|  |
| --- |
| **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 №6**

по дисциплине: «Основы объектно-ориентированного программирования»



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Выполнил студент гр. О-5КМ91 | |  | \_\_\_\_\_\_\_\_\_\_ | |  | Жуйков А. К. |
|  | |  |  | |  |  |
|  | |  |  | 18 июня 2021 г. | | |
|  | |  |  | |  |  |
| Отчёт принял | доцент, к.т.н. |  | \_\_\_\_\_\_\_\_\_\_ | |  | Калентьев А. А. |
|  |  |  |  | |  |  |
|  |  |  |  | \_\_\_ \_\_\_\_\_\_ 2021 г. | | |

Томск 2021

# Содержание

[Введение 3](#_Toc74888895)

[1 Техническое задание 4](#_Toc74888896)

[1.1 Предпроектный анализ 4](#_Toc74888897)

[2 Описание программной системы 5](#_Toc74888898)

[2.1 Диаграмма вариантов использования 5](#_Toc74888899)

[2.2 UML диаграмма классов 5](#_Toc74888900)

[2.3 Описание классов, образующих связь типа «общее-частное» 7](#_Toc74888901)

[2.4 Дерево ветвлений Git 9](#_Toc74888902)

[2.5 Тестирование программы 10](#_Toc74888903)

[2.5.1 Тестовый случай «Добавить» 10](#_Toc74888904)

[2.5.2 Тестовый случай «Удалить» 12](#_Toc74888905)

[2.5.3 Тестовый случай «Найти» 13](#_Toc74888906)

[2.5.4 Тестовый случай «Сохранить данные» 14](#_Toc74888907)

[2.5.5 Тестовый случай «Загрузить данные» 16](#_Toc74888908)

[Заключение 19](#_Toc74888909)

[Список использованных источников 20](#_Toc74888910)

[Приложение А Техническое задание на создание автоматизированной   
системы 21](#_Toc74888911)

**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. Разработка технического задания (далее – ТЗ) на разработанную программу (ТЗ приведено в Приложении А);
2. Формирование диаграммы вариантов использования для разработанной программы;
3. Формирование UML-диаграммы классов;
4. Описание классов, образующих связь типа «общее-частное»;
5. Представление дерева ветвлений Git;
6. Проведение тестирования программы.

**1 Техническое задание**

**1.1 Предпроектный анализ**

Проблема предметной области заключается в том, что у заказчика отсутствует удобная программа для определения объёма затраченного топлива различными типами транспорта, имеющимися у заказчика.

**Цель:** разработать программу для вычисления объёма затраченного топлива различными типами транспорта.

Задачи:

* Программа должна предоставлять возможность расчёта объёма затраченного топлива для следующих типов транспорта: машина, машина-гибрид (далее – гибрид), вертолёт
* Программа должна предоставлять возможность отображения транспорта и затраченного топлива.
* Программа должна иметь возможность сохранять список транспорта, для которого производится расчёт затраченного объёма топлива в специальном формате.
* Программа должна иметь возможность редактировать состав списка исследуемых единиц транспорта.

Техническое задание приведено в приложении А.

**2 Описание программной системы**

**2.1 Диаграмма вариантов использования**

Диаграмма вариантов использования разработанной программы представлена на рисунке 1.

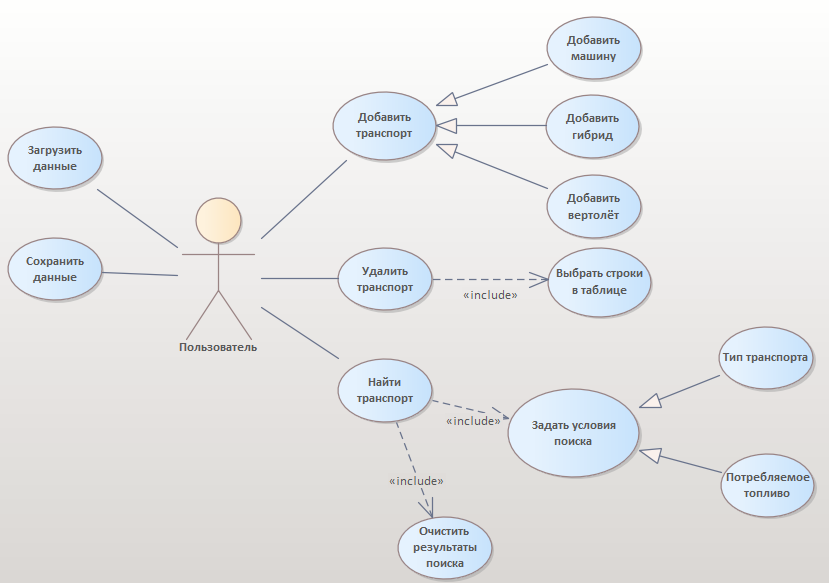


Рисунок 1 – Диаграмма вариантов использования

**2.2 UML диаграмма классов**

Диаграмма классов приведена на рисунке 2.

