|  |
| --- |
| **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-5КМ91 | |  | \_\_\_\_\_\_\_\_\_\_ | |  | Логинов Д. В. |
|  | |  |  | |  |  |
|  | |  |  | 17 июня 2021 г. | | |
|  | |  |  | |  |  |
| Отчёт принял | доцент, к.т.н. |  | \_\_\_\_\_\_\_\_\_\_ | |  | Калентьев А. А. |
|  |  |  |  | |  |  |
|  |  |  |  | \_\_\_ \_\_\_\_\_\_ 2021 г. | | |

Томск 2021

**Содержание**

[**Введение** 3](#_Toc74829062)

[**1 Описание программной системы** 4](#_Toc74829063)

[1.1 UML диаграмма вариантов использования 4](#_Toc74829064)

[1.2 UML диаграмма классов 4](#_Toc74829065)

[1.3 Описание классов, образующих связь типа «общее-частное» 6](#_Toc74829066)

[1.4 Дерево ветвлений Git 9](#_Toc74829067)

[1.5 Тестирование программы 10](#_Toc74829068)

[**Список использованных источников** 22](#_Toc74829069)

[**ПРИЛОЖЕНИЕ А** 23](#_Toc74829070)

**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:

1. Составление технического задания (далее – ТЗ) на разработанную программу (ТЗ представлено в Приложении А);
2. Составление UML диаграммы вариантов использования для разработанной программы;
3. Составление UML диаграммы классов;
4. Описание классов, образующих связь типа «общее-частное»;
5. Привести дерево ветвлений Git;
6. Провести тестирование программы.

**1 Описание программной системы**

**1.1 UML диаграмма вариантов использования**

Диаграмма вариантов использования приведена на рисунке 1.

