|  |
| --- |
| **Министерство образования и науки Российской Федерации**  Федеральное государственное автономное образовательное  учреждение высшего образования  **«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**  **ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»** |

Подразделение: Инженерная школа энергетики

Направление подготовки: 09.04.03 – Прикладная информатика

Отделение: Электроэнергетики и электротехники

**Проектная документация**

**Отчёт по лабораторной работе №5**

по дисциплине: «Основы объектно-ориентированного программирования»

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Выполнил студент гр. О-5КМ01 | |  | \_\_\_\_\_\_\_\_\_\_ | |  | Наумкин А.В. |
|  | |  |  | |  |  |
|  | |  |  |  | | |
|  | |  |  | |  |  |
| Отчёт принял | доцент, к.т.н. |  | \_\_\_\_\_\_\_\_\_\_ | |  | Калентьев А. А. |
|  |  |  |  | |  |  |
|  |  |  |  | \_\_\_ \_\_\_\_\_\_ | | |

Томск 2022

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc117436543)

[1 Основная часть 4](#_Toc117436544)

[1.1 UML диаграмма вариантов использования 4](#_Toc117436545)

[1.2 UML диаграмма классов 5](#_Toc117436546)

[1.3 Описание классов, образующих связь типа «общее-частное» 7](#_Toc117436547)

[1.4 Дерево ветвлений Git 8](#_Toc117436548)

[1.5 Тестирование программы 8](#_Toc117436549)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 16](#_Toc117436550)

**ВВЕДЕНИЕ**

Корректная и полная документация сопровождает разработку программного обеспечения (далее – ПО) от появления идеи до выпуска конечного продукта. Написание документации является обязательным критерием разработки и последующей поддержки проекта [1].

Целью данной лабораторной работы является разработка проектной документации на созданный программный продукт.

Для достижения поставленной цели должны быть выполнены следующие задачи:

* Составление технического задания (далее – ТЗ) на разработанную программу (Приложение А);
* Составление UML диаграммы вариантов использования для разработанной программы;
* Составление UML диаграммы классов;
* Описание классов, образующих связь типа «общее-частное»;
* Привести дерево ветвлений Git;
* Провести тестирование программы.

**1 Основная часть**

**1.1 UML диаграмма вариантов использования**

Вариант использования (use case) — это описание множества последовательных действий (включая вариации), которые выполняются некоторым субъектом с целью получения результата, значимого для некоторого действующего лица [1]. ВИ предполагает взаимодействие действующих лиц и системы или другого объекта. Действующее лицо представляет собой логически связанное множество ролей, которые играют пользователи системы во время взаимодействия с ней.

Диаграмма вариантов использования для разработанного ПО приведена на рисунке 1.

