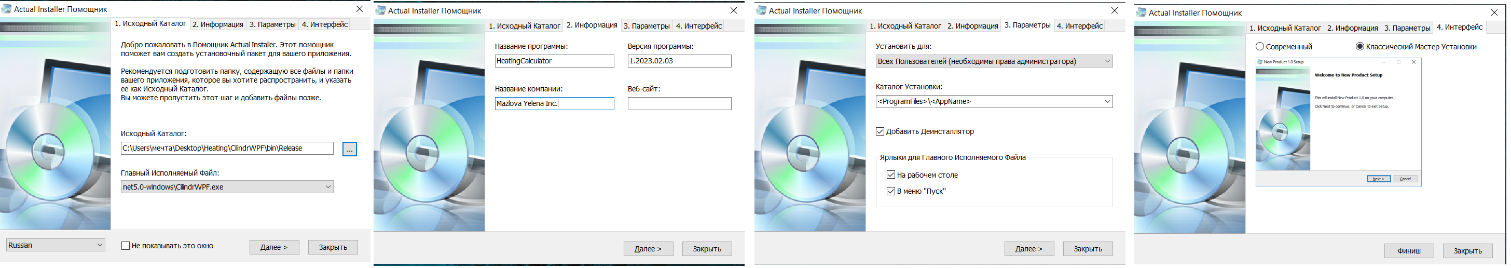


Рисунок 2.10 – Создание установочного дистрибутива с использованием утилиты ActualInstaller.

**3 РАЗРАБОТКА СИСТЕМЫ АВТОМАТИЗИРВАННОГО ТЕСТИРОВАНИЯ МАТЕМАТИЧЕСКОЙ БИБЛИОТЕКИ**

## **3.1 Проектирование системы**

Процесс тестирования программы во время разработки имеет большое значение, поскольку на данном этапе позволяет исключить ошибки в алгоритме расчёта. В данном случае тестирование математической библиотеки производилось путём сравнения её значений с данными формализованного расчёта в Microsoft Excel.

Проект тестов содержит в себе класс непосредственного тестирования «UnitTests», хранящий в себе тестовые методы всех функций математической библиотеки, принимающих участие в расчёте, а так же Excel-файл для сравнения результатов тестирования.

## **3.2 Реализация системы**

Реализация процесса тестирования производилась посредствам использования тестового проекта UnitTest.NET Core. Реализация программного кода тестирования включала в себя инициализацию переменных, подключение расчетного файла, считывание необходимых значений с ячеек таблицы и сравнение с полученными в программном расчете значениями. Все эти операции были выполнены с использованием конструкции try-catch-finally во избежание возможных исключений. Код тестового класса представлен в приложении Г. Результат выполнения тестирования представлен на рисунке 3.1.

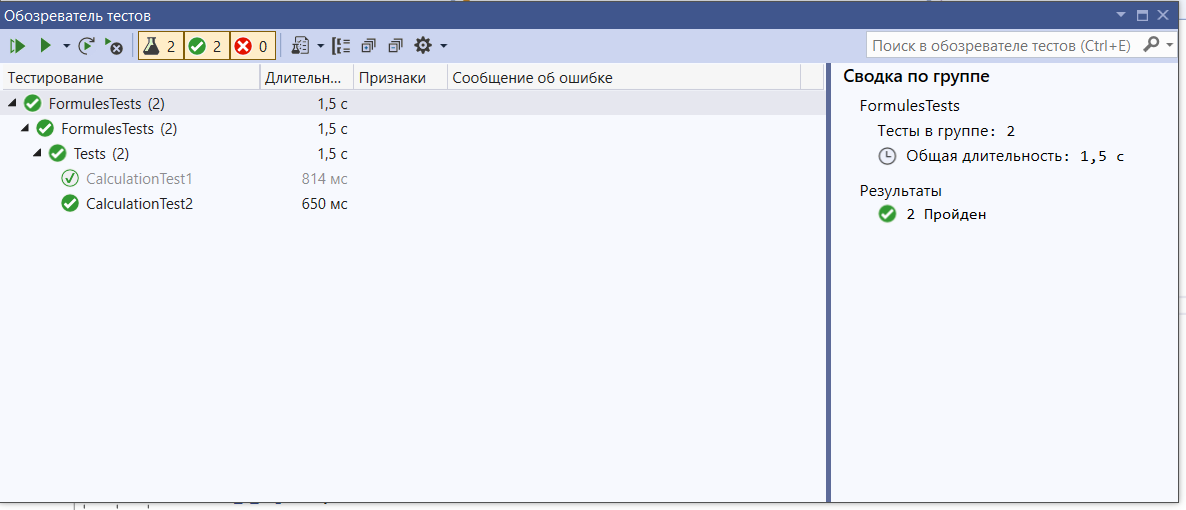


Рисунок 3.1 – Результат успешного выполнения тестирования

# **4 ОПИСАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

## **4.1 Установка и настройка программного средства**

Установка программы проста и понятна (рисунок 4.1). При запуске установочного дистрибутива пользователю необходимо указать папку установки, затем выбрать где создать ярлыки запуска приложения и приложение установится на ПК. Взаимодействие с файлом-установщиком представлено на рисунке 4.1.

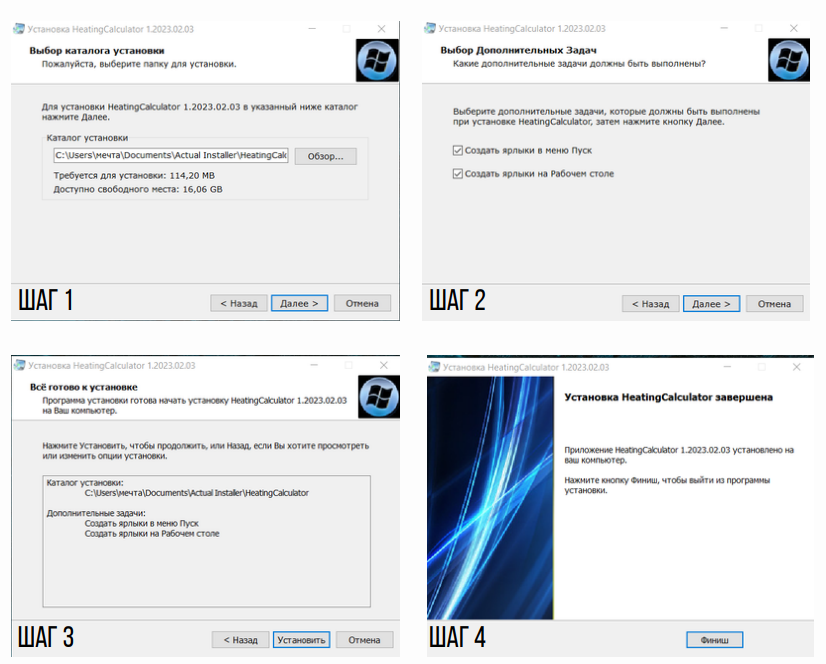


Рисунок 4.1 – Установка приложения

## **4.2 Функциональные возможности программного продукта**

В процессе разработки программного продукта были реализованы следующие функциональные возможности:

* Ввод и корректировка исходных данных с возможностью их сохранения для расчета во внешнем конфигурационном файле, который восстанавливается в программе при новом сеансе работы пользователя с программой;
* Выполнение расчетов и отображение результатов в численном и графическом виде на пользовательской форме;
* Формирование отчета по проведенным расчетам;
* Возможность экспорта отчета в другие приложения.