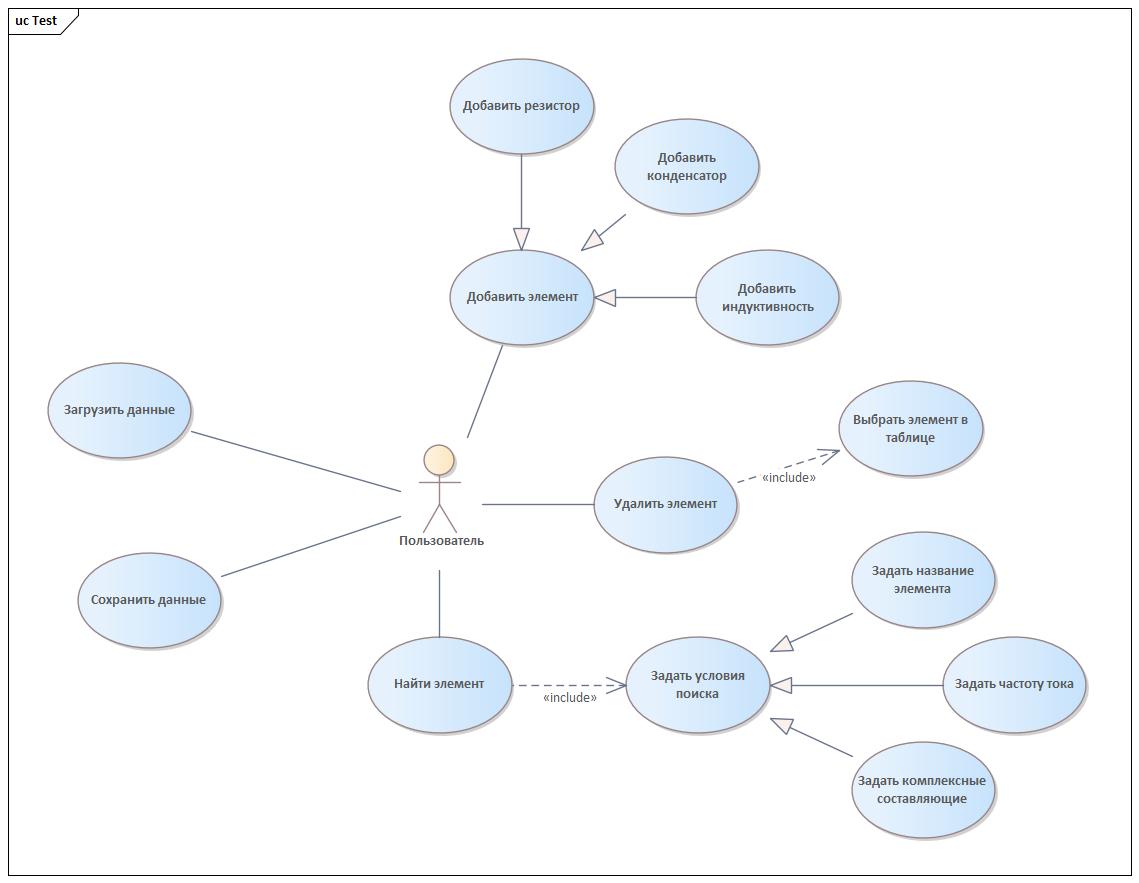


Рисунок 1 – Диаграмма вариантов использования

**1.2 UML диаграмма классов**

Диаграмма классов приведена на рисунке 2.