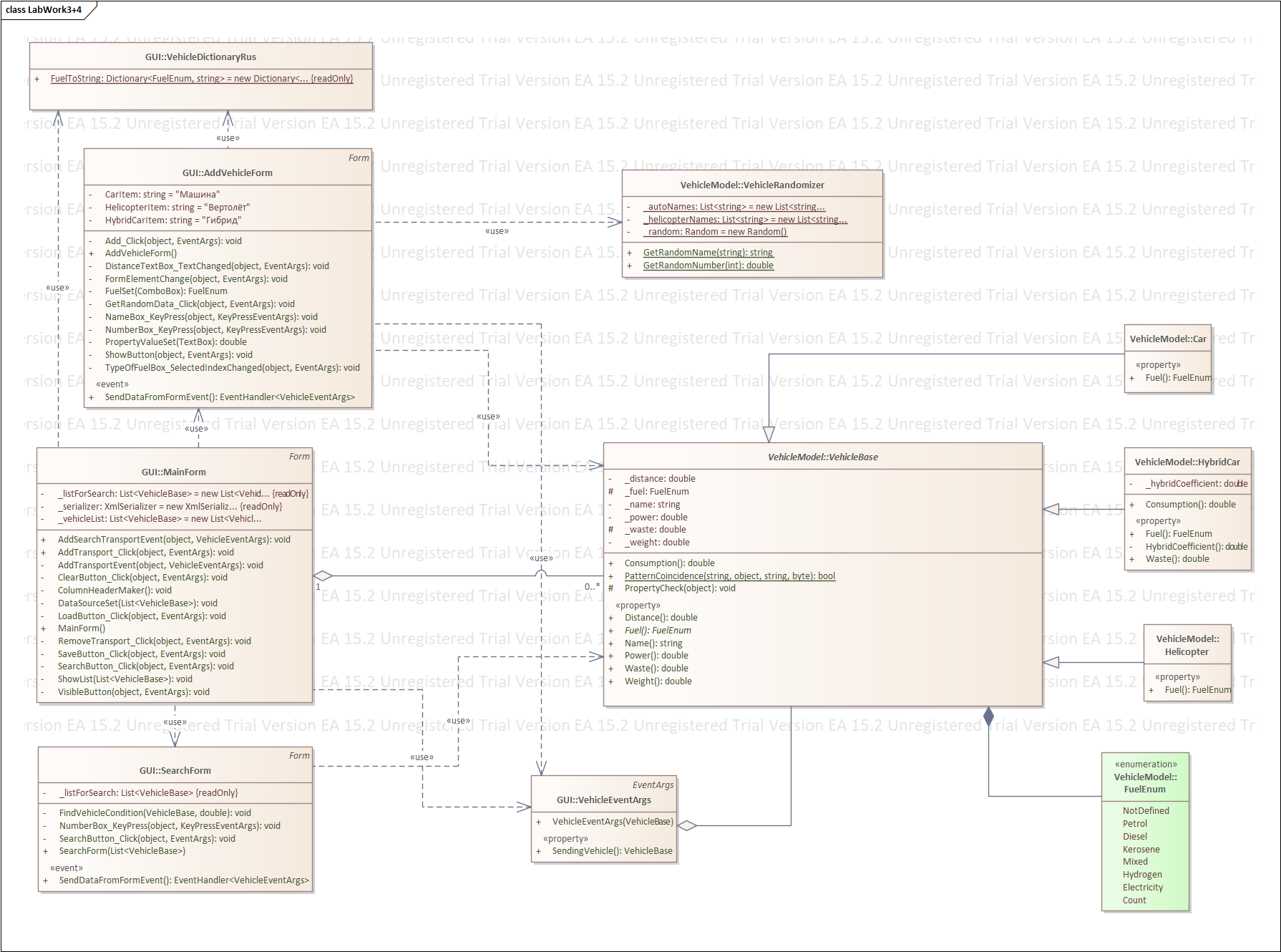


Рисунок 2 – UML-диаграмма классов

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

В таблице 1 представлено описание абстрактного класса VehicleBase с его полями, свойствами и методами.