## **4.3 Технология выполнения расчетов**

Расчёт промежуточных и результативных значений происходит при обработке события нажатия на соответствующую кнопку на пользовательском интерфейсе. Введённые пользователем исходные данные проверяются на корректность, а затем передаются в математическую библиотеку посредствам создания её экземпляра, после чего происходит вызов необходимых расчётных функций и методов. Помимо этого, при обработке данных происходит их запись во внешний файл для вывода при следующем сеансе. В результате успешного расчета полученные данные отображаются в соответствующих полях на форме, а также передаются в метод отображения графика.

# **ЗАКЛЮЧЕНИЕ**

В ходе работы над проектом была разработана информационно-моделирующая система «Расчёт нагрева цилиндра» в среде Microsoft Visual Studio, в соответствии со всеми установленными требованиями, с использованием системы разработки настольных приложений WPF в рамках ознакомления с основными этапами разработки программного обеспечения.

Процесс разработки предусматривал постановку задачи, проверку корректности алгоритма расчёта, организацию работы в системе отслеживания ошибок, проектирование и реализацию программного обеспечения, а также разработку системы автоматизированного тестирования математической библиотеки, используемой в ходе расчётов.

Создание подобных информационно-моделирующих систем способствует процессу внедрения и совершенствования информационных систем и технологий в промышленных комплексах и на предприятиях, которые, в свою очередь, обеспечивают увеличение эффективности всех отраслей производства.

Предполагается, что внедрение данного программного продукта значительно упростит процесс расчета температуры и времени нагрева неограниченного цилиндра при ГУ III рода не только в рамках университетской программы, но и для реальных металлургических комплексов, тем самым значительно сократив не только трудозатраты, но и материальные ресурсы предприятий.

# **БИБЛИОГРАФИЧЕСКИЙ СПИСОК**

1. Юркинский В.П. Теплофизика. Сборник задач по тепломассопереносу: Учеб. пособие. СПб.: Изд-во Политехн. ун-та, 2012. – 129 с.
2. Расчеты по теплообмену
3. ГОСТ 24104-85 ЕСС АСУ «Автоматизированные системы управления. Общие требования».
4. ГОСТ 24104-86 ЕСС АСУ «Автоматизированные системы управления. Стадии создания».
5. ГОСТ 34.201-89 «Информационная технология. Комплекс стандартов на автоматизированные системы. Виды, комплектность и обозначение документов при создании автоматизированных систем».
6. ГОСТ 34.602-89 «Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы».
7. ГОСТ 34.602-89 «Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы».
8. ГОСТ 34.603-92 «Информационная технология. Комплекс стандартов на автоматизированные системы. Виды и порядок освоения».
9. Википедия: Azure DevOps Server. Полный текст: <https://ru.wikipedia.org/wiki/Azure_DevOps_Server>
10. Википедия: GitHub. Полный текст: <https://ru.wikipedia.org/wiki/GitHub>
11. Документация по ASP.NET. Полный текст: <https://docs.microsoft.com/ru-ru/aspnet/core/?view=aspnetcore-5.0>
12. Служба приложений Azure. Полный текст: <https://azure.microsoft.com/ru-ru/services/app-service/>
13. Chart.js – Линейная диаграмма. Полный текст: <https://www.chartjs.org/docs/3.4.1/charts/line.html>
14. Бова В.В. Основы проектирования информационных систем и технологий: учебное пособие / Бова В.В., Кравченко Ю.А. – Ростов-на-Дону; Таганрог: Издательство Южного федерального университета, 2018. – 106 с. <http://biblioclub.ru/index.php?page=book_red&id=499515&sr=1>
15. Маглинец Ю.А. Анализ требований к автоматизированным информационным системам: учебное пособие. –  М.: Интернет-Университет Информационных Технологий, 2008. – 200 с. <http://biblioclub.ru/index.php?page=book_red&id=439330&sr=1>
16. Стасышин В.М. Проектирование информационных систем и баз данных: учебное пособие. – Новосибирск: НГТУ, 2012. – 100 с. <http://biblioclub.ru/index.php?page=book_red&id=228774&sr=1>
17. Гамма Э. Приемы объектно-ориентированного проектирования. Паттерны проектирования [Текст]: [пер. с англ.] / Э.Гамма, Р.Хелм. – СПб.: Питер, 2007. – 368 с.
18. Лошкарев Н.Б. Указания к оформлению дипломных и курсовых проектов и работ. Методические указания [Текст] / Н.Б.Лошкарев, А.Н.Лошкарев, Л.А.Зайнуллин. – Екатеринбург: УГТУ – УПИ, 2007. – 49 с.

# **ПРИЛОЖЕНИЕ А СПРАВОЧНЫЕ ТАБЛИЦЫ**

Таблица А.1 - Коэффициенты для расчета времени нагрева (охлаждения) и температурного поля неограниченного цилиндра при граничных условиях III рода

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Bi | μ2 | Pц | Mц | Nц |
| 0,00 | 0,000 | 1,000 | 1,000 | 1,000 |
| 0,01 | 0,020 | 0,997 | 1,000 | 1,002 |
| 0,10 | 0,195 | 0,975 | 1,000 | 1,025 |
| 0,20 | 0,381 | 0,951 | 0,999 | 1,048 |
| 0,30 | 0,557 | 0,927 | 0,998 | 1,071 |
| 0,40 | 0,725 | 0,904 | 0,997 | 1,093 |
| 0,50 | 0,885 | 0,881 | 0,995 | 1,114 |
| 0,60 | 1,037 | 0,859 | 0,993 | 1,134 |
| 0,70 | 1,182 | 0,837 | 0,992 | 1,154 |
| 0,80 | 1,320 | 0,816 | 0,989 | 1,172 |
| 0,90 | 1,452 | 0,796 | 0,987 | 1,190 |
| 1,00 | 1,577 | 0,776 | 0,984 | 1,207 |
| 1,50 | 2,123 | 0,686 | 0,970 | 1,281 |
| 2,00 | 2,558 | 0,610 | 0,953 | 1,338 |
| 3,00 | 3,199 | 0,492 | 0,923 | 1,419 |
| 4,00 | 3,641 | 0,407 | 0,895 | 1,470 |
| 5,00 | 3,959 | 0,345 | 0,872 | 1,503 |
| 10,00 | 4,750 | 0,191 | 0,804 | 1,568 |
| 15,00 | 5,066 | 0,130 | 0,772 | 1,585 |
| 20,00 | 5,235 | 0,099 | 0,754 | 1,592 |
| 30,00 | 5,411 | 0,066 | 0,736 | 1,596 |
| 40,00 | 5,501 | 0,050 | 0,724 | 1,599 |
| 50,00 | 5,557 | 0,040 | 0,718 | 1,600 |

