

**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-5КМ11

\_\_\_\_\_

Копцев И.О.

Отчёт принял доцент, к.т.н.

\_\_\_\_\_

Калентьев А.А.

## Содержание

ВВЕДЕНИЕ.....	3
1 Основная часть .....	4
1.1 UML диаграмма вариантов использования.....	4
1.2 UML диаграмма классов .....	4
1.3 Описание классов, образующих связь типа «общее-частное».....	6
1.4 Дерево ветвлений Git.....	7
1.5 Тестирование программы.....	8
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	Ошибка! Закладка не определена.
ПРИЛОЖЕНИЕ А .....	18

## 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 — описание множества последовательных действий, которые выполняются некоторым субъектом с целью получения результата, значимого для некоторого действующего лица. 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 ПО приведена на рис. 1.



Рисунок 1. Диаграмма вариантов использования

### 1.2 UML диаграмма классов

Диаграмма классов — это 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 рис. 2.

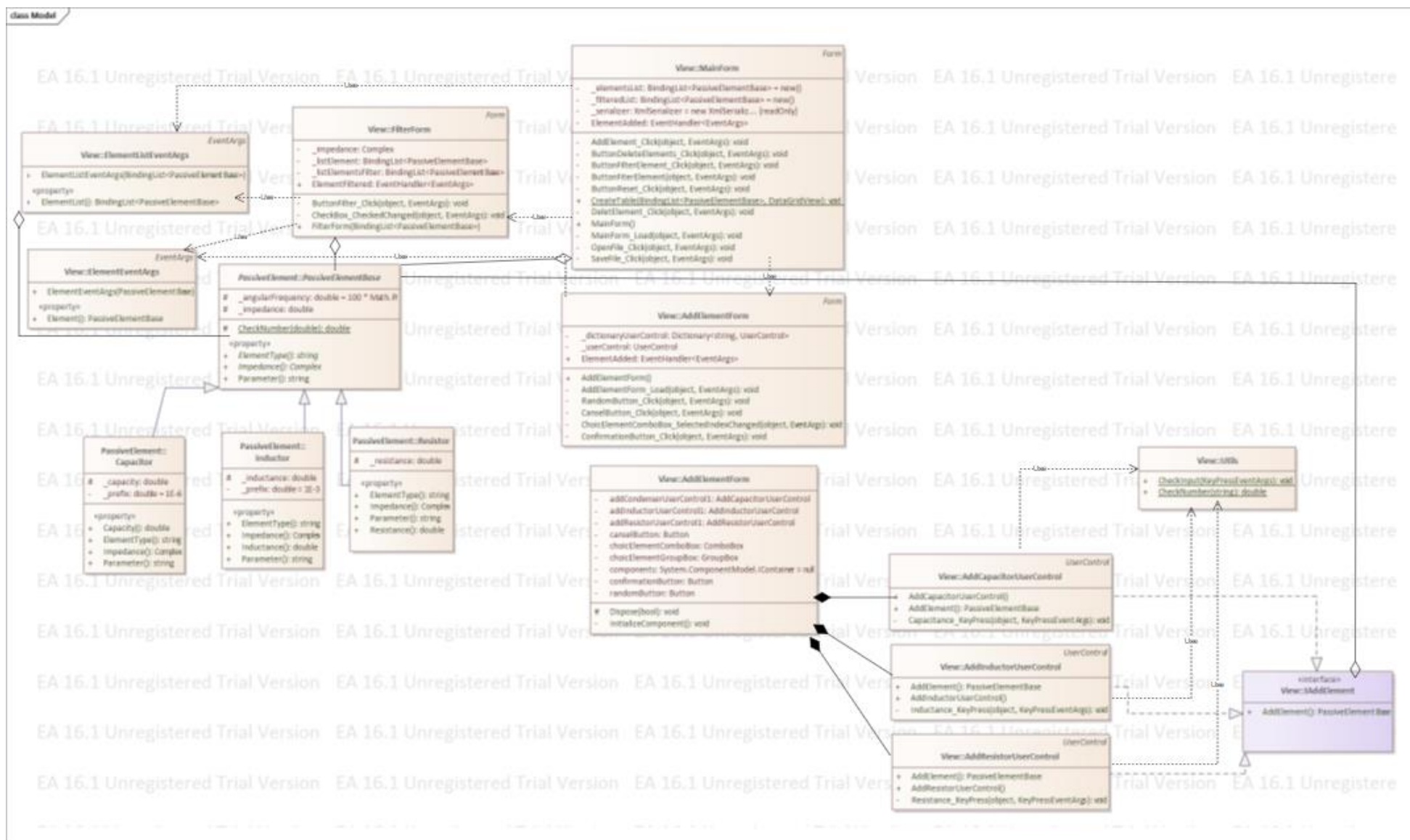


Рисунок 2. UML диаграмма классов

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

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

Таблица 1 – Описание класса *PassiveElementBase*

Название	Тип	Описание
Описание класса		
Класс <i>PassiveElementBase</i> – абстрактный базовый класс для описания пассивных элементов электрической цепи		
Свойства		
+ ElementType	string	Тип пассивного элемента. Абстрактное свойство, переопределяется в производных классах.
+ Parameter	string	Параметр пассивного элемента. Абстрактное свойство, переопределяется в производных классах.
+ Impedance	Complex	Сопротивление пассивного элемента. Абстрактное свойство, переопределяется в производных классах.
Методы		
# <u>CheckNumber</u>	double	Проверяет корректность ввода параметров пассивных элементов double – любой параметр пассивного элемента

В табл. 2–4 приведены описания классов *Resistor*, *Inductor* и *Capasitor*, которые наследуются от *PassiveElementBase*.