Таблица 1 – Описание класса VehicleBase

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Описание класса | | |
| Класс VehicleBase – абстрактный транспорт | | |
| Поля | | |
| – \_distance | double | Пройденная транспортом дистанция, км |
| # \_fuel | FuelEnum | Тип топлива |
| – \_name | string | Имя транспорта |
| – \_power | double | Мощность транспорта, л. с. |
| # \_waste | double | Расход топлива, л/км |
| – \_weight | double | Вес транспорта, т |
| Словари | | |
| + FuelToString | <FuelEnum, string> | Словарь соответствия типа топлива его строковому названию |
| Свойства | | |
| + Distance | double | Пройденная транспортом дистанция, км |
| + *Fuel* | FuelEnum | Тип топлива.  Абстрактное свойство, переопределяется в производных классах |
| + Name | string | Имя транспорта |
| + Power | double | Мощность транспорта, л. с. |
| + Waste | double | Расход топлива, л/км |
| + Weight | double | Вес транспорта, т |
| Методы | | |
| + Consumption() | double | Определяет объём потребления топлива транспортом, л |
| + PatternCoincidence(string, object, string, byte) | bool | Определяет совпадение введённой строки с шаблоном:  string – введённая строка  object – нажатая клавиша  string – шаблон  byte – максимальная длина строки |
| # Property-Check  (object) | void | Проверяет величину, поступающую  в поле:  object – проверяемая величина |

В таблицах 2–4 приведены описания классов Car, HybridCar и Helicopter, которые наследуются от VehicleBase.

Таблица 2 – Описание класса Car

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Описание класса | | |
| Класс Car – машина | | |
| Свойства | | |
| + Fuel | FuelEnum | Тип топлива (бензин/дизель) |

Таблица 3 – Описание класса HybridCar

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Описание класса | | |
| Класс HybridCar – гибрид | | |
| Поля | | |
| – \_hybridCoefficient | double | Коэффициент гибридности,  снижающий потребление топлива  гибридом |
| Свойства | | |
| + Fuel | FuelEnum | Тип топлива  (водород/электричество/ /смешанное топливо) |
| – HybridCoefficient | double | Коэффициент гибридности,  снижающий потребление топлива  гибридом |
| + Waste | double | Расход топлива, л/км |
| Методы | | |
| + Consumption() |  | Определяет объём потребляемого  топлива с учётом коэффициента  гибридности |

Таблица 4 – Описание класса Helicopter

|  |  |  |  |
| --- | --- | --- | --- |
| Название | | Тип | Описание |
| Описание класса | | | |
| Класс Helicopter – вертолёт | | | |
| Свойства | | | |
| + Fuel | FuelEnum | | Тип топлива (керосин) |

**2.4 Дерево ветвлений Git**

На рисунке 3 представлено дерево ветвлений Git, полученное после окончания работы над программой.

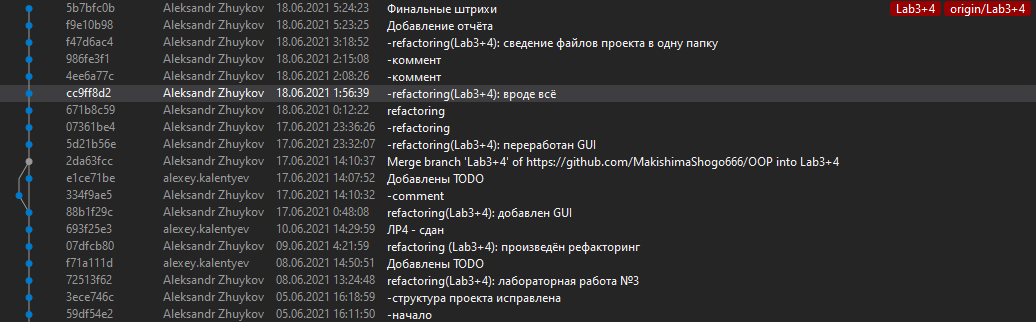


Рисунок 3 – Дерево ветвлений Git

**2.5 Тестирование программы**

В данном разделе представлены результаты функционального тестирования разработанной программы.

Графический интерфейс пользователя представлен на рисунке 4.

