Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра компьютерных систем в управлении и проектировании (КСУП)

РАЗРАБОТКА ПРОЕКТНОЙ ДОКУМЕНТАЦИИ

Отчет

по дисциплине «Групповое Проектное Обучение»

Студент гр. 586-2

\_\_\_\_\_\_\_\_\_\_В.М. Козырева

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2018 г.

Руководитель

К.т.н., доцент каф. КСУП

\_\_\_\_\_\_\_\_\_\_\_ А.Е. Горяинов

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2018 г.

Томск 2018

Оглавление

[**Техническое задание** 3](#_Toc517281057)

[**Введение** 4](#_Toc517281058)

[**Основная часть** 5](#_Toc517281059)

[**UML – диаграммы** 5](#_Toc517281060)

[3.1.1 UML – диаграмма вариантов использования 5](#_Toc517281061)

[3.1.2 UML – диаграмма классов 5](#_Toc517281062)

[**Дерево ветвления Git** 8](#_Toc517281063)

[**Тестирование программы** 9](#_Toc517281064)

[3.3.1 Функциональное тестирование 9](#_Toc517281065)

[3.3.2 Модульное тестирование 10](#_Toc517281066)

[**Заключение** 11](#_Toc517281067)

[**Список использованной литературы** 12](#_Toc517281068)

# **Техническое задание**

Задача:

Разработка программы, которая должна рассчитывать комплексное сопротивление пассивных элементов электрической цепи.

Требования к программе:

* Программа должна представлять собой исполняемый файл для Windows;
* Иметь графический интерфейс пользователя;
* Отображение данных в виде таблицы;
* Возможность добавлять разные типы элементов в таблицу и модифицировать заданные для них значения;
* Проверка введенных данных на всех этапах работы программы на корректность;
* Возможность сохранять и загружать данные;

# **Введение**

Целью данной работы является разработка программного продукта с использованием новых технологий в программировании. Изучение языка C#, системы контроля версий, платформы .NET, стандартов построения UML – диаграмм классов и UML – диаграмм вариантов использования, фреймворка NUnit.

Программа предназначена для хранения списка пассивных элементов электрической цепи. В программе реализована возможность изменения параметров элементов, также сохранения данных в файл и загрузка данных из файла.