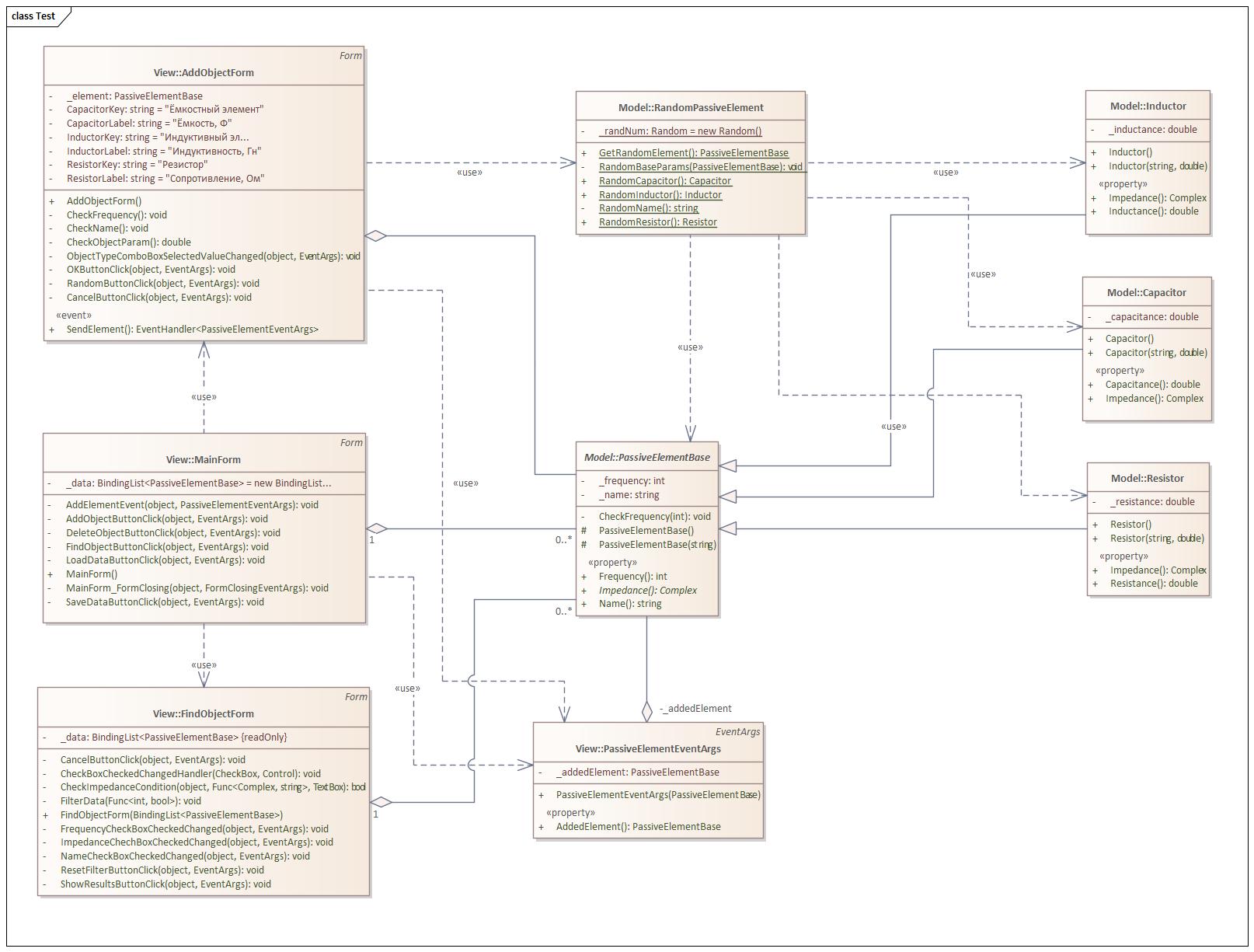


Рисунок 2 – UML диаграмма классов

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

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

Таблица 1 – Описание класса PassiveElementBase

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Описание класса | | |
| Класс PassiveElementBase – абстрактный пассивный элемент электрической схемы | | |
| Поля | | |
| – \_name | string | Наименование элемента |
| – \_frequency | int | Частота электрического тока |
| Свойства | | |
| + Name | string | Наименование элемента |
| + Frequency | int | Частота электрического тока |
| + *Impedance* | Complex | Комплексное сопротивление элемента.  Абстрактное свойство, переопределяется в производных классах |
| Методы | | |
| – CheckFrequency(int) | void | Проверяет частоту тока:  int – частота тока |
| # PassiveElementBase() |  | Конструктор класса PassiveElementBase без параметров для сериализации |
| # PassiveElementBase(string) |  | Конструктор класса PassiveElementBase:  string – наименование элемента |

В таблицах 2–4 приведены описания классов Resistor, Capacitor и Inductor, которые наследуются от PassiveElementBase.

Таблица 2 – Описание класса Resistor

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Описание класса | | |
| Класс Resistor – резистор | | |
| Поля | | |
| – \_resistance | double | Активное сопротивление |

Окончание таблицы 2

|  |  |  |
| --- | --- | --- |
| Свойства | | |
| + Resistance | double | Активное сопротивление |
| + Impedance | Complex | Комплексное сопротивление резистора |
| Методы | | |
| + Resistor() |  | Конструктор класса Resistor без параметров для сериализации |
| + Resistor(string, double) |  | Конструктор класса Resistor:  string – наименование элемента;  double – активное сопротивление.  Вызывает конструктор базового класса |

Таблица 3 – Описание класса Capacitor

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Описание класса | | |
| Класс Capacitor – конденсатор | | |
| Поля | | |
| – \_capacitance | double | Ёмкость |
| Свойства | | |
| + Capacitance | double | Ёмкость |
| + Impedance | Complex | Комплексное сопротивление ёмкостного элемента |
| Методы | | |
| + Capacitor() |  | Конструктор класса Capacitor без параметров для сериализации |
| + Capacitor(string, double) |  | Конструктор класса Capacitor:  string – наименование элемента;  double – ёмкость.  Вызывает конструктор базового класса |

Таблица 4 – Описание класса Inductor

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Описание класса | | |
| Класс Inductor – Индуктивность | | |
| Поля | | |
| – \_inductance | double | Индуктивность |
| Свойства | | |
| + Inductance | double | Индуктивность |
| + Impedance | Complex | Комплексное сопротивление индуктивного элемента |

Окончание таблицы 4

|  |  |  |
| --- | --- | --- |
| Методы | | |
| + Inductor() |  | Конструктор класса Inductor без параметров для сериализации |
| + Inductor(string, double) |  | Конструктор класса Inductor:  string – наименование элемента;  double – индуктивность.  Вызывает конструктор базового класса |