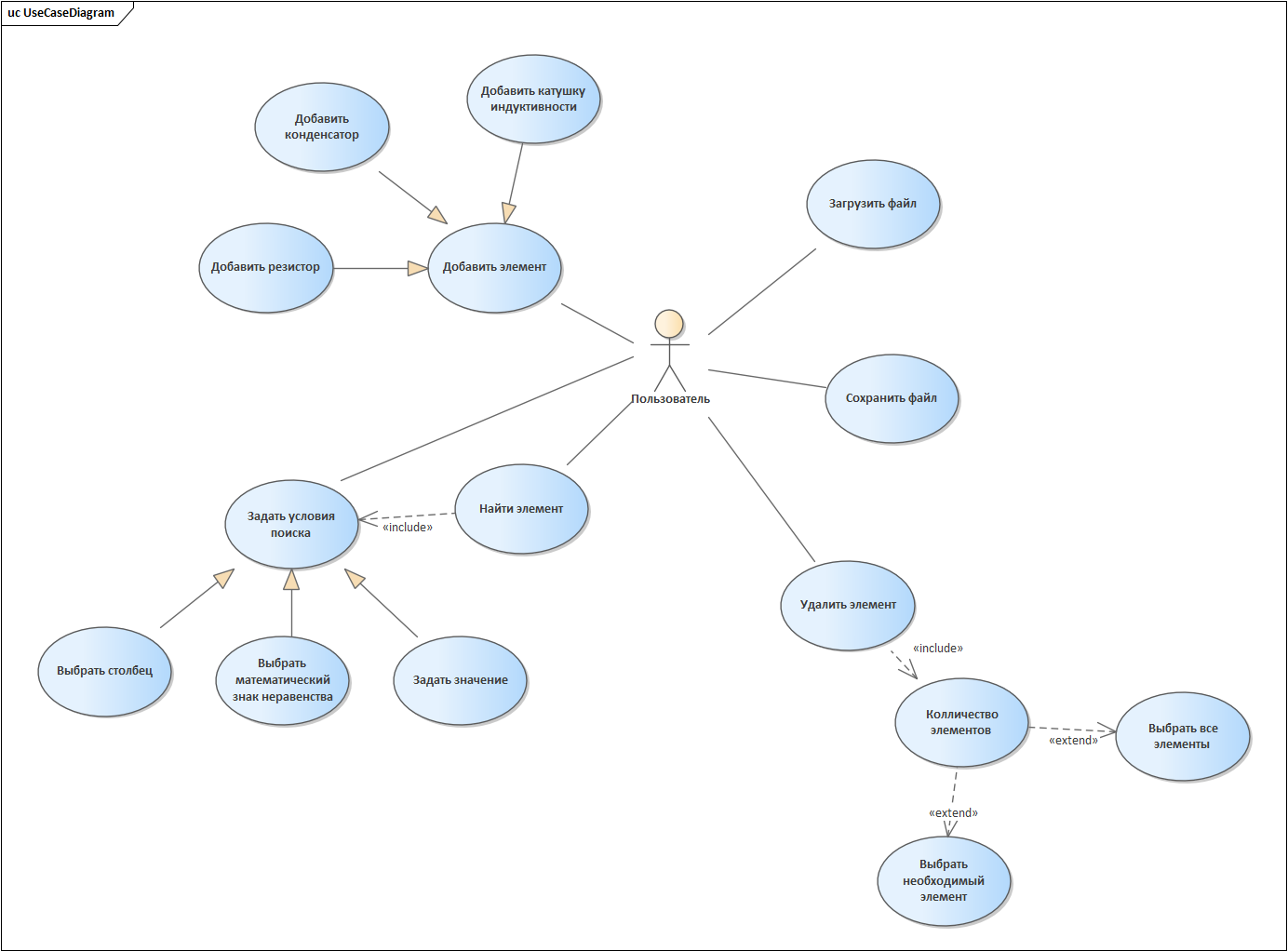


Рисунок 1 – Диаграмма вариантов использования

**1.2 UML диаграмма классов**

Диаграмма классов — это центральная методика моделирования, которая используется практически во всех объектно-ориентированных методах. Эта диаграмма описывает типы объектов в системе и различные виды статических отношений, которые существуют между ними. Широко применяется не только для документирования и визуализации, но также для конструирования посредством прямого или обратного проектирования.

Диаграмма классов приведена на рисунке 2.

«use»

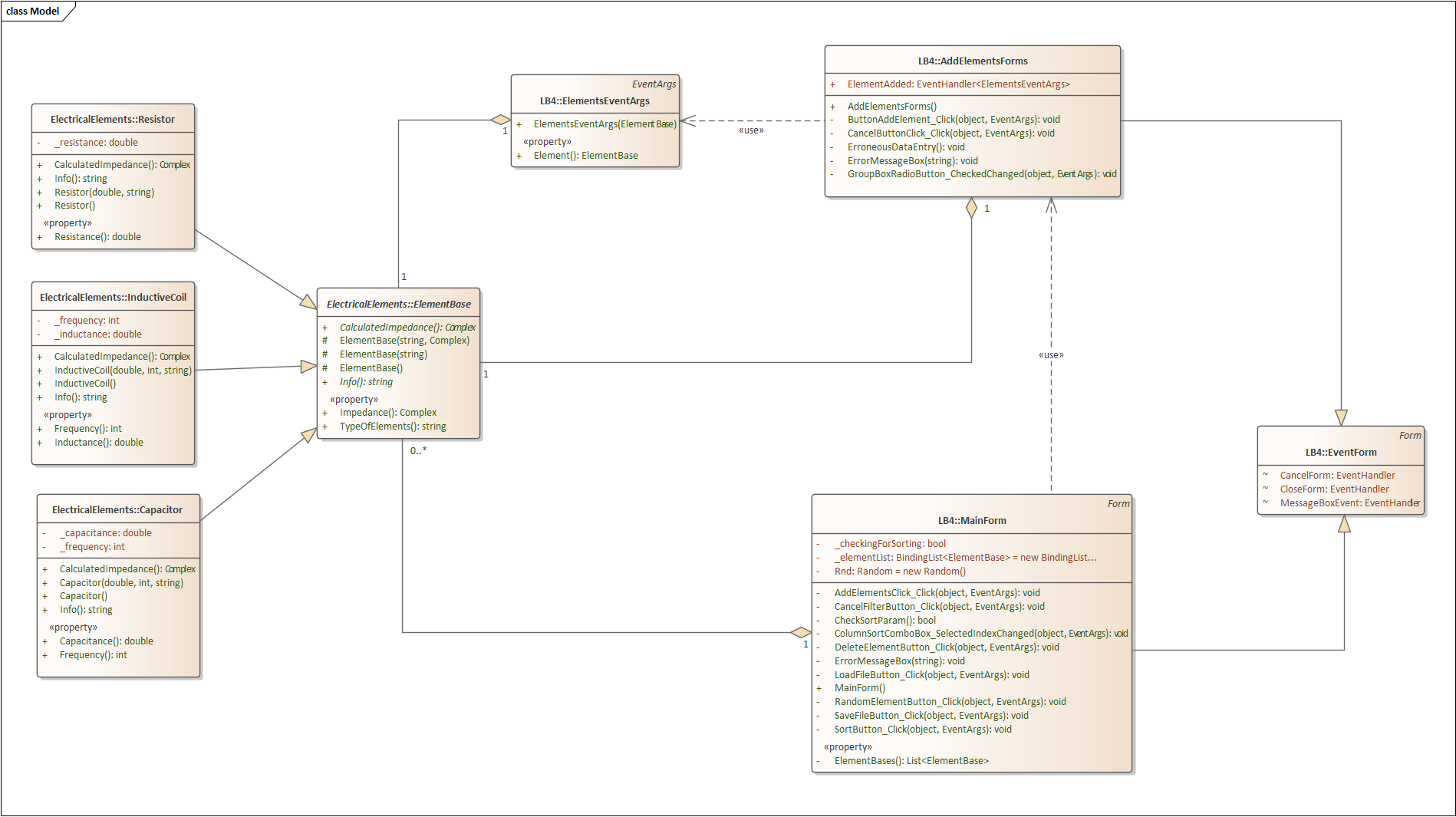


Рисунок 2 – UML диаграмма классов

**1.3 Описание классов, образующих связь типа «общее-частное»**

В таблице 1 приведено описание абстрактного класса *ElementBase* с его полями, свойствами и методами.

Таблица 1 – Описание класса *ElementBase*

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Описание класса | | |
| Класс *ElementBase* – абстрактный базовый класс для элементов | | |
| Свойства | | |
| Impedance (): | Complex | Импеданс элемента |
| TypeOfElements () | string | Тип элемента |
| Методы | | |
| + CalculatedImpedance() | Complex | Расчет имепеданса |

В таблицах 2–3 приведены описания классов 2 *Capacitor*,  
 3 *InductiveCoil*, 4 *Resistor,* которые наследуются от *ElementBase*.