Таблица А.2 – Теплофизические свойства материалов

|  |  |  |  |
| --- | --- | --- | --- |
| Материал | λм, Вт/(м\*К) | См, Дж/(кг\*К) | ρм, кг/м3 |
| сталь | 42,00 | 712,00 | 7860,00 |
| чугун | 62,80 | 541,00 | 7000,00 |
| олово | 66,10 | 234,00 | 7300,00 |
| свинец | 33,40 | 140,00 | 11300,00 |

# **ПРИЛОЖЕНИЕ Б ФРАГМЕНТ ЛИСТИНГА ПРОГРАММНОГО КОДА КЛАССА TEMPINITIAL.XAML.CS**

namespace CilindrWPF

{

/// <summary>

/// Логика взаимодействия для TempInitial.xaml

/// </summary>

public partial class TempInitial : UserControl

{

public Formules Formules;

public ResultTemp TempHeat;

public TempInitial()

{

InitializeComponent();

List<double> textBoxesValues = new List<double>();

try

{

ReadInitialFile();

//Обработка ошибки при отсутствии файла исходных данных

textBoxesValues.Add(Convert.ToDouble(R.Text));

textBoxesValues.Add(Double.Parse(material.Text));

textBoxesValues.Add(Convert.ToDouble(lamdaM.Text));

textBoxesValues.Add(Convert.ToDouble(cM.Text));

textBoxesValues.Add(Convert.ToDouble(roM.Text));

textBoxesValues.Add(Convert.ToDouble(alfa.Text));

textBoxesValues.Add(Convert.ToDouble(t.Text));

}

catch

{

if (textBoxesValues.Count == 0)

{

GetDefaultValues();

}

}

}

//Правило валидации строки

public int Valid(string s)

{

string substr = System.Globalization.CultureInfo.CurrentCulture.NumberFormat.NumberDecimalSeparator[0].ToString();

int count = (s.Length - s.Replace(substr, "").Length) / substr.Length;

return count;

}

//Метод ограничения ввода в поле (не позволяет пользователю вводить буквы и лишние символы в поля ввода)

private void R\_PreviewTextInput(object sender, TextCompositionEventArgs e)

{

e.Handled = !(Char.IsDigit(e.Text, 0) || ((e.Text == System.Globalization.CultureInfo.CurrentCulture.NumberFormat.NumberDecimalSeparator[0].ToString()) && (Valid(((TextBox)sender).Text) < 1)));

double val;

if (!Double.TryParse(e.Text, out val) && e.Text != ",")

{

e.Handled = true;

}

}

//Метод сохранения исходных данных во внешний файл

public async void GetInitialFile()

{

Dictionary<string, object> Init = new Dictionary<string, object>();

var init = new Dictionary<string, object>()

{

{"r", R.Text },

{"material", material.Text },

{"lamdaM", lamdaM.Text },

{"cM", cM.Text },

{"roM", roM.Text},

{"alfa", alfa.Text },

{"t", t.Text },

};

try

{

using (StreamWriter fs = new StreamWriter("InputTemp.txt"))

{

string[] strArray = init.Select(x => ("" + x.Key + "=" + x.Value + ";")).ToArray();

foreach (string s in strArray)

{

char[] d = s.ToCharArray();

await fs.WriteAsync(d);

}

fs.Close();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Ошибка!", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

//Метод чтения данных из внешнего файла

public async void ReadInitialFile()

{

try

{

string path = "InputTemp.txt";

using (StreamReader fstream = new StreamReader(path))

{

string v = await fstream.ReadToEndAsync();

string d = v.ToString();

var dict = d.Split(';')

.Select(part => part.Split('='))

.Where(part => part.Length == 2)

.ToDictionary(sp => sp[0], sp => sp[1]);

R.Text = Convert.ToDouble(dict["r"]).ToString();

material.Text = Convert.ToString(dict["material"]);

lamdaM.Text = Convert.ToDouble(dict["lamdaM"]).ToString();

cM.Text = Convert.ToDouble(dict["cM"]).ToString();

roM.Text = Convert.ToDouble(dict["roM"]).ToString();

alfa.Text = Convert.ToDouble(dict["alfa"]).ToString();

t.Text = Convert.ToDouble(dict["t"]).ToString();

}

}

catch

{

}

}

//Метод подстановки соответсвующих коэффициентов при выборе материала

private void material\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

if(material.SelectedItem == Stal)

{

lamdaM.Text = Convert.ToString(42);

cM.Text = Convert.ToString(712);

roM.Text = Convert.ToString(7860);

}

else if(material.SelectedItem == Chugun)

{

lamdaM.Text = Convert.ToString(62.8);

cM.Text = Convert.ToString(541);

roM.Text = Convert.ToString(7000);

}

else if (material.SelectedItem == Olovo)

{

lamdaM.Text = Convert.ToString(66.11);

cM.Text = Convert.ToString(234);

roM.Text = Convert.ToString(7300);

}

else if(material.SelectedItem == Svinec)

{

lamdaM.Text = Convert.ToString(33.4);

cM.Text = Convert.ToString(140);

roM.Text = Convert.ToString(11300);

}

}

//Получение введенных значений для расчета

public void GetSourceValues()

{

double \_r = Convert.ToDouble(R.Text);

string \_material = Convert.ToString(material.Text);

double \_lamdaM = Convert.ToDouble(lamdaM.Text);

double \_cM = Convert.ToDouble(cM.Text);

double \_roM = Convert.ToDouble(roM.Text);

double \_alfa = Convert.ToDouble(alfa.Text);

double \_t = Convert.ToDouble(t.Text);

double \_tp = 0;

double \_tbegin = 0;

double \_tend = 0;

Formules = new Formules(\_r, \_lamdaM, \_cM, \_roM, \_alfa, \_t, \_material, \_tp, \_tend, \_tbegin);

}

//Метод вывода значений по умолчанию (при первом запуске программы)

public void GetDefaultValues()

{

R.Text = "0,055";

material.Text = "Сталь";

lamdaM.Text = "42";

cM.Text = "712";

roM.Text = "7860";

alfa.Text = "525";

t.Text = "581";

}

//Метод нажатия на кнопку "Расчет"

private void CalcTemp\_Click(object sender, RoutedEventArgs e)

{

List<string> textBoxesValues = new List<string>();

textBoxesValues.Add(R.Text);

textBoxesValues.Add(material.Text);

textBoxesValues.Add(lamdaM.Text);

textBoxesValues.Add(cM.Text);

textBoxesValues.Add(roM.Text);

textBoxesValues.Add(alfa.Text);

textBoxesValues.Add(t.Text);

//Проверка наличия пустых полей

bool IsValuesCorrect = true;

foreach (string s in textBoxesValues)