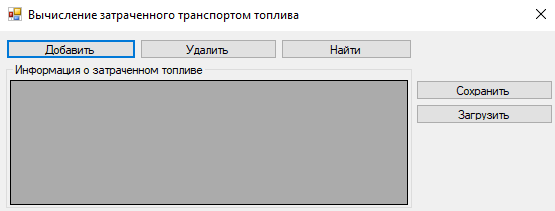


Рисунок 4 – Графический интерфейс пользователя

### 2.5.1 Тестовый случай «Добавить»

Для добавления транспорта в таблицу на рисунке 4 необходимо вызвать соответствующую форму путём нажатия кнопки «Добавить» (рисунок 5).

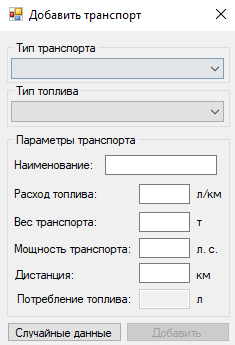


Рисунок 5 – Форма для добавления транспорта

Внести транспорт в таблицу можно только после заполнения всех полей, кнопка «Добавить» станет активной, и её можно будет нажать (рисунки 6–7).

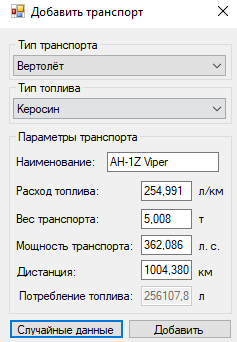


Рисунок 6 – Заполнение полей

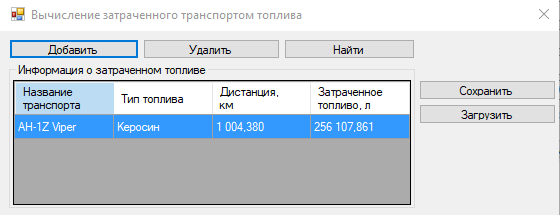


Рисунок 7 – Успешное добавление нового транспорта

В программе предусмотрена система обработки некорректного ввода данных пользователем таким образом, что вероятность такого ввода минимизирована. Например, нельзя ввести текстовое значения в поля ввода чисел, нельзя ввести отрицательное значение.

### 2.5.2 Тестовый случай «Удалить»

Для удаления транспорта необходимо выделить соответствующую строку (строки) в таблице и нажать на кнопку «Удалить» (рисунки 8–11).

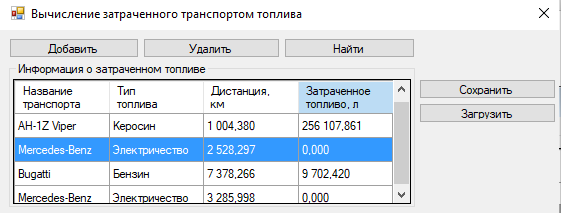


Рисунок 8 – Выбор транспорта в таблице для удаления

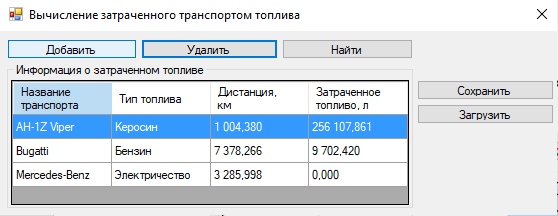


Рисунок 9 – Результат нажатия кнопки «Удалить»

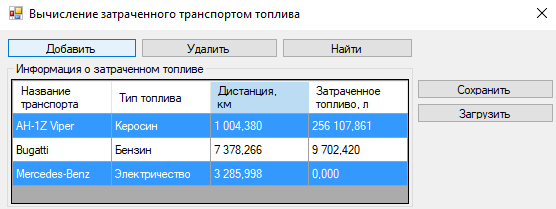


Рисунок 10 – Выбор нескольких единица транспорта для удаления

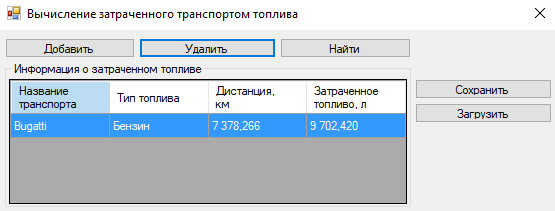


Рисунок 11 – Результат удаления выбранных единиц транспорта

### 2.5.3 Тестовый случай «Найти»

Для поиска единиц транспорта, удовлетворяющим определённым условиям в списке необходимо нажать кнопку «Найти» (рисунок 12). Откроется соответствующая форма для поиска (рисунок 13).