Таблица 2 – Описание класса *InductiveСoil*

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Описание класса | | |
| Класс *InductiveСoil* – Класс катушки индуктивности | | |
| Поля | | |
| –\_frequency | int | Частота электрического тока |
| – \_inductance | double | Индуктивность |
| Свойства | | |
| + Frequency | int | Частота электрического тока |
| + Inductance | double | Индуктивность |
| Методы | | |
| + CalculatedImpedance() | Complex | Расчет имепеданса |

Таблица 3 – Описание класса *Capacitor*

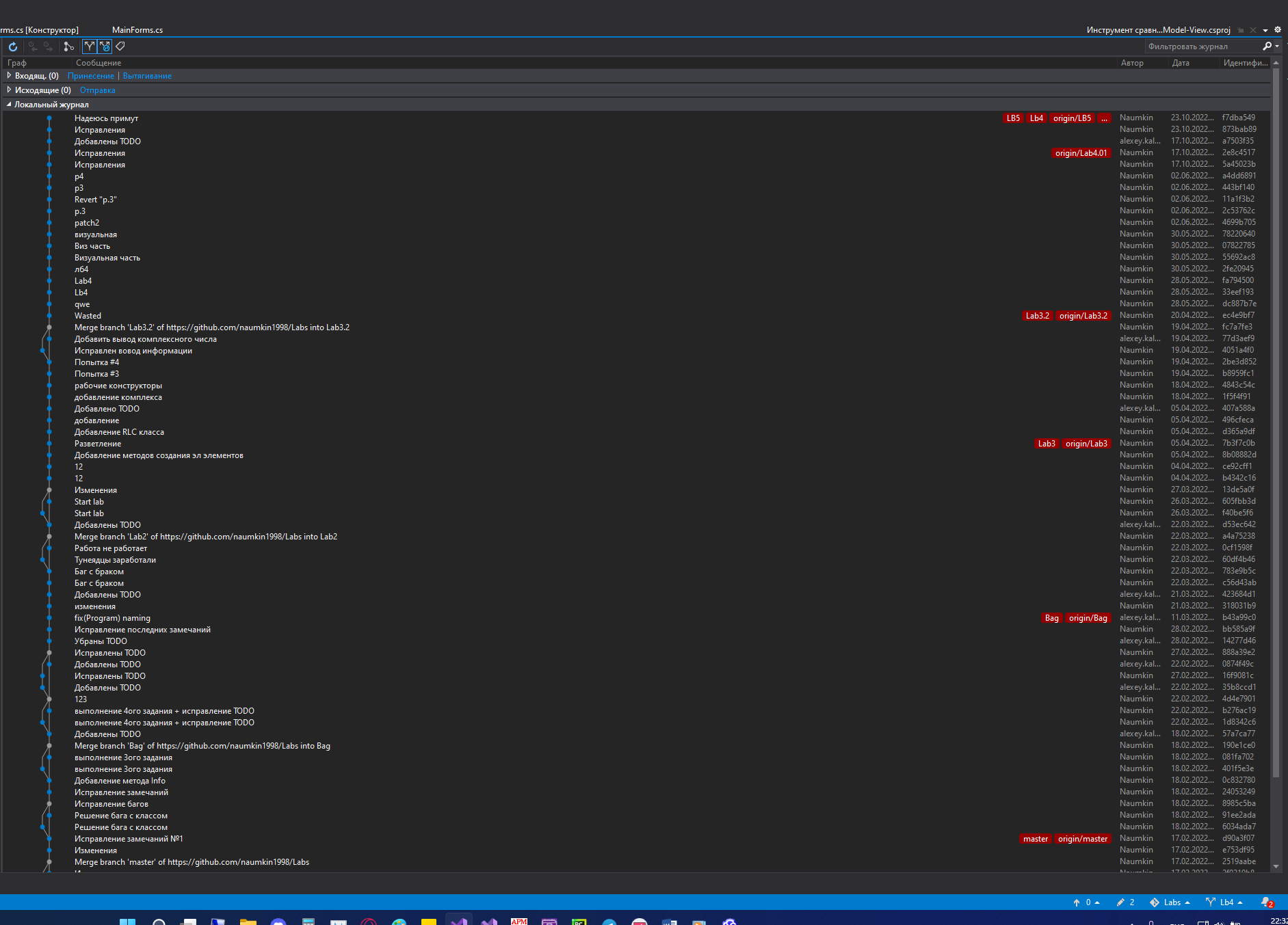
|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Описание класса | | |
| Класс *Capacitor* – Класс конденсатора | | |
| Поля | | |
| –\_frequency | int | Частота электрического тока |
| – \_capacitance | double | Емкость |
| Свойства | | |
| + Frequency | int | Частота электрического тока |
| + Capacitance | double | Емкость |
| Методы | | |
| + CalculatedImpedance() | Complex | Расчет имепеданса |

Таблица 4 –Описание класса *Resistor*

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Описание класса | | |
| Класс *Resistor* – Класс резистора | | |
| Поля | | |
| – \_resistance | double | Сопротивление |
| Свойства | | |
| + Resistance | double | Частота электрического тока |
| Методы | | |
| + CalculatedImpedance() | Complex | Расчет имепеданса |

**1.4 Дерево ветвлений Git**

На рисунке 3 представлено дерево ветвлений Git, полученное по окончании работы с проектом.

 Рисунок 3 – Дерево ветвлений Git

**1.5 Тестирование программы**

Далее приводится процесс функционального тестирования программы.

Графический интерфейс пользователя представлен на рисунке 4.