{

string[] filterEmpty = s.Split(new char[] { ' ' });

int counter = 0;

foreach (string n in filterEmpty)

{

if (n == "")

{

counter++;

}

}

if (counter >= 1 || s == String.Empty || s == ",")

{

MessageBox.Show("Заполните все поля и/или удалите пробелы!", "ВНИМАНИЕ", MessageBoxButton.OK, MessageBoxImage.Stop);

IsValuesCorrect = false;

counter = 0;

break;

}

}

if (IsValuesCorrect == true)

{

GetSourceValues();

GetInitialFile();

Formules.Bi();

Formules.A();

Formules.Fo();

Formules.TempDiff\_Surface();

Formules.TempDiff\_Mass();

Formules.TempDiff\_Centr();

ResultTemp.Formules = Formules;

TempHeat.GetCountValues();

}

}

}

}

# **ПРИЛОЖЕНИЕ В ФРАГМЕНТ ЛИСТИНГА ПРОГРАММНОГО КОДА КЛАССА RESULTTIME.XAML.CS**

namespace CilindrWPF

{

/// <summary>

/// Логика взаимодействия для ResultTime.xaml

/// </summary>

public partial class ResultTime : UserControl

{

public ResultTime()

{

InitializeComponent();

}

public static Formules Formules;

public TimeInitial Initial;

//Вывод расчетных значений показателей

public void GetCountValues()

{

Bi.Text = Math.Round(Formules.Bi(), 1).ToString();

A.Text = Math.Round(Formules.A(), 10).ToString("#,##0.000000000");

T\_DS.Text = Math.Round(Formules.Temp\_DiffSurface(), 8).ToString();

TH.Text = Math.Round(Formules.Time\_heath(), 0).ToString("#,##0");

}

//Метод создания отчетного файла Excel

public void CreateFile(string patch)

{

if (patch != string.Empty)

{

using (SLDocument Doc = new SLDocument())

{

Doc.SetCellValue("A1", "Исходные данные");

Doc.SetCellValue("A2", "Радиус цилиндра r, м");

Doc.SetCellValue("B2", Initial.R.Text);

Doc.SetCellValue("A3", "Материал цилиндра");

Doc.SetCellValue("B3", Initial.material.Text);

Doc.SetCellValue("A4", "Коэффициент теплопроводности материала цилиндра λм, Вт/(м\*К)");

Doc.SetCellValue("B4", Initial.lamdaM.Text);

Doc.SetCellValue("A5", "Теплоемкость материала цилиндра См, Дж/(кг\*К)");

Doc.SetCellValue("B5", Initial.cM.Text);

Doc.SetCellValue("A6", "Плотность материала цилиндра ρм, кг/м3");

Doc.SetCellValue("B6", Initial.roM.Text);

Doc.SetCellValue("A7", "Коэффициент теплоотдачи α, Вт/(м2\*К)");

Doc.SetCellValue("B7", Initial.alfa.Text);

Doc.SetCellValue("A8", "Температура печи tпеч, ⁰C");

Doc.SetCellValue("B8", Initial.t\_pech.Text);

Doc.SetCellValue("A9", "Начальная температура цилиндра tнач, ⁰C");

Doc.SetCellValue("B9", Initial.t\_begin.Text);

Doc.SetCellValue("A10", "Конечная температура нагрева цилиндра tкон, ⁰C");

Doc.SetCellValue("B10", Initial.t\_end.Text);

Doc.SetCellValue("D1", "Результаты расчета");

Doc.SetCellValue("D2", "Число теплового подобия Bi");

Doc.SetCellValue("E2", Bi.Text);

Doc.SetCellValue("D3", "Коэффициент температуропроводности материала а, м2/с");

Doc.SetCellValue("E3", A.Text);

Doc.SetCellValue("D4", "Время нагрева цилиндра до заданной конечной температуры τ, с");

Doc.SetCellValue("E4", TH.Text);

Doc.AutoFitColumn(1, 6);

Doc.SaveAs(patch);

}

}

}

//Метод нажатия на кнопку "Сохранить отчет"

private void ReportTime\_Click(object sender, RoutedEventArgs e)

{

try

{

SaveFileDialog saveFile = new SaveFileDialog();

saveFile.Title = "Выберите папку";

saveFile.DefaultExt = "xlsx";

if (saveFile.ShowDialog() == true)

{

CreateFile(saveFile.FileName);

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Ошибка!", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

}

}

# **ПРИЛОЖЕНИЕ Г ФРАГМЕНТ ЛИСТИНГА ПРОГРАММНОГО КОДА КЛАССА UNITTEST1.CS**

namespace FormulesTests

{

public class Tests

{

private string fileName = "ForTest.xlsx";

Microsoft.Office.Interop.Excel.Application objExcel = null;

Microsoft.Office.Interop.Excel.Workbook WorkBook = null;

private object oMissing = System.Reflection.Missing.Value;

[Test]

public void CalculationTest1()

{

double \_r = 0.055;

double \_lamdaM = 42;

double \_cM = 712;

double \_roM = 7860;

double \_alfa = 525;

double \_t = 581;

string \_material = "Сталь";

double \_t\_pech = 1420;

double \_t\_beg = 20;

double \_t\_end = 1200;

Formules Form = new Formules(\_r, \_lamdaM, \_cM, \_roM, \_alfa, \_t, \_material, \_t\_end, \_t\_beg, \_t\_pech);

try

{

objExcel = new Microsoft.Office.Interop.Excel.Application();

WorkBook = objExcel.Workbooks.Open(

Directory.GetCurrentDirectory() + "\\" + fileName);

Microsoft.Office.Interop.Excel.Worksheet WorkSheet = (Microsoft.Office.Interop.Excel.Worksheet)WorkBook.Sheets["Прямая задача"];

((Microsoft.Office.Interop.Excel.Range)WorkSheet.Cells[2, 7]).Value2 = Form.r;

((Microsoft.Office.Interop.Excel.Range)WorkSheet.Cells[3, 7]).Value2 = Form.lamdaM;

((Microsoft.Office.Interop.Excel.Range)WorkSheet.Cells[4, 7]).Value2 = Form.cM;

((Microsoft.Office.Interop.Excel.Range)WorkSheet.Cells[5, 7]).Value2 = Form.roM;

((Microsoft.Office.Interop.Excel.Range)WorkSheet.Cells[6, 7]).Value2 = Form.alfa;

((Microsoft.Office.Interop.Excel.Range)WorkSheet.Cells[7, 7]).Value2 = Form.t;

// отобразить в журнале тестирования

Console.WriteLine("--- Исходные данные");

Console.WriteLine("Радиус цилиндра: {0}", Form.r.ToString());

Console.WriteLine("Коэффициент теплопроводности материала цилиндра: {0}", Form.lamdaM.ToString());

Console.WriteLine("Теплоемкость материала цилиндра: {0}", Form.cM.ToString());

Console.WriteLine("Плотность материала цилиндра: {0}", Form.roM.ToString());