**1.4 Дерево ветвлений Git**

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

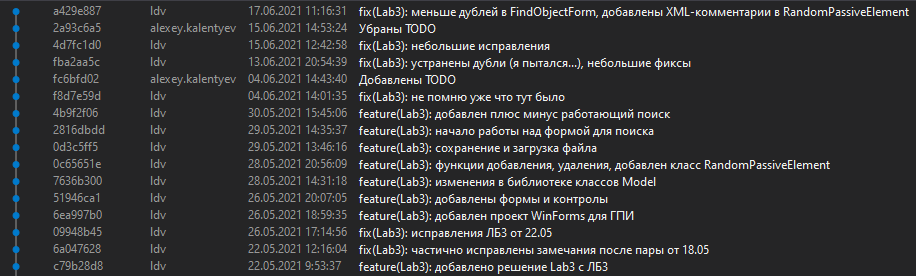


Рисунок 3 – Дерево ветвлений Git

**1.5 Тестирование программы**

Далее приводится процесс функционального тестирования программы.

Графический интерфейс пользователя представлен на рисунке 4.

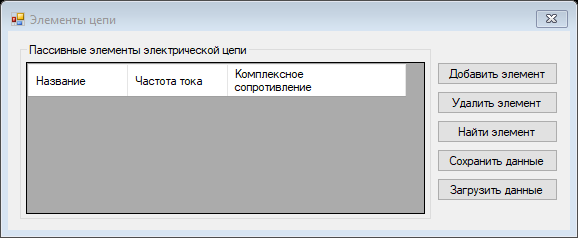


Рисунок 4 – Графический интерфейс пользователя

**1.5.1 Тестовый случай «Добавить элемент»**

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

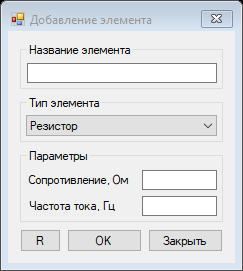


Рисунок 5 – Форма для добавления элемента

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

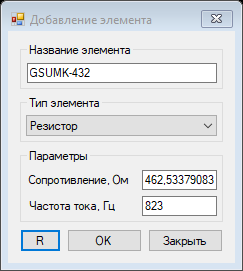


Рисунок 6 – Заполнение полей

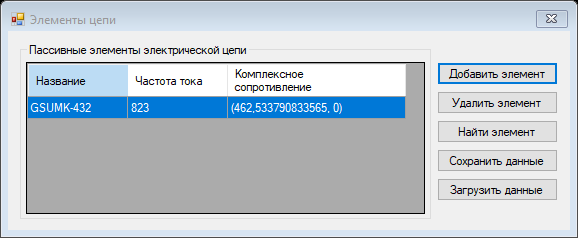


Рисунок 7 – Успешное добавление нового элемента

В программе предусмотрена система обработки некорректного ввода данных пользователем. Например, при попытке ввести отрицательное значение частоты появится сообщение об ошибке (рисунки 8 и 9).

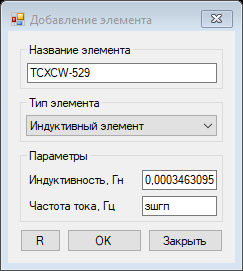


Рисунок 8 – Некорректный ввод частоты

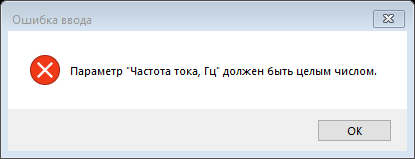


Рисунок 9 – Сообщение об ошибке

Подобная обработка предусмотрена также для названия элемента и его специфичного параметра, например, для активного сопротивления у резистора.

**1.5.2 Тестовый случай «Удалить элемент»**

Для удаления элемента необходимо выбрать его в таблице и нажать на кнопку «Удалить элемент» (рисунки 10 и 11).

Также предусмотрена возможность удаления нескольких выделенных элементов (рисунки 12 и 13).