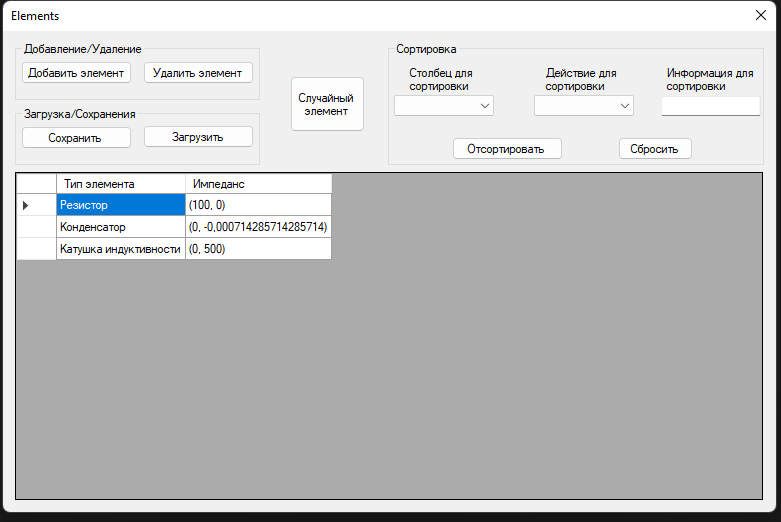


Рисунок 4 – Графический интерфейс пользователя

**1.5.1 Тестовый случай «**Добавить элемент**»**

Для добавления элемента необходимо вызвать соответствующую форму путём нажатия кнопки «Добавить элемент» (рисунок 5).

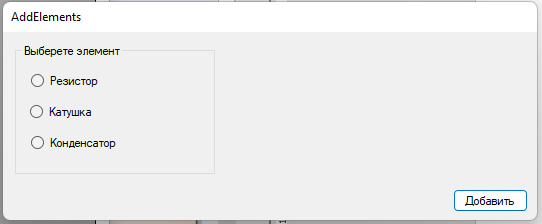


Рисунок 5 – Форма для добавления элемента

Параметры элемента (тип элемента и параметры) можно указать форме.

После ввода данных необходимо нажать кнопку «Добавить», элемент появится в таблице главной формы (рисунки 6 и 7).

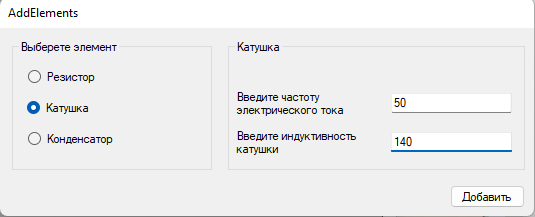


Рисунок 6 – Заполнение полей

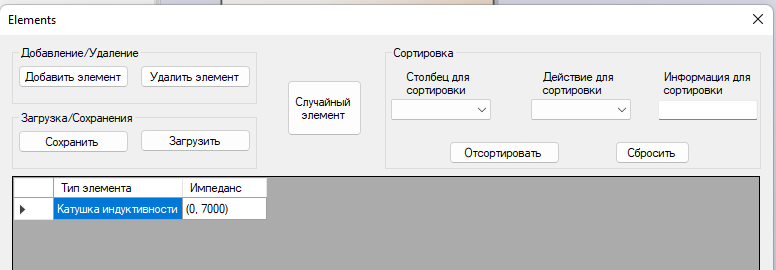


Рисунок 7 – Успешное добавление нового элемента

В программе предусмотрена система обработки некорректного ввода данных пользователем. Например, при попытке ввести число вне заранее определенного диапазона, появится соответствующее сообщение об ошибке (рисунки 8-9). Аналогичным образом обрабатывается попытка ввода NaN.

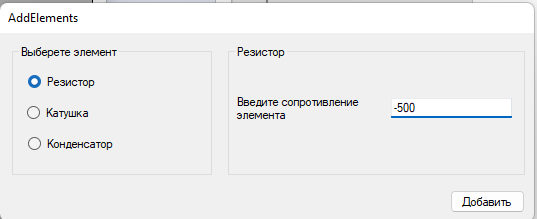


Рисунок 8 – Некорректный ввод (отрицательное число)

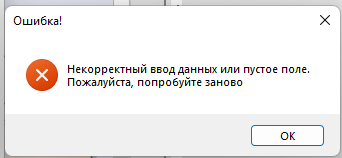
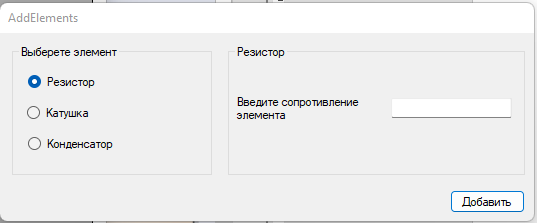


Рисунок 9 – Сообщение об ошибке



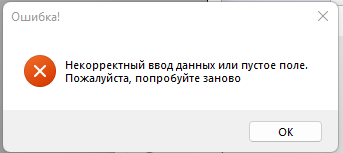


Рисунок 10 – Обработка ошибки NaN

Подобная обработка предусмотрена для всех элементов.

**1.5.2 Тестовый случай «Удалить элемент»**

Для удаления одного или нескольких элементов необходимо выбрать их в таблице и нажать на кнопку «Удалить элемент» (рисунки 11 и 12).