Таблица 2 – Описание класса *Resistor*

Название	Тип	Описание
Описание класса		
Класс <i>Resistor</i> – резистивный элемент		
Поля		
– _resistance	double	Активное сопротивление
Свойства		
+ Resistance	double	Активное сопротивление
+ ElementType	string	Тип пассивного элемента
+ Parametrs	string	Параметр пассивного элемента
+ Impedance	Complex	Вычисление сопротивления пассивного элемента

Таблица 3 – Описание класса *Inductor*

Название	Тип	Описание
Описание класса		
Класс <i>Inductor</i> – катушка индуктивности		
Поля		
– <i>_inductance</i>	double	Индуктивность
Свойства		
+ <i>Inductance</i>	double	Индуктивность
+ <i>ElementType</i>	string	Тип пассивного элемента
+ <i>Parametrs</i>	string	Параметр пассивного элемента
+ <i>Impedance</i>	Complex	Вычисление сопротивления пассивного элемента

Таблица 4 – Описание класса *Capasitor*

Название	Тип	Описание
Описание класса		
Класс <i>Capasitor</i> – шар		
Поля		
– <i>_capacity</i>	double	Ёмкость
Свойства		
+ <i>Capacity</i>	double	Индуктивность
+ <i>ElementType</i>	string	Тип пассивного элемента
+ <i>Parametrs</i>	string	Параметр пассивного элемента
+ <i>Impedance</i>	Complex	Вычисление сопротивления пассивного элемента

## 1.4. Дерево ветвлений Git

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

fix(View): изменены размеры форм AddUserControl	LB4	origin/LB4	KoptsevIO	04.11.202...	64502b6d
fix(Model): TODO исправлены			KoptsevIO	04.10.202...	44ed394c
Добавлены TODO			ikalentii	04.10.202...	b5aeb73
fix(View): исправлено отображение AddElementForm			KoptsevIO	03.10.202...	06642392
fix(View): исправлено отображение MainForm			KoptsevIO	03.10.202...	a062f232
fix(View): отредактирована MainForm			KoptsevIO	02.10.202...	f3bb92b
feature(Model): добавлено округление выводимых значений в DataGrid MainForm			KoptsevIO	02.10.202...	fd531434
fix(Model): TODO: rename complite			KoptsevIO	02.10.202...	51c970c3
fix(View): TODO: duplication complite			KoptsevIO	02.10.202...	58deaf1f
fix(View): TODO: duplication complite			KoptsevIO	02.10.202...	4ee80a2c
fix(View):			KoptsevIO	02.10.202...	fc72352
fix(View): TODO: encapsulation complite			KoptsevIO	02.10.202...	32cd3dd1
fix(View): TODO: RSDN complite			KoptsevIO	02.10.202...	796d74bf
fix(View): TODO: remove			KoptsevIO	02.10.202...	b9e1100f
fix(View): TODO: remove complite			KoptsevIO	02.10.202...	030d0bfd
fix(View): исправлены TODO			KoptsevIO	02.10.202...	b004a8ab
Добавлены TODO			ikalentii	28.09.202...	11edea62
fix(View): исправлены размеры форм			KoptsevIO	28.09.202...	7968c83e
fix(PassiveElement): добавлено округление до целого			KoptsevIO	28.09.202...	678a8ef6
fix(Model): добавлены пропущенные XML-комментарии			KoptsevIO	28.09.202...	ae5eccd5
fix(View): исправлена фильтрация по значению комплексного сопротивления			KoptsevIO	27.09.202...	5f712085

Рисунок 3. Дерево ветвлений Git

## 1.5. Тестирование программы

Далее приводится процесс функционального тестирования программы. Графический интерфейс пользователя представлен на рис. 4.

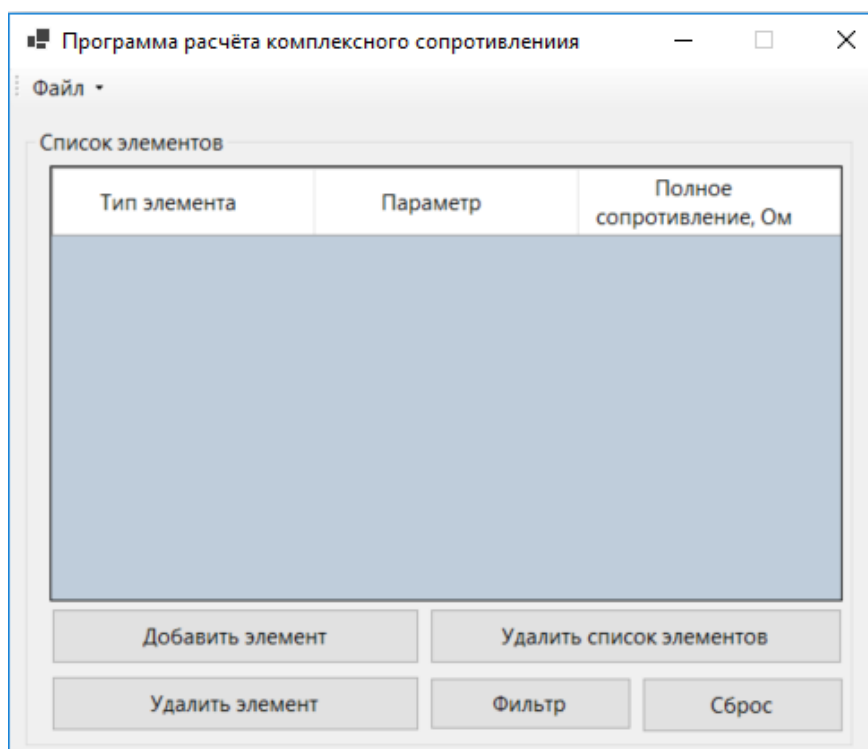


Рисунок 4. Графический интерфейс пользователя

### 1.5.1. Тестовый случай «Добавить элемент»

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

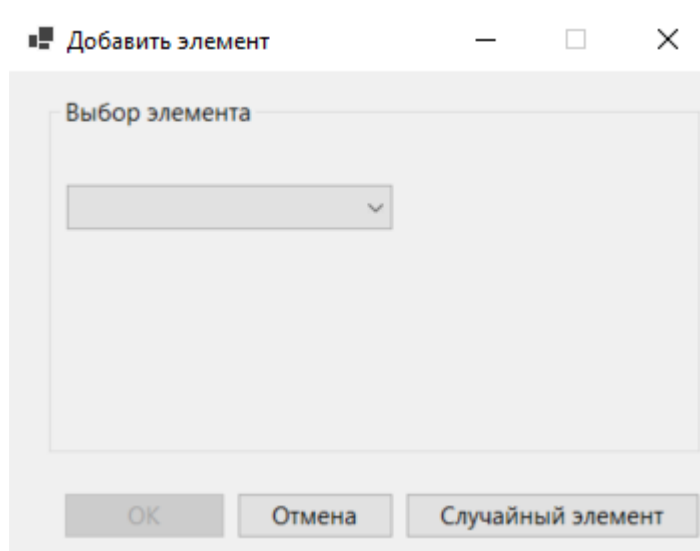


Рисунок 5. Форма для добавления элемента



Параметры любого из выбранных пассивных элементов можно ввести, выбрав соответствующий элемент в выпадающем меню. После ввода данных необходимо нажать кнопку «ОК», элемент появится в таблице главной формы (рис. 6 и 7).