Console.WriteLine("Коэффициент теплоотдачи: {0}", Form.alfa.ToString());

Console.WriteLine("Время нагрева цилиндра: {0}", Form.t.ToString());

double Bi = double.Parse(((Microsoft.Office.Interop.Excel.Range)WorkSheet.Cells[2, 17]).Value.ToString());

double A = double.Parse(((Microsoft.Office.Interop.Excel.Range)WorkSheet.Cells[3, 17]).Value.ToString());

double Fo = double.Parse(((Microsoft.Office.Interop.Excel.Range)WorkSheet.Cells[4, 17]).Value.ToString());

double TempDiff\_Centr = double.Parse(((Microsoft.Office.Interop.Excel.Range)WorkSheet.Cells[5, 17]).Value.ToString());

double TempDiff\_Mass = double.Parse(((Microsoft.Office.Interop.Excel.Range)WorkSheet.Cells[6, 17]).Value.ToString());

double TempDiff\_Surface = double.Parse(((Microsoft.Office.Interop.Excel.Range)WorkSheet.Cells[7, 17]).Value.ToString());

Assert.AreEqual(Bi, Math.Round(Form.Bi(), 1), 0.001);

System.Diagnostics.Debug.WriteLine("Число теплового подобия - BI: expected =" +

Bi + "; actual=" + Math.Round(Form.Bi(), 1));

Assert.AreEqual(A, Math.Round(Form.A(), 7), 0.001);

System.Diagnostics.Debug.WriteLine("Коэффициент температуропроводности материала - А: expected =" +

A + "; actual=" + Math.Round(Form.A(), 7));

Assert.AreEqual(Fo, Math.Round(Form.Fo(), 3), 0.001);

System.Diagnostics.Debug.WriteLine("Число подобия Фурье - Fo: expected =" +

Fo + "; actual=" + Math.Round(Form.Fo(), 3));

Assert.AreEqual(TempDiff\_Centr, Math.Round(Form.TempDiff\_Centr(), 4), 0.001);

System.Diagnostics.Debug.WriteLine("Разность температур на оси цилиндра: expected =" +

TempDiff\_Centr + "; actual=" + Math.Round(Form.TempDiff\_Centr(), 4));

Assert.AreEqual(TempDiff\_Mass, Math.Round(Form.TempDiff\_Mass(), 4), 0.001);

System.Diagnostics.Debug.WriteLine("Разность температур для массы цилиндра: expected =" +

TempDiff\_Mass + "; actual=" + Math.Round(Form.TempDiff\_Mass(), 4));

Assert.AreEqual(TempDiff\_Surface, Math.Round(Form.TempDiff\_Surface(), 4), 0.001);

System.Diagnostics.Debug.WriteLine("Разность температур на поверхности цилиндра: expected =" +

TempDiff\_Surface + "; actual=" + Math.Round(Form.TempDiff\_Surface(), 4));

// отобразить в журнале тестирования

Console.WriteLine("");

Console.WriteLine("--- Результаты расчета");

Console.WriteLine("Число теплового подобия, метод Bi(): expected = " +

Bi + "; actual=" + Math.Round(Form.Bi(), 1));

// отобразить в журнале тестирования

Console.WriteLine("");

Console.WriteLine("--- Результаты расчета");

Console.WriteLine("Коэффициент температуропроводности материала, метод A(): expected = " +

A + "; actual=" + Math.Round(Form.A(), 7));

// отобразить в журнале тестирования

Console.WriteLine("");

Console.WriteLine("--- Результаты расчета");

Console.WriteLine("Число подобия Фурье, метод Fo(): expected = " +

Fo + "; actual=" + Math.Round(Form.Fo(), 3));

// отобразить в журнале тестирования

Console.WriteLine("");

Console.WriteLine("--- Результаты расчета");

Console.WriteLine("Разность температур на оси цилиндра, метод TempDiff\_Centr(): expected = " +

TempDiff\_Centr + "; actual=" + Math.Round(Form.TempDiff\_Centr(), 4));

// отобразить в журнале тестирования

Console.WriteLine("");

Console.WriteLine("--- Результаты расчета");

Console.WriteLine("Разность температур для массы цилиндра, метод TempDiff\_Mass(): expected = " +

TempDiff\_Mass + "; actual=" + Math.Round(Form.TempDiff\_Mass(), 4));

// отобразить в журнале тестирования

Console.WriteLine("");

Console.WriteLine("--- Результаты расчета");

Console.WriteLine("Разность температур на поверхности цилиндра, метод TempDiff\_Surface(): expected = " +

TempDiff\_Surface + "; actual=" + Math.Round(Form.TempDiff\_Surface(), 4));

WorkBook.Close(false, null, null);

objExcel.Quit();

}

catch

{

if (WorkBook != null) WorkBook.Close(false, null, null);

if (objExcel != null) objExcel.Quit();

}

finally

{

//WorkBook.Close(false, null, null);

//objExcel.Quit();

}

}

[Test]

public void CalculationTest2()

{

double \_r = 0.055;

double \_lamdaM = 42;

double \_cM = 712;

double \_roM = 7860;

double \_alfa = 525;

double \_t = 581;

string \_material = "Сталь";

double \_t\_pech = 1420;

double \_t\_beg = 20;

double \_t\_end = 1200;

Formules Form = new Formules(\_r, \_lamdaM, \_cM, \_roM, \_alfa, \_t, \_material, \_t\_end, \_t\_beg, \_t\_pech);

try

{

objExcel = new Microsoft.Office.Interop.Excel.Application();

WorkBook = objExcel.Workbooks.Open(

Directory.GetCurrentDirectory() + "\\" + fileName);

Microsoft.Office.Interop.Excel.Worksheet WorkSheet = (Microsoft.Office.Interop.Excel.Worksheet)WorkBook.Sheets["Обратная задача"];

((Microsoft.Office.Interop.Excel.Range)WorkSheet.Cells[2, 7]).Value2 = Form.r;

((Microsoft.Office.Interop.Excel.Range)WorkSheet.Cells[4, 7]).Value2 = Form.lamdaM;

((Microsoft.Office.Interop.Excel.Range)WorkSheet.Cells[5, 7]).Value2 = Form.cM;

((Microsoft.Office.Interop.Excel.Range)WorkSheet.Cells[6, 7]).Value2 = Form.roM;

((Microsoft.Office.Interop.Excel.Range)WorkSheet.Cells[7, 7]).Value2 = Form.alfa;

((Microsoft.Office.Interop.Excel.Range)WorkSheet.Cells[8, 7]).Value2 = Form.t\_p;

((Microsoft.Office.Interop.Excel.Range)WorkSheet.Cells[9, 7]).Value2 = Form.t\_beg;

((Microsoft.Office.Interop.Excel.Range)WorkSheet.Cells[10, 7]).Value2 = Form.t\_end;

// отобразить в журнале тестирования

Console.WriteLine("--- Исходные данные");

Console.WriteLine("Радиус цилиндра: {0}", Form.r.ToString());