# **Основная часть**

## **UML – диаграммы**

### 3.1.1 UML – диаграмма вариантов использования

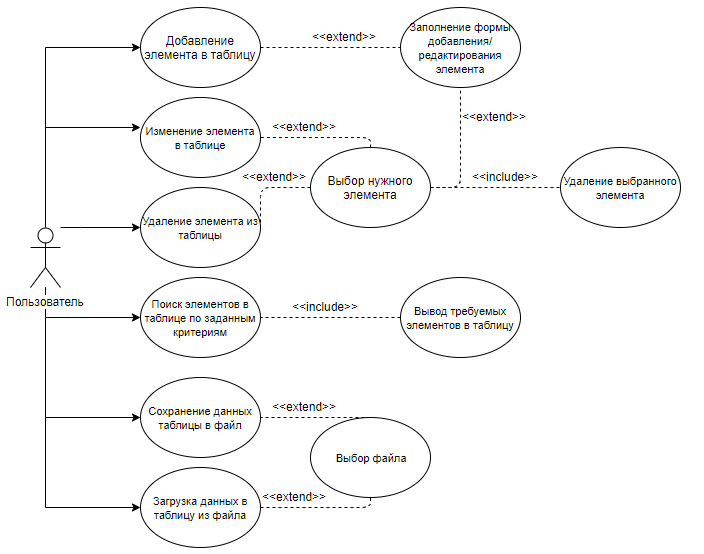


Рисунок 3.1 – UML – диаграмма вариантов использования

### 3.1.2 UML – диаграмма классов

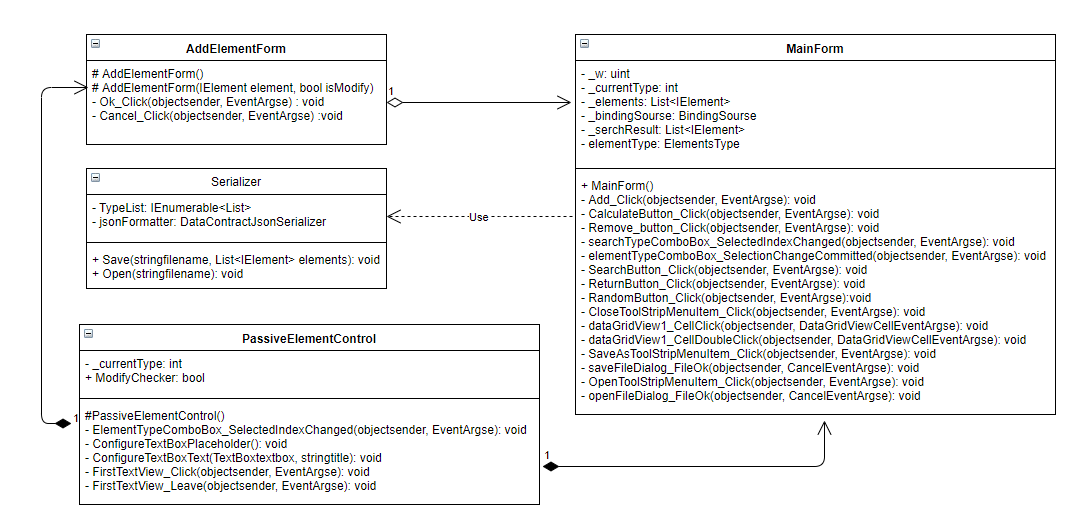


Рисунок 3.2 – UML-диаграмма классов пользовательского интерфейса

В представленных таблицах 3.1-3.4 описаны классы, образующие связь типа «общее-частное».

Таблица 3.1 – Описание интерфейса *IElement*

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Описание интерфейса | | |
| Интерфейс *IElement* – сущность для описания абстрактного пассивного элемента в программе | | |
| Свойства | | |
| + Parameter | double | Получение параметров дочерних классов |
| Методы | | |
| + GetImpedance | Complex | Комплексное сопротивление |

Таблица 3.2 – Описание интерфейса *Inductor*

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Описание класса | | |
| Класс *Inductor* – сущность для описания катушки индуктивности в программе | | |
| Свойства | | |
| + Parameter | double | Получение параметров дочерних классов |
| Методы | | |
| + GetResistance | Complex | Комплексное сопротивление |
| + ToString | String | Переопределение ToString для вывода типа элемента |
| # Inductor (double) |  | Конструктор для создания новой катушки индуктивности с помощью значения индуктивности. |

Таблица 3.3 – Описание интерфейса *Capacitor*

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Описание класса | | |
| Класс *Capacitor* – сущность для описания конденсатора в программе | | |
| Свойства | | |
| + Parameter | double | Получение параметров дочерних классов |
| Методы | | |
| + GetImpedance | Complex | Комплексное сопротивление |
| + ToString | String | Переопределение ToString для вывода типа элемента |
| # Capacitor (double) |  | Конструктор для создания нового конденсатора с помощью значения емкости. |

Таблица 3.4 – Описание интерфейса *Resistor*

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Описание класса | | |
| Класс *Resistor* – сущность для описания резистора в программе | | |
| Свойства | | |
| + Parameter | double | Получение параметров дочерних классов |
| Методы | | |
| + GetImpedance | Complex | Комплексное сопротивление |
| + ToString | String | Переопределение ToString для вывода типа элемента |
| # Resistor (double) |  | Конструктор для создания нового резистора с помощью значения сопротивления. |

В таблице 3.5 содержится описание классов пользовательского интерфейса.