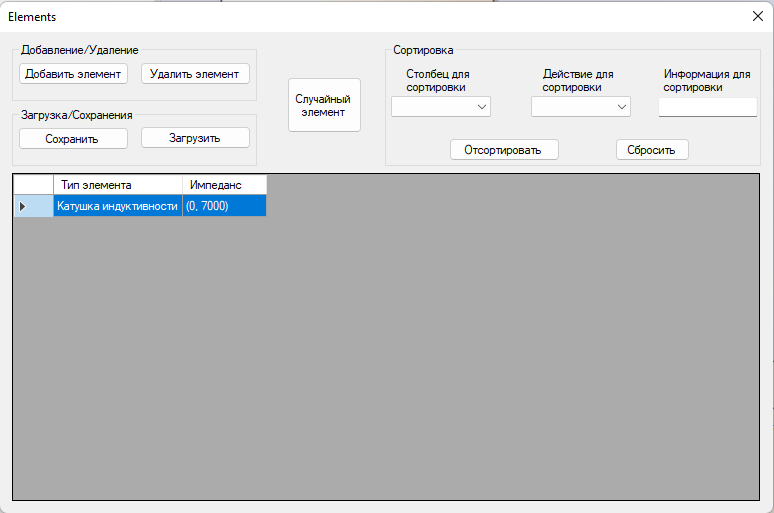


Рисунок 11 – Выбор элемента в таблице

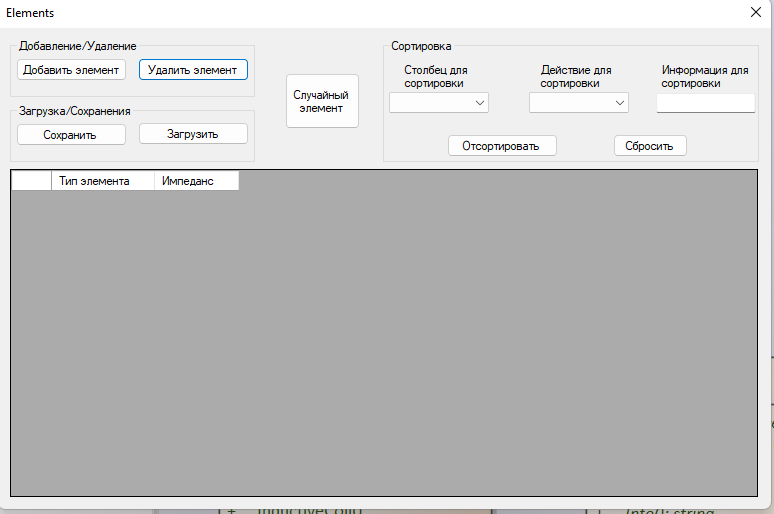


Рисунок 12 – Результат нажатия кнопки «Удалить элемент»

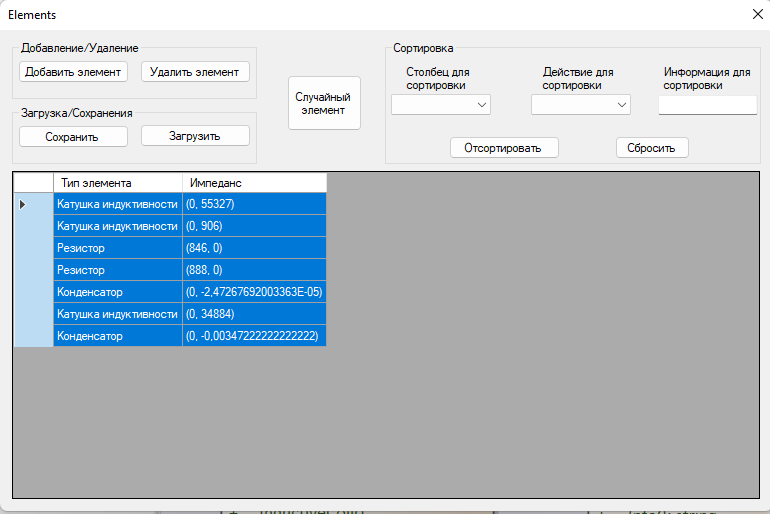


Рисунок 13 – Выбор нескольких элементов для удаления

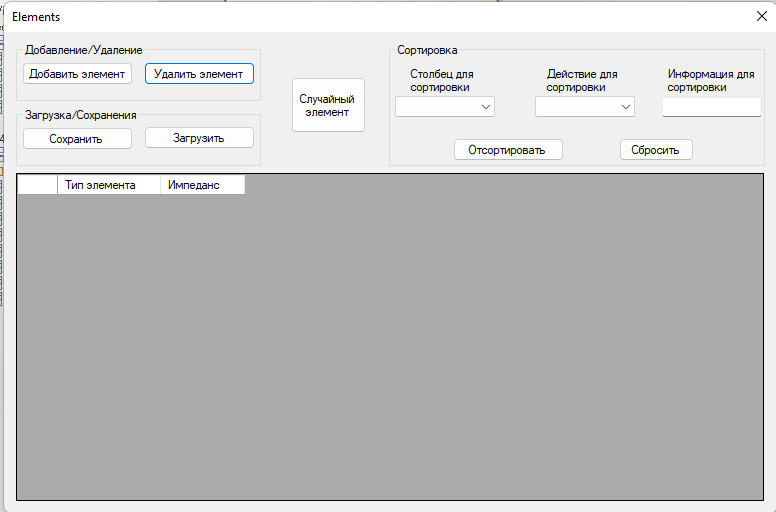


Рисунок 14 – Результат удаления выбранных элементов

**1.5.3 Тестовый случай «Search»**

Для поиска элементов предусмотрена панель параметров поиска (рисунок 15)

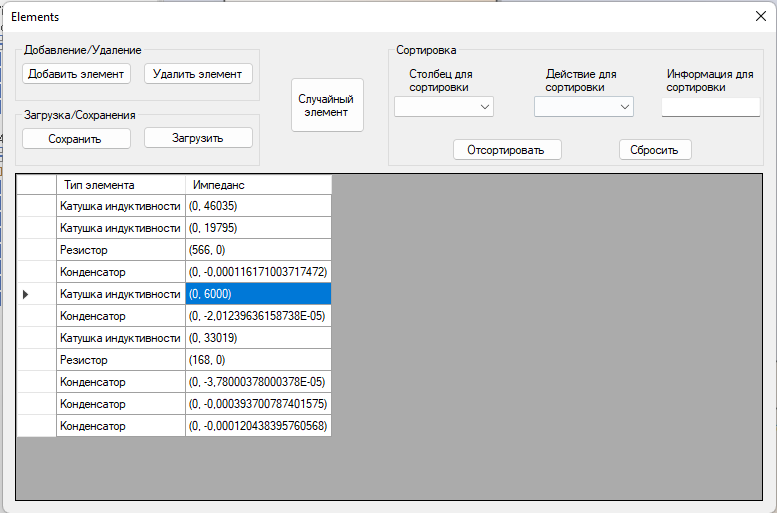


Рисунок 15 – Настройка параметров для поиска элементов

Пользователь выбирает параметры, по которым требуется найти элемент, вводит значения этих параметров и нажимает кнопку «Отсортировать» (рисунки 16 и 17).

Для того, чтобы сбросить фильтр, предусмотрена соответствующая кнопка «Сбросить» (рисунок 18).