Console.WriteLine("Коэффициент теплопроводности материала цилиндра: {0}", Form.lamdaM.ToString());

Console.WriteLine("Теплоемкость материала цилиндра: {0}", Form.cM.ToString());

Console.WriteLine("Плотность материала цилиндра: {0}", Form.roM.ToString());

Console.WriteLine("Коэффициент теплоотдачи: {0}", Form.alfa.ToString());

Console.WriteLine("Температура печи: {0}", Form.t\_p.ToString());

Console.WriteLine("Начальная температура цилиндра: {0}", Form.t\_beg.ToString());

Console.WriteLine("Конечная температура цилиндра: {0}", Form.t\_end.ToString());

double Temp\_DiffSurface = double.Parse(((Microsoft.Office.Interop.Excel.Range)WorkSheet.Cells[2, 17]).Value.ToString());

double Bi = double.Parse(((Microsoft.Office.Interop.Excel.Range)WorkSheet.Cells[3, 17]).Value.ToString());

double A = double.Parse(((Microsoft.Office.Interop.Excel.Range)WorkSheet.Cells[4, 17]).Value.ToString());

double Time\_heath = double.Parse(((Microsoft.Office.Interop.Excel.Range)WorkSheet.Cells[5, 17]).Value.ToString());

Assert.AreEqual(Temp\_DiffSurface, Math.Round(Form.Temp\_DiffSurface(), 4), 0.001);

System.Diagnostics.Debug.WriteLine("Разность температур на поверхности цилиндра: expected =" +

Temp\_DiffSurface + "; actual=" + Math.Round(Form.Temp\_DiffSurface(), 4));

Assert.AreEqual(Bi, Math.Round(Form.Bi(), 1), 0.001);

System.Diagnostics.Debug.WriteLine("Число теплового подобия - BI: expected =" +

Bi + "; actual=" + Math.Round(Form.Bi(), 1));

Assert.AreEqual(A, Math.Round(Form.A(), 7), 0.001);

System.Diagnostics.Debug.WriteLine("Коэффициент температуропроводности материала - А: expected =" +

A + "; actual=" + Math.Round(Form.A(), 7));

Assert.AreEqual(Time\_heath, Math.Round(Form.Time\_heath(), 1), 0.001);

System.Diagnostics.Debug.WriteLine("Время нагрева цилиндра - Time\_heath: expected =" +

Time\_heath + "; actual=" + Math.Round(Form.Time\_heath(), 1));

// отобразить в журнале тестирования

Console.WriteLine("");

Console.WriteLine("--- Результаты расчета");

Console.WriteLine("Разность температур на поверхности цилиндра, метод Temp\_DiffSurface(): expected = " +

Temp\_DiffSurface + "; actual=" + Math.Round(Form.Temp\_DiffSurface(), 4));

// отобразить в журнале тестирования

Console.WriteLine("");

Console.WriteLine("--- Результаты расчета");

Console.WriteLine("Число теплового подобия, метод Bi(): expected = " +

Bi + "; actual=" + Math.Round(Form.Bi(), 1));

// отобразить в журнале тестирования

Console.WriteLine("");

Console.WriteLine("--- Результаты расчета");

Console.WriteLine("Коэффициент температуропроводности материала, метод A(): expected = " +

A + "; actual=" + Math.Round(Form.A(), 7));

// отобразить в журнале тестирования

Console.WriteLine("");

Console.WriteLine("--- Результаты расчета");

Console.WriteLine("Время нагрева цилиндра, метод Time\_heath(): expected = " +

Time\_heath + "; actual=" + Math.Round(Form.Time\_heath(), 1));

WorkBook.Close(false, null, null);

objExcel.Quit();

}

catch

{

if (WorkBook != null) WorkBook.Close(false, null, null);

if (objExcel != null) objExcel.Quit();

}

finally

{

//WorkBook.Close(false, null, null);

//objExcel.Quit();

}

}

}

}

# **ПРИЛОЖЕНИЕ Д ФРАГМЕНТ ЛИСТИНГА ПРОГРАММНОГО КОДА КЛАССА FORMULES.CS**

namespace MathLib\_Cilindr

{

public class Formules

{

//Экземпляр класса

public Formules(double \_r, double \_lamdaM, double \_cM, double \_roM, double \_alfa, double \_t, string \_material, double \_tp, double \_tend, double \_tbegin)

{

r = \_r;

lamdaM = \_lamdaM;

cM = \_cM;

roM = \_roM;

alfa = \_alfa;

t = \_t;

material = \_material;

t\_p = \_tp;

t\_end = \_tend;

t\_beg = \_tbegin;

}

#region ИСХОДНЫЕ ДАННЫЕ (ПРЯМАЯ ЗАДАЧА)

//ИСХОДНЫЕ ДАННЫЕ (РАСЧЕТ ТЕМПЕРАТУР НАГРЕВА ЦИЛИНДРА)

private double \_r;

/// <summary>

/// РАДИУС ЦИЛИНДРА, м

/// </summary>

public double r

{

get { return \_r; }

set { \_r = value; }

}

private double \_lamdaM;

/// <summary>

/// КОЭФФИЦИЕНТ ТЕПЛОПРОВОДНОСТИ МАТЕРИАЛА, Вт/(м\*К)

/// </summary>

public double lamdaM

{

get { return \_lamdaM; }

set { \_lamdaM = value; }

}

private double \_cM;

/// <summary>

/// КОЭФФИЦИЕНТ ТЕПЛОЁМКОСТИ МАТЕРИАЛА, Дж/(кг\*К)

/// </summary>

public double cM

{

get { return \_cM; }

set { \_cM = value; }

}

private double \_roM;

/// <summary>

/// ПЛОТНОСТЬ МАТЕРИАЛА, кг/м3

/// </summary>

public double roM

{

get { return \_roM; }

set { \_roM = value; }

}

private double \_alfa;

/// <summary>

/// КОЭФФИЦИЕНТ ТЕПЛООТДАЧИ, Вт/(м2\*К)

/// </summary>

public double alfa

{

get { return \_alfa; }

set { \_alfa = value; }

}

private double \_t;

/// <summary>

/// ВРЕМЯ НАГРЕВА, с

/// </summary>

public double t

{

get { return \_t; }

set { \_t = value; }

}

private string \_material;

/// <summary>

/// МАТЕРИАЛ ЦИЛИНДРА

/// </summary>

public string material

{

get { return \_material; }

set { \_material = value; }

}

//ПОСТОЯННЫЕ ЗНАЧЕНИЯ

private double \_Rc;

/// <summary>

/// ФУНКЦИЯ ЧИСЛА БИО Рц

/// </summary>

public double Rc

{

get { return \_Rc; }

set { \_Rc = value; }

}

private double \_Mc;

/// <summary>

/// ФУНКЦИЯ ЧИСЛА БИО Мц

/// </summary>

public double Mc

{

get { return \_Mc; }

set { \_Mc = value; }

}

private double \_Nc;