Таблица 3.5 – Описание классов пользовательского интерфейса

|  |  |
| --- | --- |
| **Название** | **Описание** |
| MainForm | Главная форма |
| AddElementForm | Форма добавления и редактирования элементов. Содержит PassiveElementControl. |
| PassiveElementControl | Пользовательский элемент управления. Предоставляет функции добавления и редактирования существующих элементов, а также отображения расширенных данных о параметрах выбранного элемента в таблице MainForm. |

## **Дерево ветвления Git**

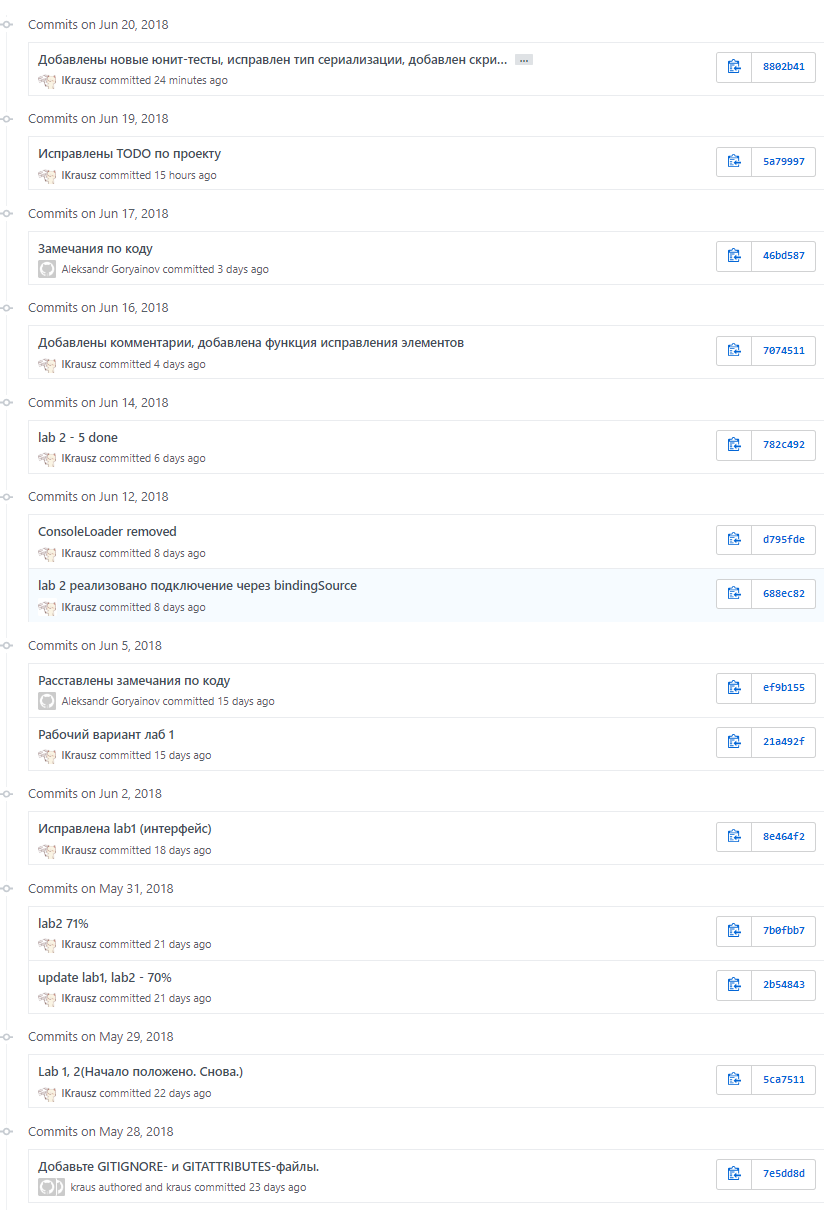


Рисунок 3.3 – Дерево ветвлений Git

## **Тестирование программы**

Тестирование программы является важным этапом разработки программы, поскольку позволяет разработчику гарантировать пользователям корректную работу конечного продукта.

### 3.3.1 Функциональное тестирование

Функциональное тестирование проводилось на протяжении всей разработки программы. В таблице 2.6 представлены тестовые случаи на разных стадиях разработки и результаты работы программы.