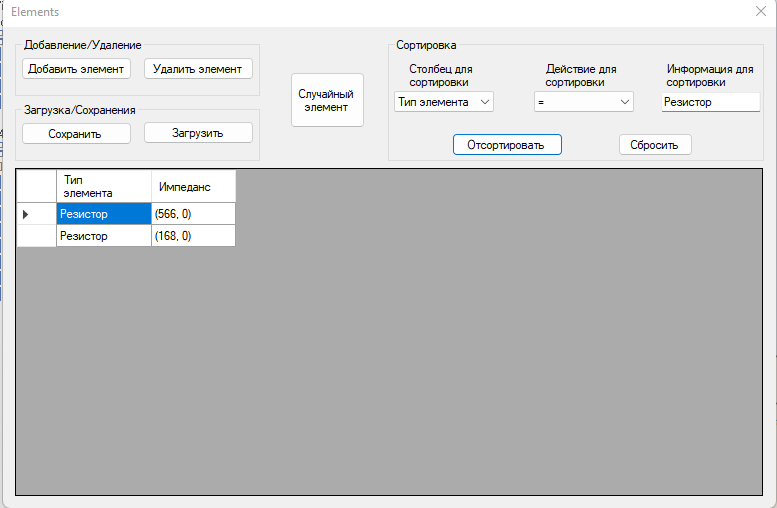


Рисунок 16 – Поиск элемента по типу

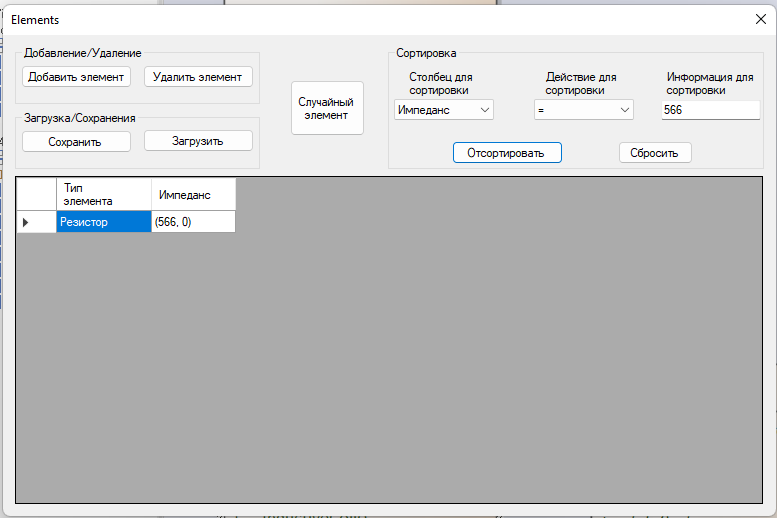


Рисунок 17 – Поиск по импедансу элемента

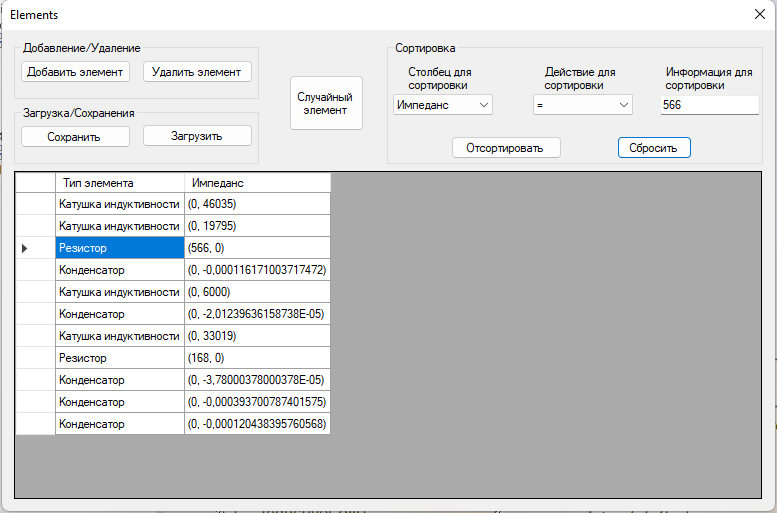


Рисунок 18 – Сброс условий поиска

**1.5.4 Тестовый случай «Сохранить»**

Для сохранения данных в таблице необходимо нажать на кнопку «Сохранить». Откроется системный диалог сохранения файла, где пользователь выбирает директорию и указывает имя файла (рисунок 19).