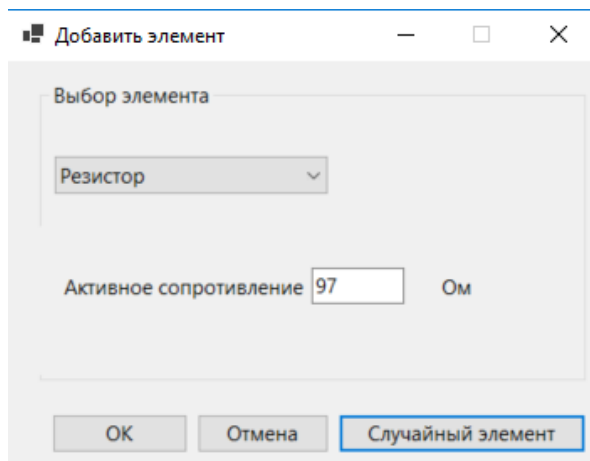
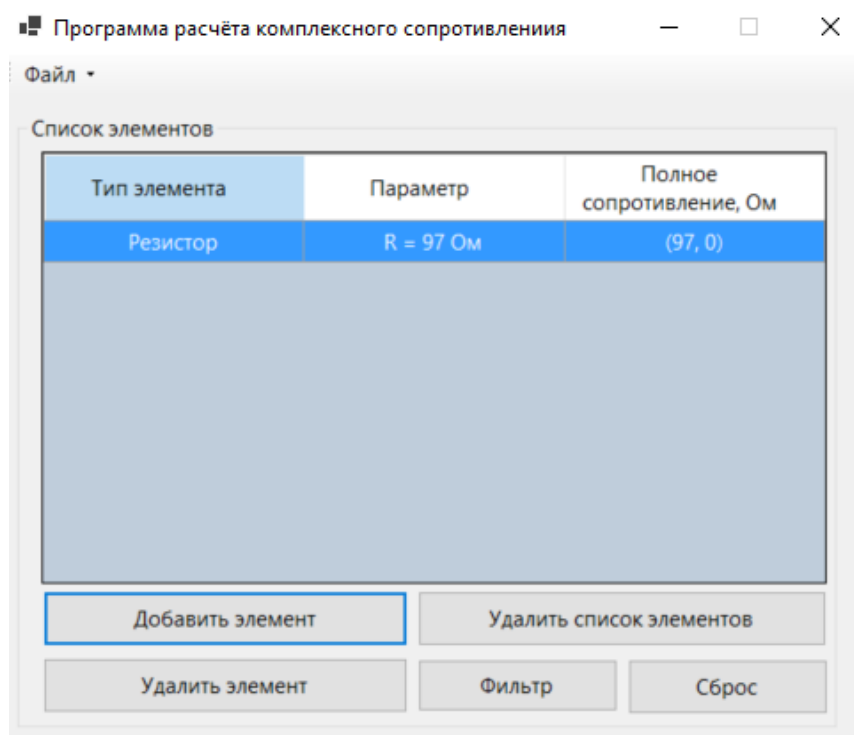


Рисунок 6. Заполнение полей



Тип элемента	Параметр	Полное сопротивление, Ом
Резистор	R = 97 Ом	(97, 0)

Рисунок 7. Успешное добавление нового элемента

В программе предусмотрена система обработки некорректного ввода данных пользователем. Например, при попытке ввести отрицательное число появится соответствующее сообщение об ошибке (рис. 8-9). Аналогичным образом обрабатывается попытка ввода Null.

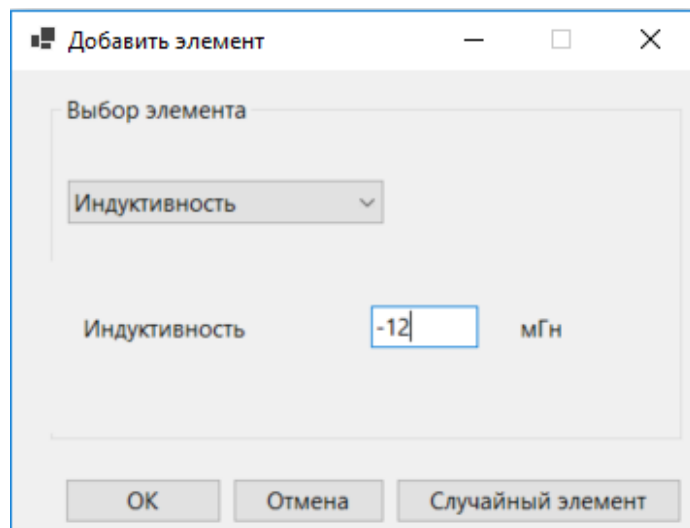


Рисунок 8. Некорректный ввод

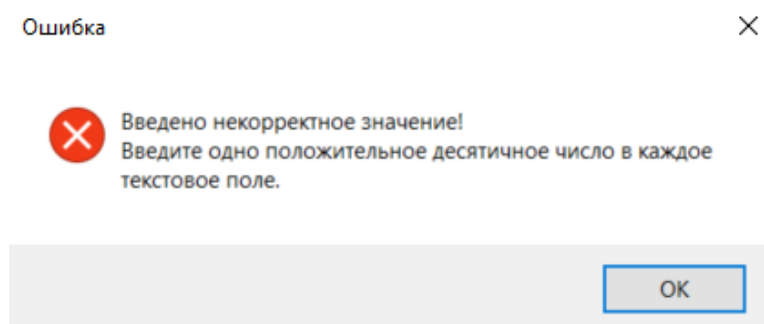


Рисунок 9. Сообщение об ошибке

### 1.5.2. Тестовый случай «Удалить элемент»

Для удаления одного или нескольких элементов необходимо выбрать их в таблице и нажать на кнопку «Удалить элемент» (рис. 10 и 11).