/// <summary>

/// ФУНКЦИЯ ЧИСЛА БИО Nц

/// </summary>

public double Nc

{

get { return \_Nc; }

set { \_Nc = value; }

}

private double \_mu;

/// <summary>

/// ФУНКЦИЯ ЧИСЛА БИО МЮц

/// </summary>

public double mu

{

get { return \_mu; }

set { \_mu = value; }

}

#endregion ИСХОДНЫЕ ДАННЫЕ (ПРЯМАЯ ЗАДАЧА)

#region ИСХОДНЫЕ ДАННЫЕ (ОБРАТНАЯ ЗАДАЧА)

//ИСХОДНЫЕ ДАННЫЕ(РАСЧЁТ ВРЕМЕНИ НАГРЕВА ЦИЛИНДРА)

private double \_tp;

///<summary>

///Температура печи, град.С

///</summary>

public double t\_p

{

get { return \_tp; }

set { \_tp = value; }

}

private double \_t\_end;

///<summary>

///Конечная температура нагрева цилиндра, град.С

///</summary>

public double t\_end

{

get { return \_t\_end; }

set { \_t\_end = value; }

}

private double \_t\_beg;

///<summary>

///Начальная температура цилиндра, град.С

///</summary>

public double t\_beg

{

get { return \_t\_beg; }

set { \_t\_beg = value; }

}

#endregion ИСХОДНЫЕ ДАННЫЕ (ОБРАТНАЯ ЗАДАЧА)

#region ВЫБОР ПАРАМЕТРОВ ЧИСЛА БИО

public void Search()

{

switch (\_Bi)

{

case 0.00:

mu = 0.000;

Rc = 1.000;

Mc = 1.000;

Nc = 1.000;

break;

case 0.01:

mu = 0.020;

Rc = 0.997;

Mc = 1.000;

Nc = 1.002;

break;

case 0.10:

mu = 0.195;

Rc = 0.975;

Mc = 1.000;

Nc = 1.025;

break;

case 0.20:

mu = 0.381;

Rc = 0.951;

Mc = 0.999;

Nc = 1.048;

break;

case 0.30:

mu = 0.557;

Rc = 0.927;

Mc = 0.998;

Nc = 1.071;

break;

case 0.40:

mu = 0.725;

Rc = 0.904;

Mc = 0.997;

Nc = 1.093;

break;

case 0.50:

mu = 0.885;

Rc = 0.881;

Mc = 0.995;

Nc = 1.114;

break;

case 0.60:

mu = 1.037;

Rc = 0.859;

Mc = 0.993;

Nc = 1.134;

break;

case 0.70:

mu = 1.182;

Rc = 0.837;

Mc = 0.992;

Nc = 1.154;

break;

case 0.80:

mu = 1.320;

Rc = 0.816;

Mc = 0.989;

Nc = 1.172;

break;

case 0.90:

mu = 1.452;

Rc = 0.796;

Mc = 0.987;

Nc = 1.190;

break;

case 1.00:

mu = 1.577;

Rc = 0.776;

Mc = 0.984;

Nc = 1.207;

break;

case 1.10:

mu = 1.577;

Rc = 0.776;

Mc = 0.984;

Nc = 1.207;

break;

case 1.20:

mu = 1.577;

Rc = 0.776;

Mc = 0.984;

Nc = 1.207;

break;

case 1.30:

mu = 1.577;

Rc = 0.776;

Mc = 0.984;

Nc = 1.207;

break;

case 1.40:

mu = 2.123;

Rc = 0.686;

Mc = 0.970;

Nc = 1.281;

break;

case 1.50:

mu = 2.123;

Rc = 0.686;

Mc = 0.970;

Nc = 1.281;

break;

case 1.60:

mu = 2.123;

Rc = 0.686;

Mc = 0.970;

Nc = 1.281;

break;

case 1.70:

mu = 2.123;

Rc = 0.686;

Mc = 0.970;

Nc = 1.281;

break;

case 1.80:

mu = 2.558;

Rc = 0.610;

Mc = 0.953;

Nc = 1.338;

break;

case 1.90:

mu = 2.558;

Rc = 0.610;

Mc = 0.953;

Nc = 1.338;

break;

case 2.00:

mu = 2.558;

Rc = 0.610;

Mc = 0.953;

Nc = 1.338;

break;

case 2.10:

mu = 2.558;

Rc = 0.610;

Mc = 0.953;

Nc = 1.338;

break;

case 2.20:

mu = 2.558;

Rc = 0.610;

Mc = 0.953;

Nc = 1.338;

break;

case 2.30:

mu = 2.558;

Rc = 0.610;

Mc = 0.953;

Nc = 1.338;

break;

case 2.40:

mu = 2.558;

Rc = 0.610;

Mc = 0.953;

Nc = 1.338;

break;

case 2.50:

mu = 2.558;

Rc = 0.610;

Mc = 0.953;

Nc = 1.338;

break;

case 2.60:

mu = 3.199;

Rc = 0.492;

Mc = 0.923;

Nc = 1.419;

break;

case 2.70:

mu = 3.199;

Rc = 0.492;

Mc = 0.923;

Nc = 1.419;

break;

case 2.80:

mu = 3.199;

Rc = 0.492;

Mc = 0.923;

Nc = 1.419;

break;

case 2.90:

mu = 3.199;

Rc = 0.492;

Mc = 0.923;

Nc = 1.419;

break;

case 3.00:

mu = 3.199;

Rc = 0.492;

Mc = 0.923;

Nc = 1.419;

break;

case 3.10:

mu = 3.199;

Rc = 0.492;

Mc = 0.923;

Nc = 1.419;

break;

case 3.20:

mu = 3.199;

Rc = 0.492;

Mc = 0.923;

Nc = 1.419;

break;

case 3.30:

mu = 3.199;

Rc = 0.492;

Mc = 0.923;

Nc = 1.419;

break;

case 3.40:

mu = 3.199;

Rc = 0.492;

Mc = 0.923;

Nc = 1.419;

break;

case 3.50:

mu = 3.199;

Rc = 0.492;

Mc = 0.923;

Nc = 1.419;

break;

case 3.60:

mu = 3.641;

Rc = 0.407;

Mc = 0.895;

Nc = 1.470;

break;

case 3.70:

mu = 3.641;

Rc = 0.407;

Mc = 0.895;

Nc = 1.470;

break;

case 3.80:

mu = 3.641;

Rc = 0.407;

Mc = 0.895;

Nc = 1.470;

break;

case 3.90:

mu = 3.641;

Rc = 0.407;

Mc = 0.895;

Nc = 1.470;

break;

case 4.00:

mu = 3.641;

Rc = 0.407;

Mc = 0.895;

Nc = 1.470;

break;

case 4.10:

mu = 3.641;

Rc = 0.407;

Mc = 0.895;

Nc = 1.470;

break;

case 4.20:

mu = 3.641;

Rc = 0.407;

Mc = 0.895;

Nc = 1.470;

break;

case 4.30:

mu = 3.641;

Rc = 0.407;

Mc = 0.895;

Nc = 1.470;

break;

case 4.40:

mu = 3.641;