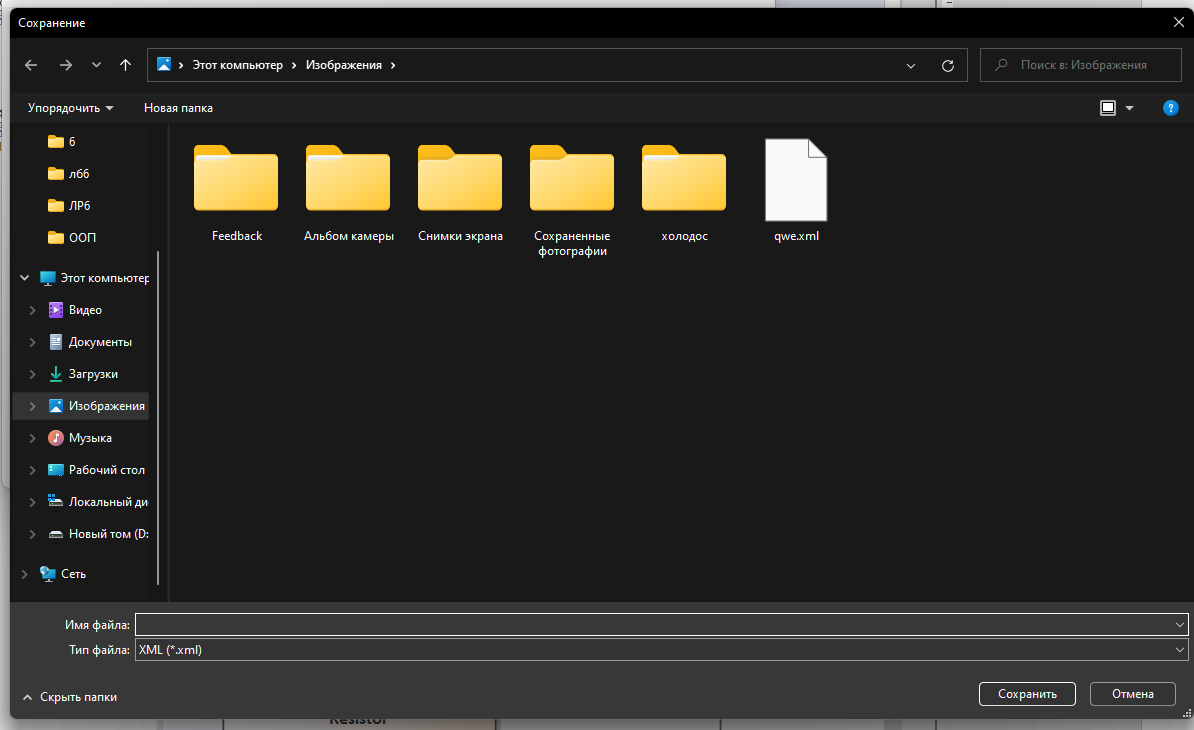


Рисунок 19 – Сохранение файла

**1.5.5 Тестовый случай «Загрузить данные»**

Для загрузки данных в таблицу необходимо нажать на соответствующую кнопку «Load».

Далее откроется системный диалог загрузки файла (рисунок 20).

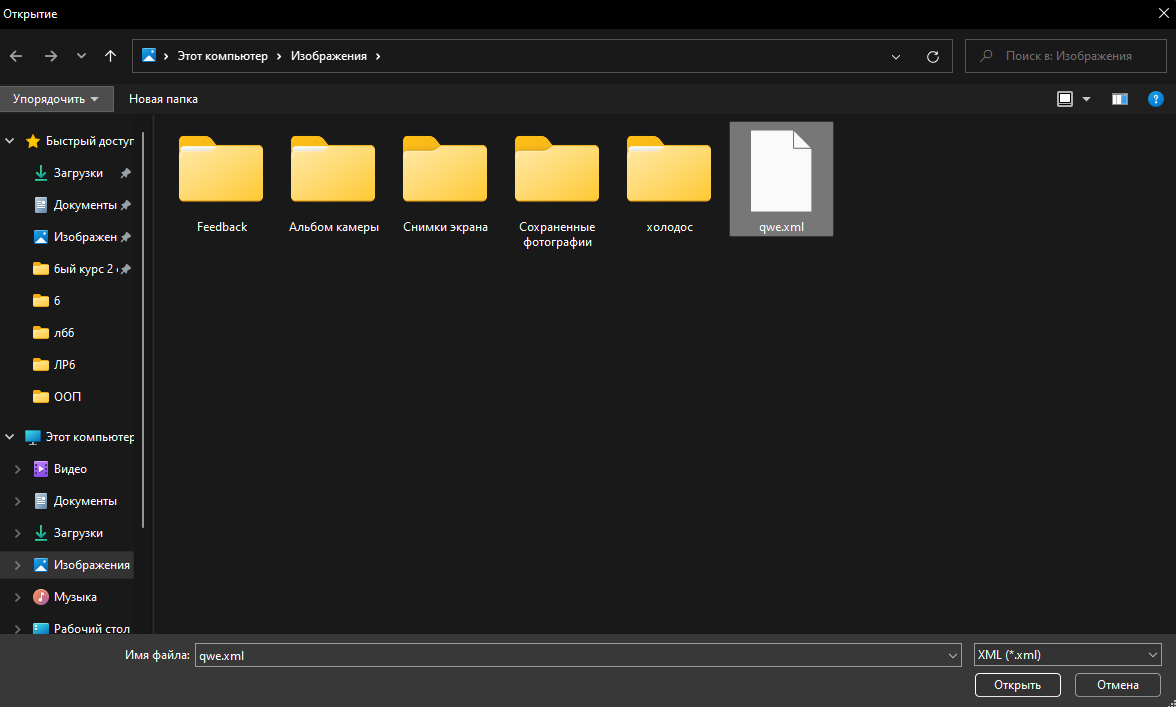


Рисунок 20 – Выбор файла для загрузки

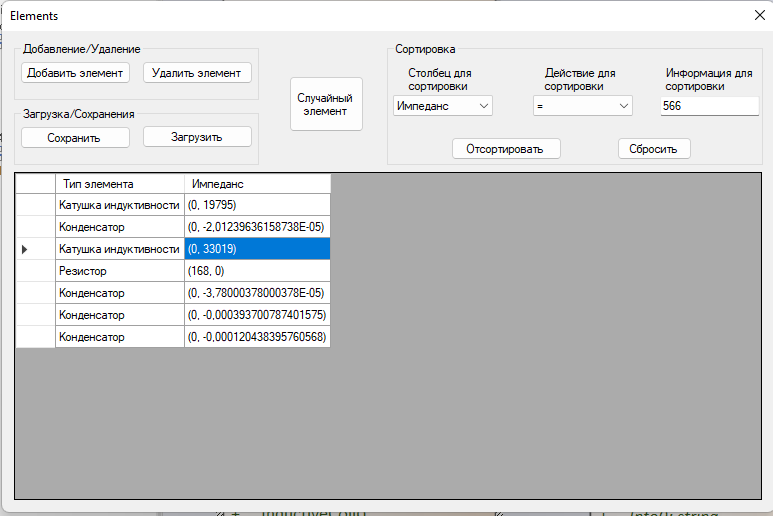


Рисунок 21 – Результат загрузки данных

В случае, если схема файла не соответствует установленному формату, появится соответствующее сообщение (рисунок 22).

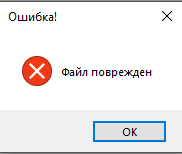


Рисунок 22 – Загрузка повреждённого файла

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Калентьев, А. А. Новые технологии в программировании : учебное пособие / А. А. Калентьев, Д. В. Гарайс, А. Е. Гориянов. – Томск : Эль Контент, 2014. – 176 с. – ISBN 978-5-4332-0185-9.