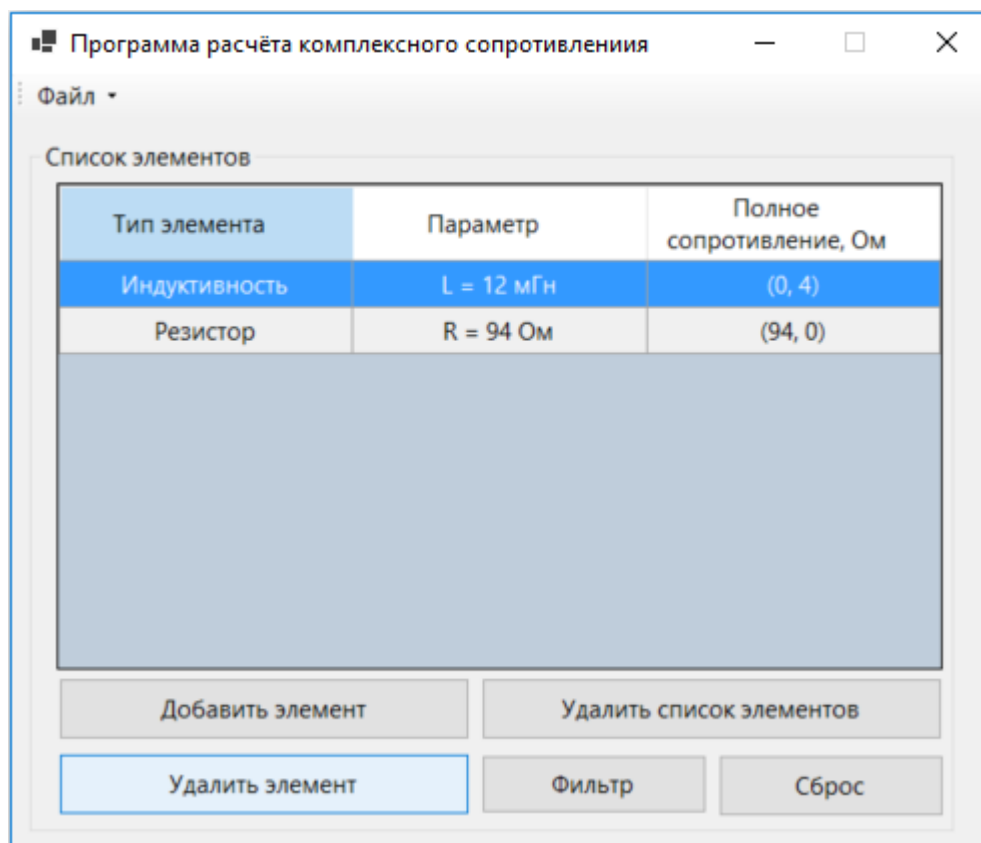


Рисунок 10. Выбор элемента в таблице

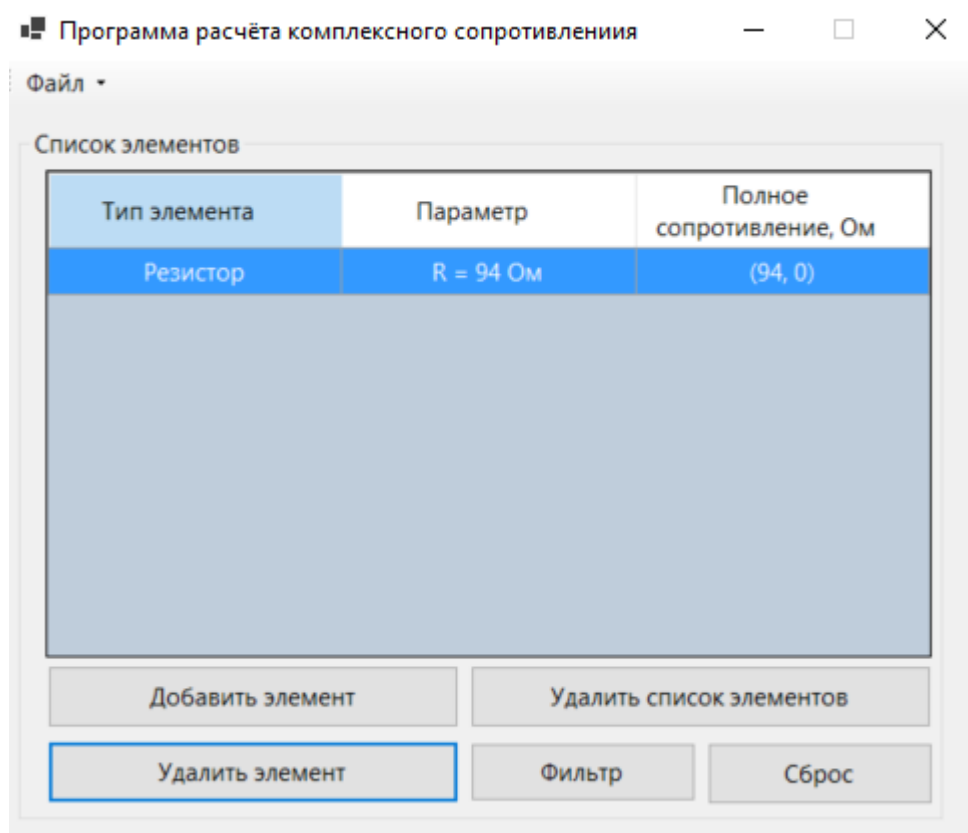


Рисунок 11. Результат нажатия кнопки «Удалить элемент»

### 1.5.3. Тестовый случай «Найти элемент»

Для поиска элементов в списке необходимо нажать кнопку «Фильтр» (рис. 12). Откроется соответствующая форма для поиска (рис. 15).