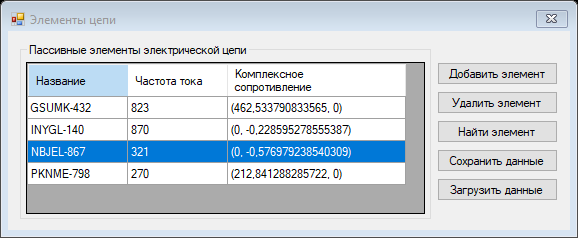


Рисунок 10 – Выбор элемента в таблице

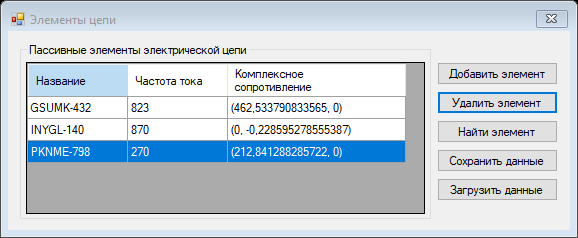


Рисунок 11 – Результат нажатия кнопки «Удалить элемент»

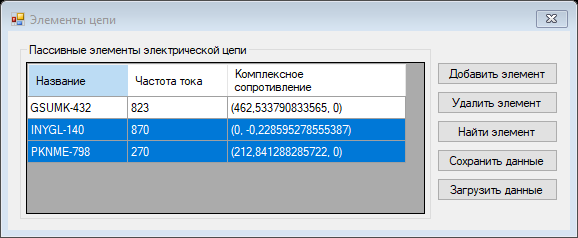


Рисунок 12 – Выбор нескольких элементов для удаления

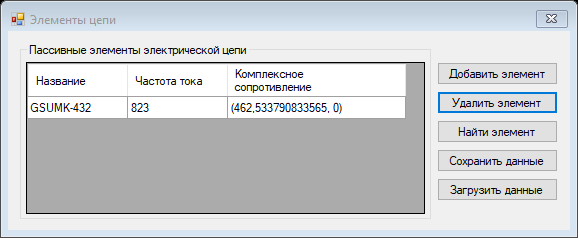


Рисунок 13 – Результат удаления выбранных элементов

**1.5.3 Тестовый случай «Найти элемент»**

Для поиска элементов в списке необходимо нажать кнопку «Найти элемент» (рисунок 14). Откроется соответствующая форма для поиска (рисунок 15).