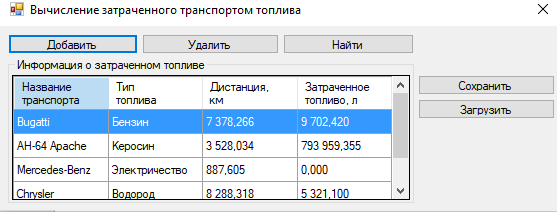


Рисунок 12 – Вызов формы для поиска транспорта

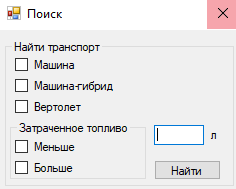


Рисунок 13 – Форма для поиска

Далее пользователь выбирает условия, которым должен удовлетворять искомый транспорт, устанавливаются соответствующие галочки и заполняется поле объёма затраченного топлива (рисунки 14–15).

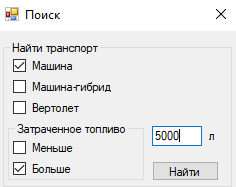


Рисунок 14 – Поиск транспорта, являющегося машиной с объёмом потребления более 5000 л

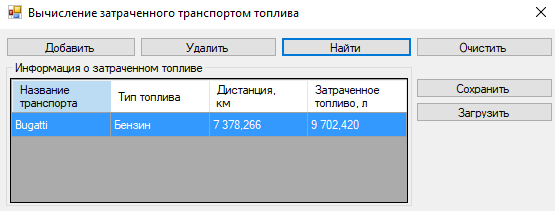


Рисунок 15 – Результат поиска элемента

Для сброса результатов поиска предусмотрена соответствующая кнопка «Очистить» на рисунке 15.

### 2.5.4 Тестовый случай «Сохранить данные»

Для сохранения данных в таблице необходимо нажать на соответствующую кнопку на форме (рисунок 16). Откроется диалоговое окно сохранения файла, где пользователь выбирает директорию и указывает имя файла (рисунок 17).

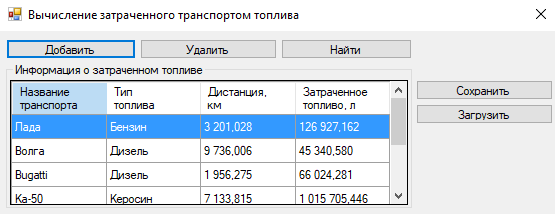


Рисунок 16 – Сохранение данных

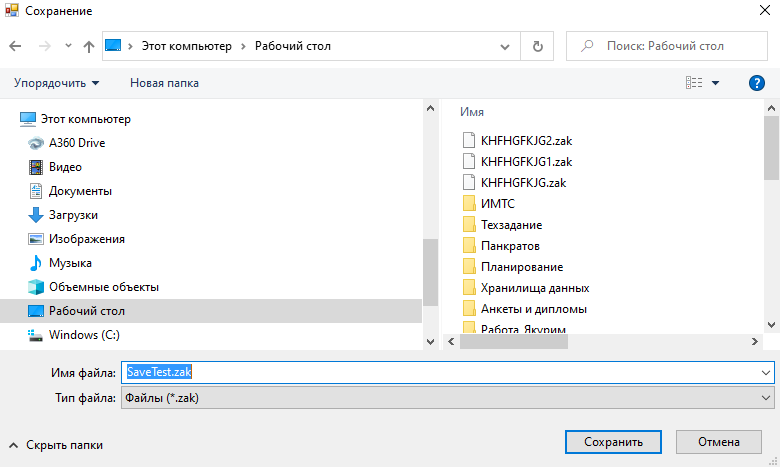


Рисунок 17 – Диалоговое окно сохранения файла

После сохранения данных в файл появится соответствующее сообщение (рисунки 18 и 19).

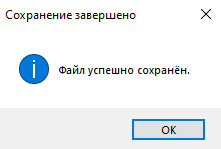


Рисунок 18 – Сообщение о сохранении файла

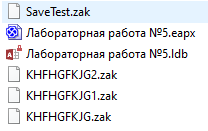


Рисунок 19 – Результат сохранения файла

В случае, если таблица пуста, сохранение не производится (рисунок 20).