**ПРИЛОЖЕНИЕ А**

**Техническое задание на создание автоматизированной системы**

Программа для просмотра списка работников

Разработчик: студент гр. О-5КМ01 Наумкин А.В.

Заказчик: Канд. техн. наук, доцент каф. КСУП ТУСУР Калентьев А. А.

Томск 2022

**1 Общие сведения**

**1.1 Полное наименование системы и её условное обозначение**

Полное наименование: «Программа для расчета импеданса для различных элементов».

Условное обозначение: Система.

**1.2 Наименование предприятий разработчика и заказчика системы**

Заказчик: Канд. техн. наук, доцент каф. КСУП ТУСУР Калентьев А. А.

Разработчик: Студент гр. О-5КМ01 НИ ТПУ Наумкин А.В.

**1.3 Плановые сроки начала и окончания работы по созданию системы**

Начало работ: 3 апреля 2022 г.

Окончание работ: 24 июня 2022 г.

**2 Назначение и цели создания системы**

**2.1 Назначение системы**

Система предназначена для расчета импеданса различных пассивных элементов электрических схем: резистор, конденсатор, индуктивность.

**2.2 Цели создания системы**

Система создается в целях автоматизации расчета импеданса различных пассивных элементов электрических схем: резистор, конденсатор, индуктивность.

**3 Характеристика объектов автоматизации**

Раньше студентам необходимо было вручную проводить расчет импеданса различных пассивных элементов электрических схем (резистор, конденсатор, индуктивность). Этот монотонный труд мог приводить к ошибкам, которые приводят к потерям баллов и эмоциональному расстройству студента. С целью ликвидации данных ситуаций разрабатывается данная система.

**4 Требования к системе**

Таблица 4.1 – Префиксы мнемонических идентификаторов требований и их расшифровка

|  |  |
| --- | --- |
| Префикс | Тип требования |
| A | Архитектурное требование |
| С | Требование к программной или аппаратной совместимости |
| D | Требование к структуре данных |
| F | Функциональное требование |
| U | Требование к пользовательскому интерфейсу |

**4.1 Требования к архитектуре**

**А01.** Система должна быть реализована в виде настольного приложения.

**4.2 Требования к структуре данных**

**D01.** Данные о параметрах элементов электрических схем должны храниться в XML-файле с расширением \*.xml.

**D01.01.** Формат XML-файла должен соответствовать следующей XSD-схеме:

<ArrayOfElementBase xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<ElementBase xsi:type="InductiveСoil">

<TypeOfElements>Катушка индуктивности</TypeOfElements>

<Impedance/>

<Frequency>37</Frequency>

<Inductance>535</Inductance>

</ElementBase>

<ElementBase xsi:type="Capacitor">

<TypeOfElements>Конденсатор</TypeOfElements>

<Impedance/>

<Frequency>82</Frequency>

<Capacitance>606</Capacitance>

</ElementBase>

<ElementBase xsi:type="InductiveСoil">

<TypeOfElements>Катушка индуктивности</TypeOfElements>

<Impedance/>

<Frequency>89</Frequency>

<Inductance>371</Inductance>

</ElementBase>

<ElementBase xsi:type="Resistor">

<TypeOfElements>Резистор</TypeOfElements>

<Impedance/>

<Resistance>168</Resistance>

</ElementBase>

<ElementBase xsi:type="Capacitor">

<TypeOfElements>Конденсатор</TypeOfElements>

<Impedance/>

<Frequency>55</Frequency>

<Capacitance>481</Capacitance>

</ElementBase>

<ElementBase xsi:type="Capacitor">

<TypeOfElements>Конденсатор</TypeOfElements>

<Impedance/>

<Frequency>20</Frequency>

<Capacitance>127</Capacitance>

</ElementBase>

<ElementBase xsi:type="Capacitor">

<TypeOfElements>Конденсатор</TypeOfElements>

<Impedance/>

<Frequency>19</Frequency>

<Capacitance>437</Capacitance>

</ElementBase>

</ArrayOfElementBase>

**4.3 Функциональные требования**

**F01.** В системе должен быть реализован список элементов.

**F01.01.** Каждый элемент должен иметь следующие обязательные параметры:

* Тип элемента;
* Параметры для расчета.

**F01.02.** В зависимости от типа элемента необходимо заполнить следующие параметры:

Для резистора:

– Сопротивление.

Для катушки индуктивности:

– Частоту электрического тока;

– Индуктивность катушки.

Для конденсатора:

– Частоту электрического тока;

– Емкость конденсатора.

**F02.** В системе должна присутствовать функция добавления элементов в список.

**F03.** В системе должна присутствовать функция удаления элементов из списка.

**F04.** В системе должна присутствовать функция поиска элементов по параметрам, указанным в **F01.01**.

**F05.** В системе должна присутствовать функция сохранения списка элементов в файл (**D01**).

**F06.** В системе должна присутствовать функция загрузки списка элементов из файла (**D01**).

**4.4 Требования к пользовательскому интерфейсу**

**U01.** Система должна иметь графический интерфейс пользователя.

**U02.** Данные должны быть представлены в табличном виде.

**U03.** В системе должна быть реализована система обработки ошибок.

**4.5 Требования к программному обеспечению**

**C01.** Система должна работать на операционной системе Windows 10 Pro 20H2. Работоспособность на других выпусках и версиях не гарантируется.

**C02.** На рабочей станции должен быть установлен .NET Framework версии 4.8

**4.6 Требования к аппаратному обеспечению**

**C03.** Процессор: процессор с частотой 1 гигагерц (ГГц) или выше

**C04.** ОЗУ: 1 ГБ для 32-разрядных систем или 2 ГБ для 64-разрядных систем

**C05.**Место на жестком диске: 16 ГБ для 32-разрядных систем или 20 ГБ для 64-разрядных систем