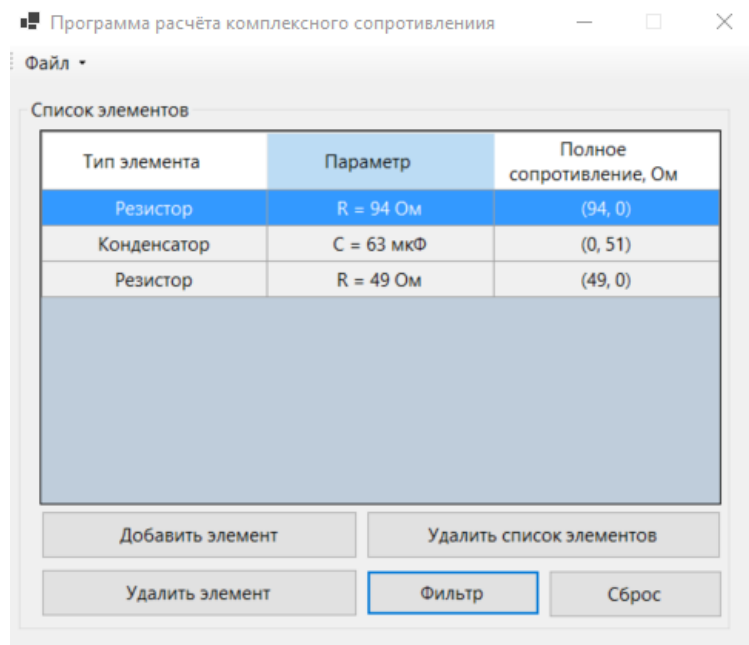


Рисунок 14. Вызов формы для поиска элементов

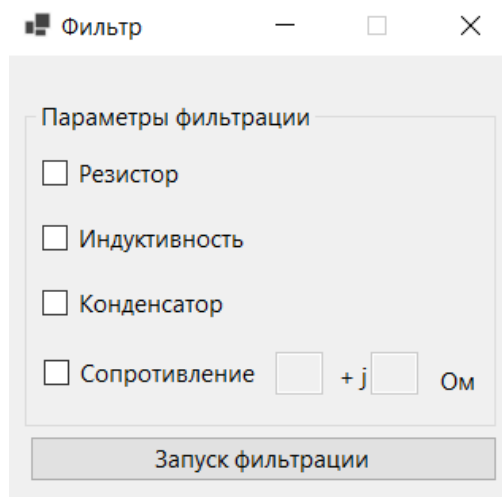


Рисунок 15. Форма для поиска

Далее пользователь выбирает параметры, по которым требуется найти элемент, вводит значения этих параметров и нажимает кнопку «Запуск фильтра» (рис. 16 и 17). Для того, чтобы сбросить фильтр, предусмотрена соответствующая кнопка «Сбросить» (рис. 18).

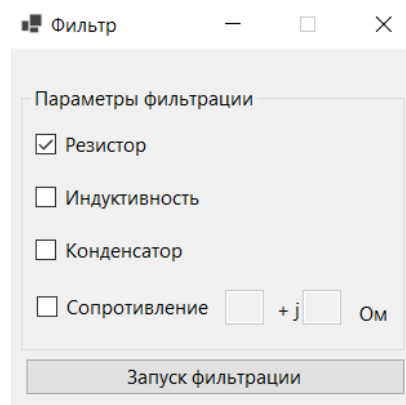


Рисунок 16. Поиск элемента по названию

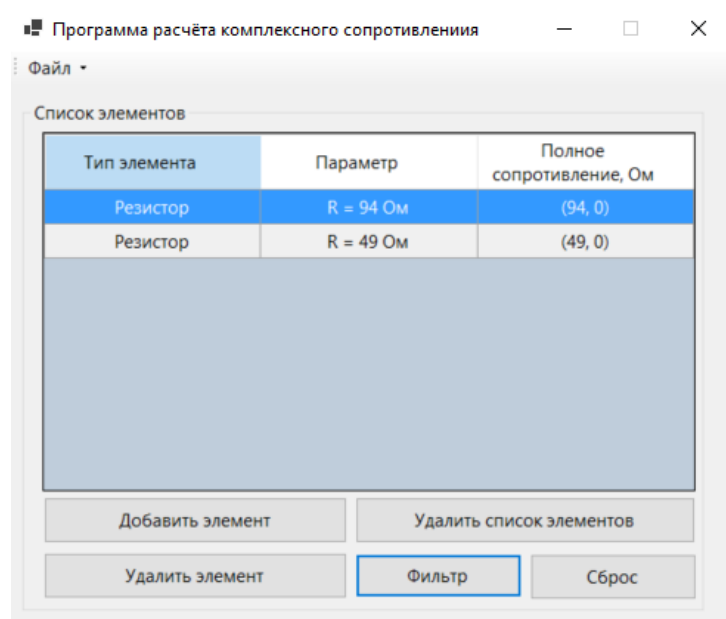


Рисунок 17. Результат поиска элемента

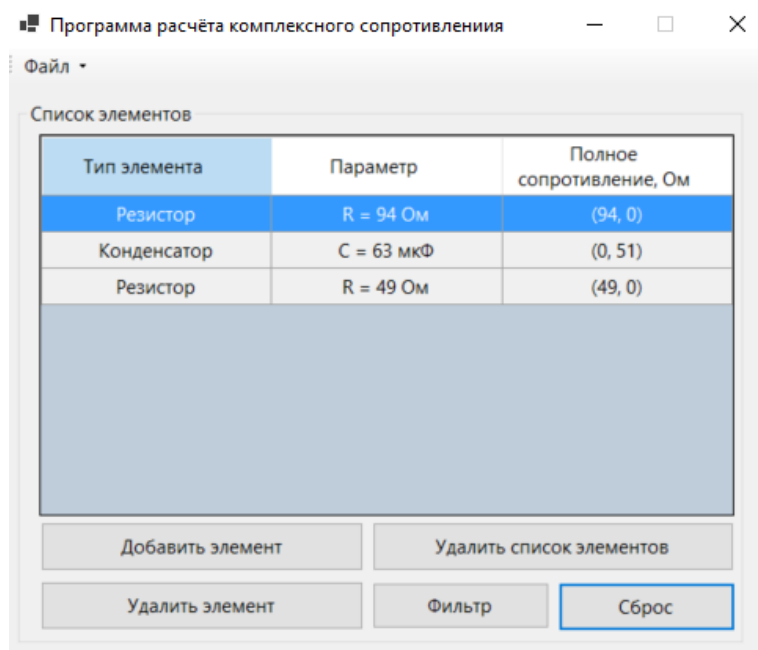


Рисунок 18. Сброс условий поиска

#### 1.5.4. Тестовый случай «Сохранить данные»

Для сохранения данных в таблице необходимо нажать на панели «Файл \ Сохранить» (рис. 19). Откроется системный диалог сохранения файла, где пользователь выбирает директорию и указывает имя файла (рис. 20).