Таблица 3.6 – Тестовые случаи и результаты работы программы

|  |  |
| --- | --- |
| **Тестовый случай** | **Результат** |
| Попытка добавления элемента при корректных данных | В таблицу добавляется элемент с заданными параметрами и список элементов |
| Попытка добавления элемента при некорректных данных | Выводится сообщение о некорректном вводе данных |
| Попытка удаления выбранного элемента из таблицы | Элемент удаляется из таблицы и списка элементов |
| Попытка расчета комплексного сопротивления | В таблице рядом с каждым элементом появляются данные о его комплексном сопротивлении |
| Попытка модификации выбранного элемента в таблице при корректных значениях | Значение или тип элемента меняется в соответствии с введенными данными |
| Попытки модификации выбранного элемента в таблице при некорректных значениях | Выводится сообщение о некорректном вводе данных |
| Сохранение данных в файл | Данные сохраняются в файл с введенным именем по указанному пути |
| Открытие файла с данными | В таблице появляются элементы, которые были сохранены прежде |

### 3.3.2 Модульное тестирование

Модульное тестирование представляет собой тестирование отдельных компонентов программы на уровне исходного кода.

Результаты прохождения модульных тестов представлены на рисунке ниже.

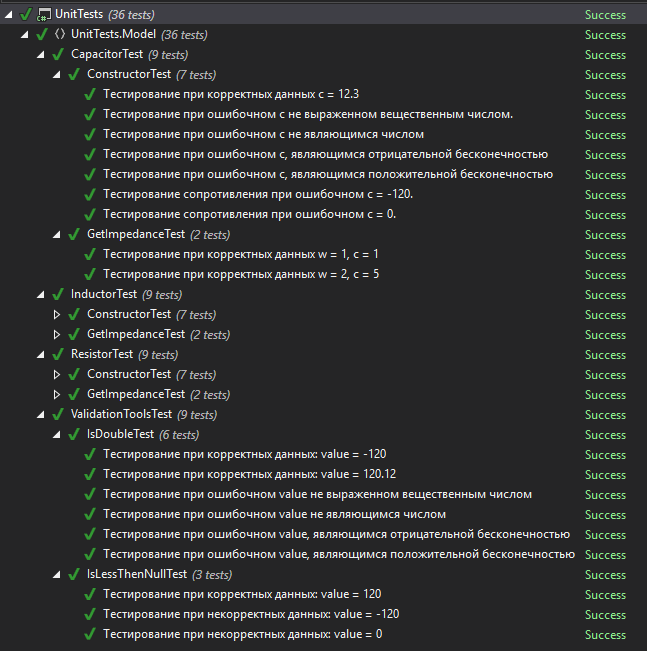
****

Рисунок 3.4 – Результаты модульного тестирования

# **Заключение**

В ходе работы была разработана программа для расчета комплексного сопротивления пассивных элементов.

Был изучен синтаксис языка C#, на котором велась разработка, также были получены навыки создания пользовательского интерфейса при помощи компонентов WindowsForms.

Для гарантии качественной работы разработанной программы было проведено тестирование продукта с использованием фреймворка для модульного тестирования «NUnit».

Также в ходе работы были получены навыки работы с системой контроля версий Git.

После завершения разработки программного продукта была написана техническая документация, которая включала в себя UML – диаграммы для бизнес-логики и для пользовательского интерфейса, описания классов и процесса тестирования, а также дерево ветвления системы контроля версий.

# **Список использованной литературы**

1. Калентьев А. А. Новые технологии в программировании : учебное пособие / А. А. Калентьев, Д. В. Гарайс, А. Е. Горяинов. — Томск : Эль Контент, 2014.
2. MSDN Library [Электронный ресурс]. URL: https://msdn.microsoft.com (Дата обращения 20.06.18)
3. NUnit.org [Электронный ресурс]. URL: nunit.org (Дата обращения 20.06.18)
4. Соглашения по оформлению кода команды RSDN [Электронный ресурс]. URL: https://rsdn.org/article/mag/200401/codestyle.XML#EODAE (Дата обращения 20.06.18)