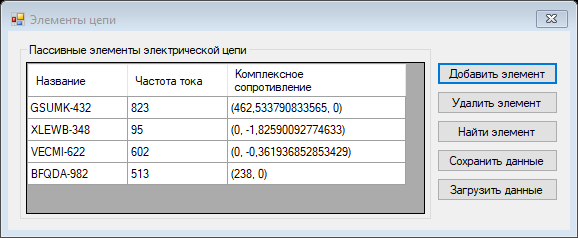


Рисунок 14 – Вызов формы для поиска элементов

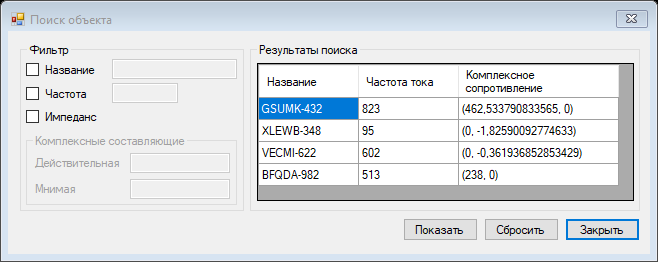


Рисунок 15 – Форма для поиска

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

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

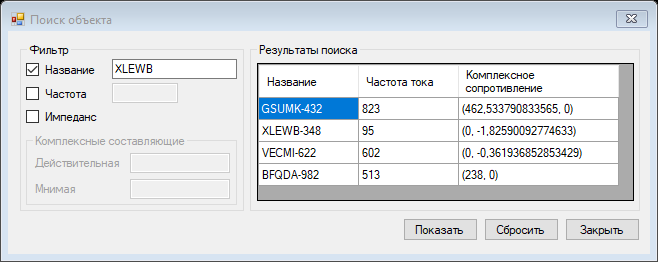


Рисунок 16 – Поиск элемента по названию

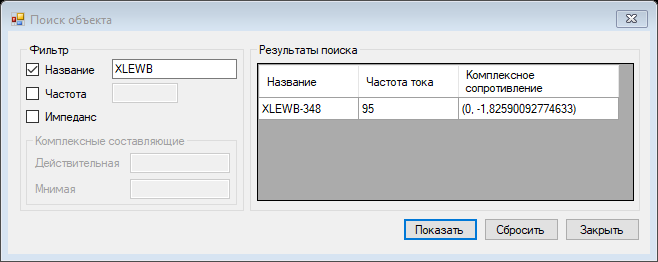


Рисунок 17 – Результат поиска элемента

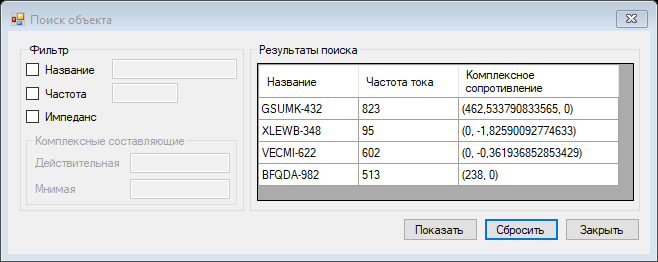


Рисунок 18 – Сброс условий поиска

**1.5.4 Тестовый случай «Сохранить данные»**

Для сохранения данных в таблице необходимо нажать на соответствующую кнопку на форме (рисунок 19). Откроется системный диалог сохранения файла, где пользователь выбирает директорию и указывает имя файла (рисунок 20).

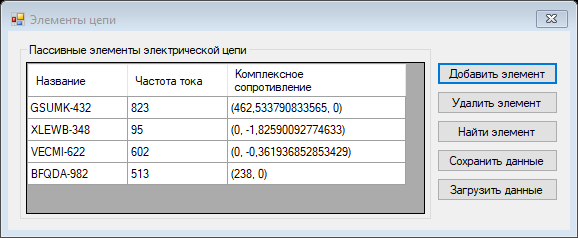


Рисунок 19 – Сохранение данных

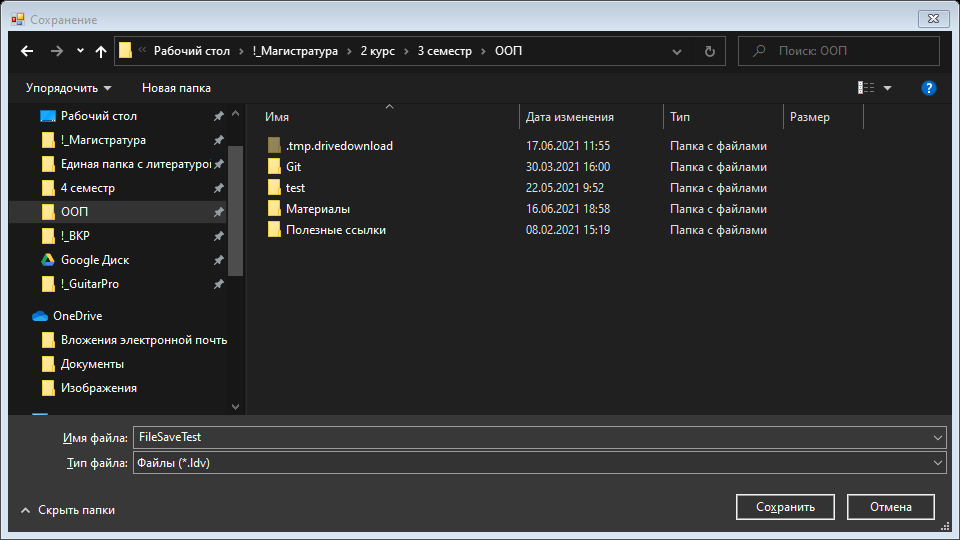


Рисунок 20 – Сохранение файла

После сохранения данных в файл появится соответствующее сообщение (рисунки 21 и 22).