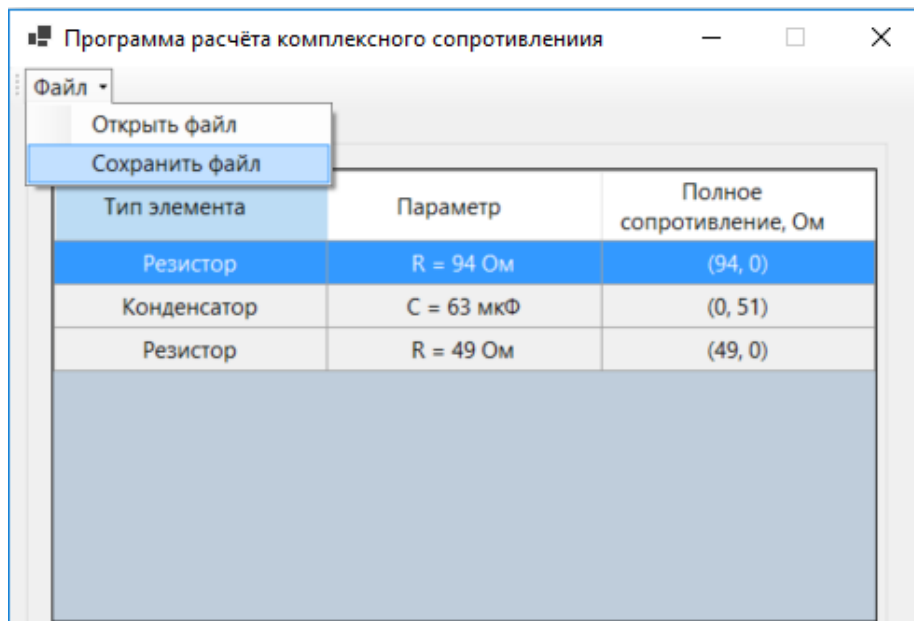


Рисунок 19. Сохранение данных

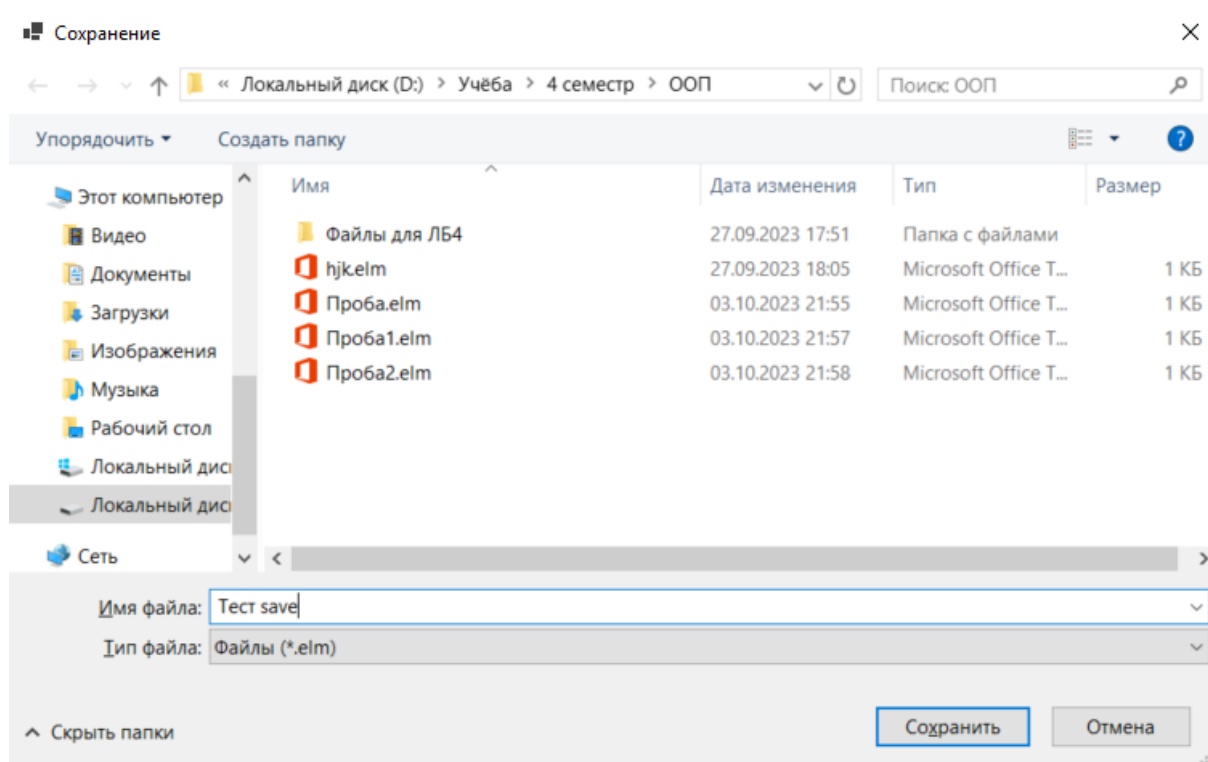


Рисунок 20. Сохранение файла

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

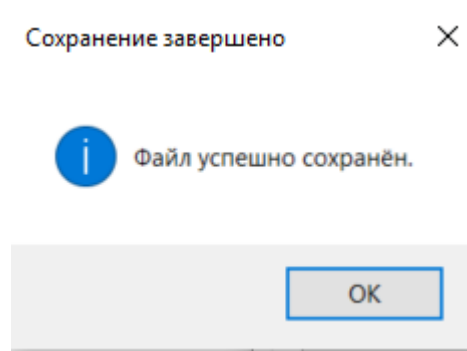


Рисунок 21. Сообщение о сохранении файла

Проба.elm	03.10.2023 21:55	Microsoft Office T...	1 КБ
Проба1.elm	03.10.2023 21:57	Microsoft Office T...	1 КБ
Проба2.elm	03.10.2023 21:58	Microsoft Office T...	1 КБ
Тест save.elm	04.11.2023 22:38	Microsoft Office T...	1 КБ