Rc = 0.407;

Mc = 0.895;

Nc = 1.470;

break;

case 4.50:

mu = 3.641;

Rc = 0.407;

Mc = 0.895;

Nc = 1.470;

break;

case 4.60:

mu = 3.959;

Rc = 0.345;

Mc = 0.872;

Nc = 1.503;

break;

case 4.70:

mu = 3.959;

Rc = 0.345;

Mc = 0.872;

Nc = 1.503;

break;

case 4.80:

mu = 3.959;

Rc = 0.345;

Mc = 0.872;

Nc = 1.503;

break;

case 4.90:

mu = 3.959;

Rc = 0.345;

Mc = 0.872;

Nc = 1.503;

break;

case 5.00:

mu = 3.959;

Rc = 0.345;

Mc = 0.872;

Nc = 1.503;

break;

case 5.10:

mu = 3.959;

Rc = 0.345;

Mc = 0.872;

Nc = 1.503;

break;

case 5.20:

case 5.30:

case 5.40:

case 5.50:

case 5.60:

case 5.70:

case 5.80:

case 5.90:

case 6.00:

case 6.10:

case 6.20:

case 6.30:

case 6.40:

case 6.50:

case 6.60:

case 6.70:

case 6.80:

case 6.90:

case 7.00:

case 7.10:

case 7.20:

case 7.30:

case 7.40:

case 7.50:

case 7.60:

case 7.70:

case 7.80:

case 7.90:

mu = 3.959;

Rc = 0.345;

Mc = 0.872;

Nc = 1.503;

break;

case 8.00:

case 8.10:

case 8.20:

case 8.30:

case 8.40:

case 8.50:

case 8.60:

case 8.70:

case 8.80:

case 8.90:

case 9.00:

case 9.10:

case 9.20:

case 9.30:

case 9.40:

case 9.50:

case 9.60:

case 9.70:

case 9.80:

case 9.90:

case 10.00:

case 10.10:

case 10.20:

case 10.30:

case 10.40:

case 10.50:

case 10.60:

case 10.70:

case 10.80:

case 10.90:

case 11.00:

case 11.10:

case 11.20:

case 11.30:

case 11.40:

case 11.50:

case 11.60:

case 11.70:

case 11.80:

case 11.90:

mu = 4.750;

Rc = 0.191;

Mc = 0.804;

Nc = 1.568;

break;

case 12.00:

case 12.10:

case 12.20:

case 12.30:

case 12.40:

case 12.50:

case 12.60:

case 12.70:

case 12.80:

case 12.90:

case 13.00:

case 13.10:

case 13.20:

case 13.30:

case 13.40:

case 13.50:

case 13.60:

case 13.70:

case 13.80:

case 13.90:

case 14.00:

case 14.10:

case 14.20:

case 14.30:

case 14.40:

case 14.50:

case 14.60:

case 14.70:

case 14.80:

case 14.90:

case 15.00:

mu = 5.066;

Rc = 0.130;

Mc = 0.772;

Nc = 1.585;

break;

case 20.00:

mu = 5.235;

Rc = 0.099;

Mc = 0.754;

Nc = 1.592;

break;

case 30.00:

mu = 5.411;

Rc = 0.066;

Mc = 0.736;

Nc = 1.596;

break;

case 40.00:

mu = 5.501;

Rc = 0.050;

Mc = 0.724;

Nc = 1.599;

break;

case 50.00:

mu = 5.557;

Rc = 0.040;

Mc = 0.718;

Nc = 1.600;

break;

}

}

#endregion

#region ПРОМЕЖУТОЧНЫЙ РАСЧЕТ

//ПРОМЕЖУТОЧНЫЙ РАСЧЕТ

private double \_Bi;

/// <summary>

/// ЧИСЛО ТЕПЛОВОГО ПОДОБИЯ БИО

/// </summary>

public double Bi()

{

\_Bi = Math.Round(((\_alfa \* \_r) / \_lamdaM), 1);

return \_Bi;

}

private double \_A;

/// <summary>

/// КОЭФФИЦИЕНТ ТЕМПЕРАТУРОПРОВОДНОСТИ МАТЕРИАЛА, м2/с

/// </summary>

public double A()

{

\_A = \_lamdaM / (\_cM \* \_roM);

return \_A;

}

private double \_Fo;

/// <summary>

/// ЧИСЛО ПОДОБИЯ ФУРЬЕ

/// </summary>

public double Fo()

{

\_Fo = (\_A \* \_t) / Math.Pow(\_r, 2);

return \_Fo;

}

#endregion ПРОМЕЖУТОЧНЫЙ РАСЧЕТ

#region РАСЧЕТ НАГРЕВА ЦИЛИНДРА

//РАСЧЕТ НАГРЕВА ЦИЛИНДРА

private double \_TDS;

/// <summary>

/// ОТНОСТИТЕЛЬНАЯ РАЗНОСТЬ ТЕМПЕРАТУР НА ПОВЕРХНОСТИ ЦИЛИНДРА

/// </summary>

public double TempDiff\_Surface()

{

Search();

\_TDS = \_Rc \* Math.Exp(-\_mu \* \_Fo);

return \_TDS;

}

private double \_TDM;

/// <summary>

/// ОТНОСИТЕЛЬНАЯ РАЗНОСТЬ ТЕПЕРАТУР ДЛЯ МАССЫ ЦИЛИНДРА

/// </summary>

public double TempDiff\_Mass()

{

Search();

\_TDM = \_Mc \* Math.Exp(-\_mu \* \_Fo);

return \_TDM;

}

private double \_TDC;

/// <summary>

/// ОТНОСИТЕЛЬНАЯ РАЗНОСТЬ ТЕМПЕРАТУР НА ОСИ ЦИЛИНДРА

/// </summary>

public double TempDiff\_Centr()

{

Search();

\_TDC = \_Nc \* Math.Exp(-\_mu \* \_Fo);

return \_TDC;

}

#endregion РАСЧЕТ НАГРЕВА ЦИЛИНДРА

#region РАСЧЕТ ВРЕМЕНИ НАГРЕА ЦИЛИНДРА

//РАСЧЕТ ВРЕМЕНИ НАГРЕВА

private double \_T\_DS;

///<summary>

///РАЗНОСТЬ ТЕМПЕРАТУР НА ПОВЕРХНОСТИ ЦИЛИНДРА

/// </summary>

public double Temp\_DiffSurface()

{

\_T\_DS = (\_tp - \_t\_end) / (\_tp - \_t\_beg);

return \_T\_DS;

}

private double \_TH;

///<summary>

///ВРЕМЯ НАГРЕВА ЦИЛИНДРА, с

/// </summary>

public double Time\_heath()

{

Search();

\_TH = (Math.Pow(\_r, 2) / (\_A \* \_mu)) \* Math.Log(\_Rc / \_T\_DS);

return \_TH;

}

#endregion РАСЧЕТ ВРЕМЕНИ НАГРЕВА ЦИЛИНДРА

}

}