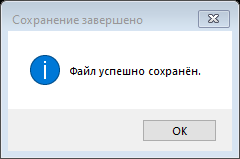


Рисунок 21 – Сообщение о сохранении файла

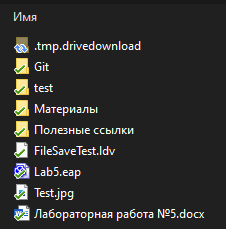


Рисунок 22 – Результат сохранения файла

В случае, если таблица пуста, сохранение не производится (рисунок 23).

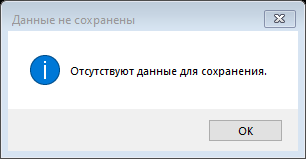


Рисунок 23 – Результат нажатия на кнопку «Сохранить данные» при пустой таблице

**1.5.5 Тестовый случай «Загрузить данные»**

Для загрузки данных в таблицу необходимо нажать на соответствующую кнопку (рисунок 24).

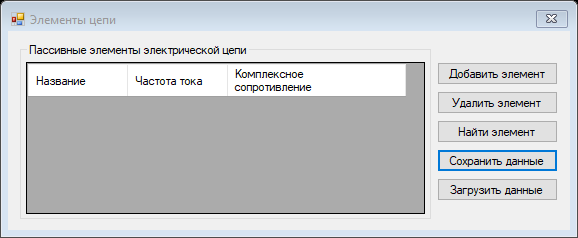


Рисунок 24 – Загрузить данные в таблицу

Далее откроется системный диалог загрузки файла (рисунок 25). После успешной загрузки появится сообщение (рисунки 26 и 27).

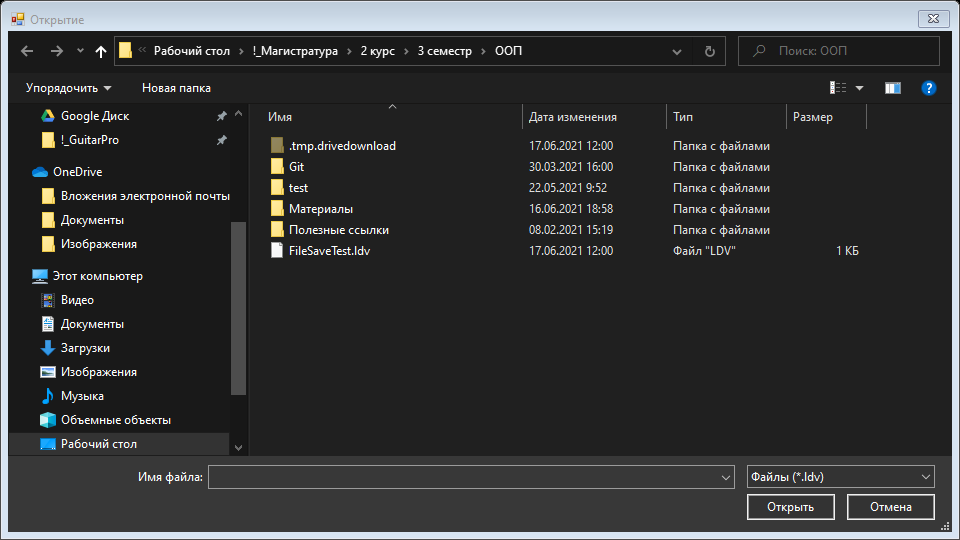


Рисунок 25 – Выбор файла для загрузки

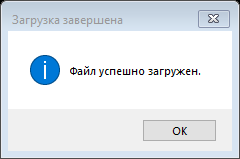


Рисунок 26 – Сообщение, информирующее об успешной загрузке файла

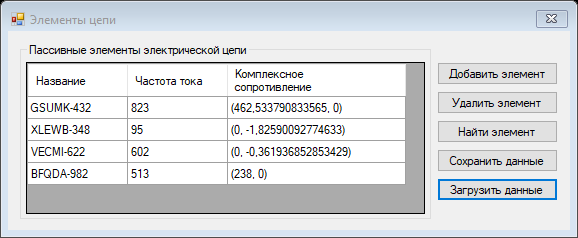


Рисунок 27 – Результат загрузки данных

В случае, если XSD схема файла не соответствует установленному формату, появится соответствующее сообщение (рисунок 28).