Рисунок 22. Результат сохранения файла

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

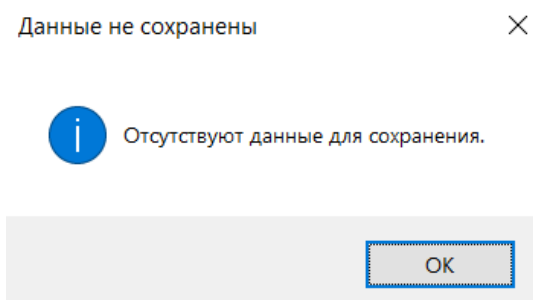


Рисунок 23. Результат нажатия на кнопку «Сохранить» при пустой таблице

### 1.5.5. Тестовый случай «Загрузить данные»

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

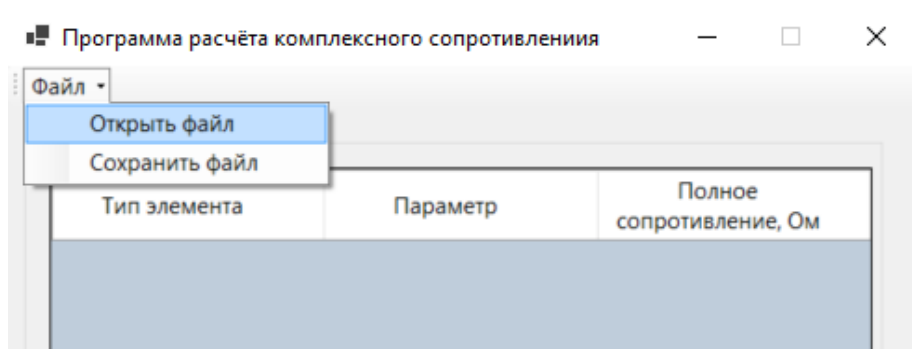


Рисунок 24. Загрузить данные в таблицу

Далее откроется диалоговое окно загрузки файла (рис. 25). После успешной загрузки появится сообщение (рис. 26 и 27).