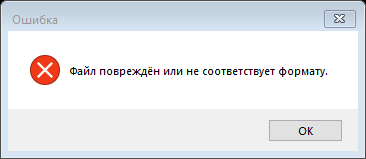


Рисунок 28 – Загрузка повреждённого файла

**1.5.6 Тестовый случай «Завершение работы с программой»**

В случае, если в таблице имеются какие-либо данные, то при выходе из программы появится информационное сообщение (рисунок 29).

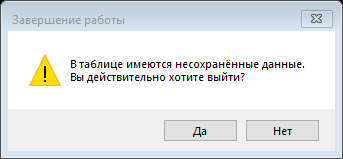


Рисунок 29 – Выход из программы

**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 расчёта комплексного сопротивления пассивных элементов электрических схем

Разработчик: студент гр. О-5КМ91 НИ ТПУ Логинов Д.В.

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

Томск, 2021

**1 Общие сведения**

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

Полное наименование: «Программа для расчёта комплексного сопротивления пассивных элементов электрических схем».

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-5КМ91 НИ ТПУ Логинов Д.В.

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

Начало работ: 22 апреля 2021 г.

Окончание работ: 11 июня 2021 г.

**2 Назначение и цели создания системы**

**2.1 Назначение системы**

Система предназначена для расчёта комплексного сопротивления различных пассивных элементов электрических схем: резистор, конденсатор, индуктивность.

**2.2 Цели создания системы**

Система создаётся в целях сокращения трудозатрат специалистов при расчётах параметров пассивных элементов электрических схем.

**3 Характеристика объектов автоматизации**

Расчёт параметров элементов электрических схем является важной задачей при проведении схемотехнических работ. Поскольку такие расчёты выполняются специалистами при проектировании схем ежедневно, представляется целесообразным автоматизировать этот процесс.

**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 \*.ldv.

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

<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">

<xs:element name="ArrayOfPassiveElementBase">

<xs:complexType>

<xs:sequence>

<xs:element name="PassiveElementBase" maxOccurs="unbounded" minOccurs="0">

<xs:complexType>

<xs:sequence>

<xs:element type="xs:string" name="Name"/>

<xs:element type="xs:short" name="Frequency"/>

<xs:element type="xs:float" name="Capacitance" minOccurs="0"/>

<xs:element type="xs:float" name="Inductance" minOccurs="0"/>

<xs:element type="xs:float" name="Resistance" minOccurs="0"/>

</xs:sequence>

</xs:complexType>

</xs:element>

</xs:sequence>

</xs:complexType>

</xs:element>

</xs:schema>

**4.3 Функциональные требования**

**F01.** Система должна рассчитывать комплексное сопротивление для следующих элементов:

* резистор;
* конденсатор;
* индуктивность.

**F01.01.** Комплексное сопротивление резистора должно определяться по выражению



где *R* – активное сопротивление резистора, Ом.

**F01.02.** Комплексное сопротивление конденсатора должно определяться по выражению



где *f* – частота электрического тока;

*C* – ёмкость конденсатора, Ф.

**F01.03.** Комплексное сопротивление индуктивного элемента должно определяться по выражению



где *f* – частота электрического тока;

*L* – индуктивность, Гн.

**F02.** В системе должен быть реализован список элементов электрических схем.

**F02.01.** Каждый элемент должен иметь следующие параметры:

* наименование;
* частота электрического тока;
* комплексное сопротивление.

**F03.** В системе должна присутствовать функция добавления элементов в список.

**F04.** В системе должна присутствовать функция удаления элементов из списка.

**F05.** В системе должна присутствовать функция поиска элементов по параметрам, указанным в **F02.01**.

**F06.** В системе должна присутствовать функция сохранения списка элементов в файл (**D01**).

**F07.** 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 4.7.2.

**4.6 Требования к аппаратному обеспечению**

**C03.** Процессор – не менее 1 ГГц.

**C04.** ОЗУ – не менее 2 ГБ.