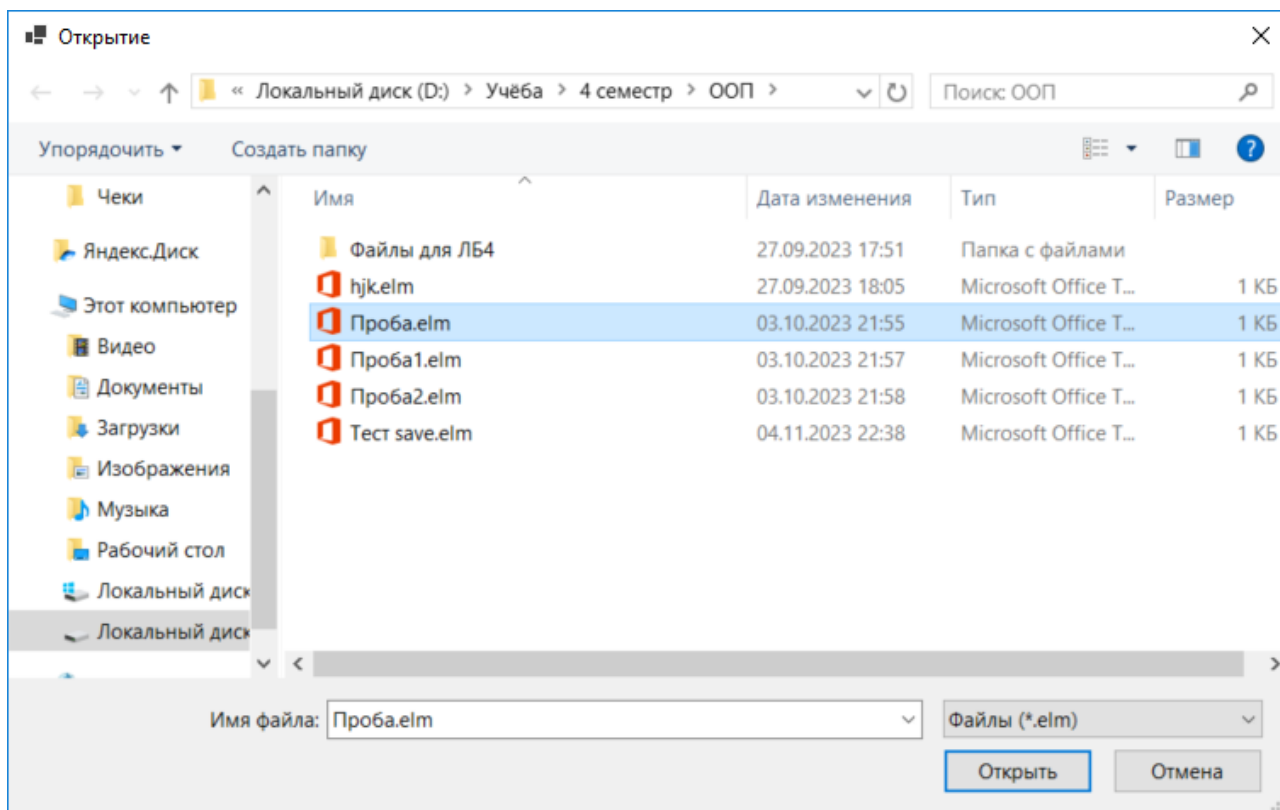


Рисунок 25. Выбор файла для загрузки

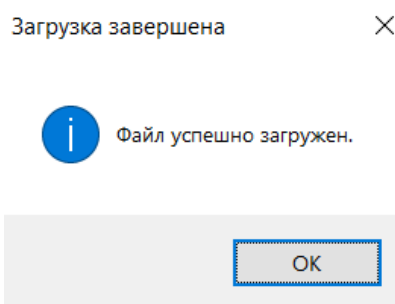


Рисунок 26. Сообщение, информирующее об успешной загрузке файла



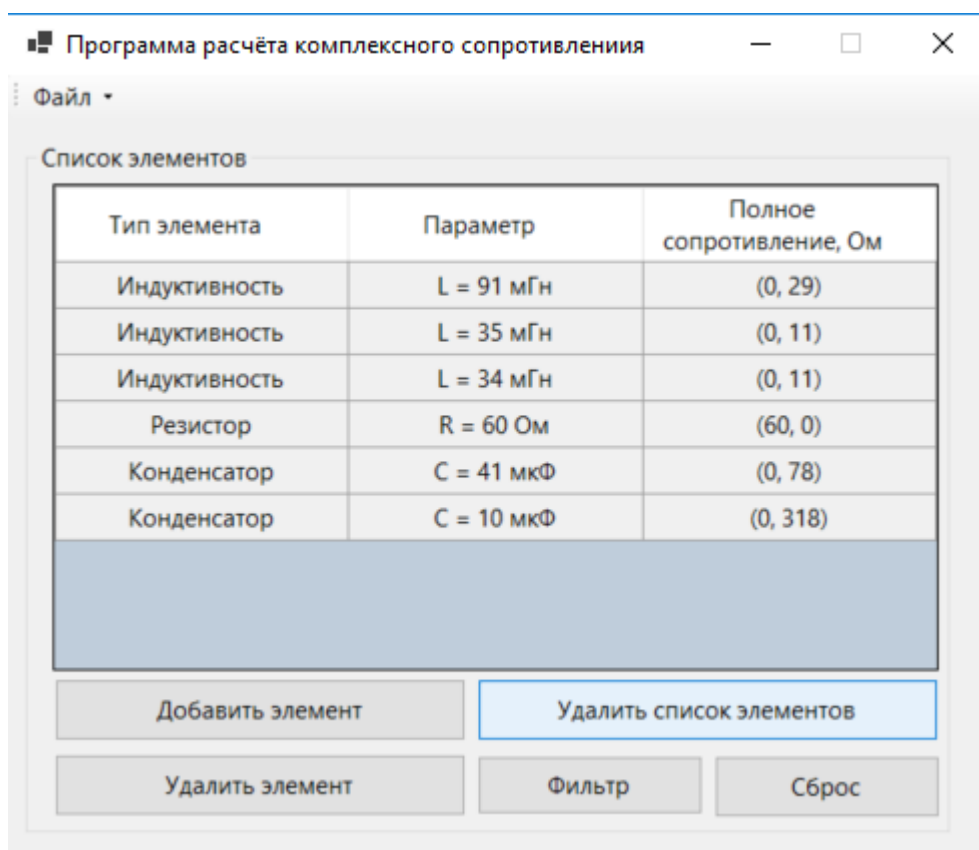


Рисунок 27. Результат загрузки данных

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

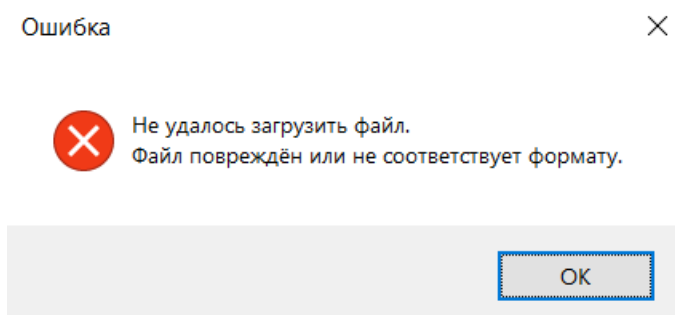


Рисунок 28. Загрузка повреждённого файла

## **ПРИЛОЖЕНИЕ А**

**Техническое задание на создание автоматизированной системы**  
**Программа для расчёта комплексного сопротивления пассивных элементов**  
**электрической цепи**

Разработчик: студент гр. О-5КМ11 НИ ТПУ Копцев И.О.

Заказчик: к.т.н., доцент каф. КСУП ТУСУР Калентьев А.А.

Томск 2023

## **1. Общие сведения**

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

Полное наименование: «Программа для расчёта комплексного сопротивления пассивных элементов электрической цепи».

Условное обозначение: Система.

### **1.2. Наименование предприятий разработчика и заказчика системы**

Заказчик: к.т.н., доцент каф. КСУП ТУСУР Калентьев А.А.

Разработчик: студент гр. О-5KM11 НИ ТПУ Копцев И.О.

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

Начало работ: 17 мая 2023 г.

Окончание работ: 04 ноября 2023 г.

## **2. Назначение и цели создания системы**

### **2.1. Назначение системы**

Система предназначена для автоматизации расчёта комплексного сопротивления пассивных элементов электрической цепи.

Пользователями Системы являются специалисты инженерных отраслей, студенты и школьники.

### **2.2. Цели создания системы**

Система создаётся в целях сокращения трудозатрат человека при расчётах комплексного сопротивления пассивных элементов электрической цепи.

### **3. Характеристика объектов автоматизации**

Вычислять комплексное сопротивление пассивных элементов приходится множеству специалистов инженерных отраслей, студентам энергетикам, радиоэлектроникам и школьникам на уроках физики или математики. Например, при выполнении расчётно-графического задания по дисциплине «Теоретические основы электротехники» требуется определить сопротивление более ста элементов электрической цепи в соответствии с заданием. Поэтому представляется целесообразным автоматизировать этот процесс. Задача усложняется в случаях, когда схема состоит из разных элементов, сопротивление которых рассчитываются по различным выражениям, ввиду их физических особенностей.

В основу Системы положена следующая реализация: пользователь выбирает из предложенного ему списка (резистивный элемент, катушка индуктивности и конденсатор) элемент и вводит характеризующий его параметр (для резистора – активное сопротивление, для катушки – индуктивность, для конденсатора – ёмкость). По заложенным в Системе методам и принятым на вход параметрам определяется комплексное сопротивление для соответствующего элемента и результат выводится пользователю.

#### 4. Требования к системе

Таблица 4.1 – Префиксы мнемонических идентификаторов требований и их расшифровка

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

##### 4.1 Требования к архитектуре

**A01.** Система должна быть реализована в виде настольного приложения.

##### 4.2 Требования к структуре данных

**D01.** Данные о элементах электрических схем должны храниться в -файле с расширением \*.elm.

##### 4.3 Функциональные требования

**F01.** Система должна рассчитывать полное сопротивление для следующих пассивных элементов:

- резистивный элемент;
- катушка индуктивности;
- конденсатор.

**F01.01.** Полное сопротивление резистивного элемента должно определяться по выражению:

$$\underline{Z} = R + j0,$$

где  $R$  – активное сопротивление, Ом.

**F01.02.** Полное сопротивление катушки индуктивности должно определяться по выражению:

$$\underline{Z} = 0 + j\omega L,$$

где  $\omega$  – циклическая частота, рад/с;  $L$  – индуктивность, мГн.

**F01.03.** Полное сопротивление конденсатора должно определяться по выражению:

$$\underline{Z} = 0 + j \frac{1}{\omega C},$$

где  $\omega$  – циклическая частота, рад/с;  $C$  – ёмкость, мкФ.

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

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

- тип элемента;
- параметр в соответствии с типом элемента.

**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.

Работоспособность на других выпусках и версиях не гарантируется.

**C02.** На рабочей станции должен быть установлен .NET Framework версии 6.0 или старше.

#### **4.6 Требования к аппаратному обеспечению**

**C03.** Процессор – не менее 2500 МГц.

**C04.** ОЗУ – не менее 2 ГБ.