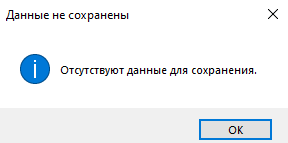


Рисунок 20 – Результат нажатия на кнопку «Сохранить данные»   
при пустой таблице

### 2.5.5 Тестовый случай «Загрузить данные»

Для загрузки данных в таблицу необходимо нажать на соответствующую кнопку (рисунок 21).

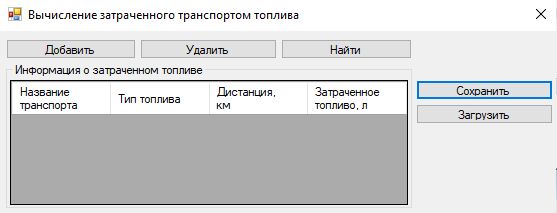


Рисунок 21 – Загрузить данные в таблицу

Далее откроется системный диалог загрузки файла (рисунок 22). После успешной загрузки появится сообщение (рисунки 23 и 24).

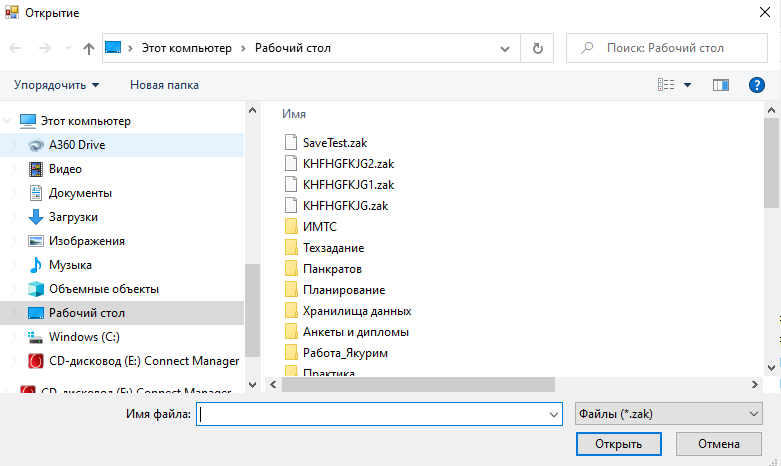


Рисунок 22 – Выбор файла для загрузки

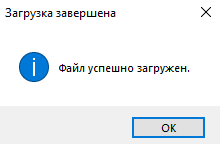


Рисунок 23 – Сообщение, информирующее об успешной загрузке файла



Рисунок 24 – Результат загрузки данных

В случае, если структура файла не соответствует установленному формату, либо имеются повреждённые данные, появится соответствующее сообщение (рисунок 25).

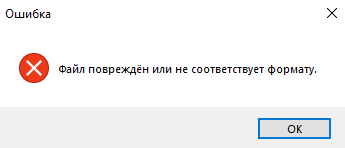


Рисунок 25 – Загрузка повреждённого файла

# Заключение

В результате выполнения лабораторной работы были выполнены следующие задачи:

1. разработано техническое задание;
2. составлена диаграмма вариантов использования;
3. составлена UML диаграмма классов;
4. для классов, образующих связь типа «общее-частное» (наследование, реализация), приведено описание в виде таблиц;
5. приведено дерево ветвления Git;
6. произведено тестирование 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 объёма потребления топлива   
различных типов транспорта

Разработчик: студент гр. О-5КМ91 НИ ТПУ Жуйков А. К.

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 объёма потребления топлива различных типов транспорта».

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 Жуйков А. К.

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

Начало работ: 20 мая 2021 г.

Окончание работ: 18 июня 2021 г.

**2 Назначение и цели создания системы**

**2.1 Назначение системы**

Система предназначена для расчёта объёма потребления затраченного топлива для различных типов транспорта: машина, гибрид, вертолёт.

**2.2 Цели создания системы**

Система создаётся в целях сокращения трудозатрат специалистов предприятия заказчика при расчётах объёмов затраченного топлива различными типами транспорта.

**3 Характеристика объектов автоматизации**

Расчёт объёма затраченного топлива является важной задачей при планировании хозяйственной деятельности и закупок транспортных предприятий. Так как подобные расчёты выполняются специалистами довольно часто, автоматизация указанного процесса является актуальной задачей.

**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 загружаться и сохраняться в XML-файле с расширением \*.zak.

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

<? xml version = "1.0" encoding = "utf-8" ?>

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

< xsd:element name = "ArrayOfVehicleBase" >

< xsd:complexType >

< xsd:sequence >

< xsd:element maxOccurs = "unbounded" name = "VehicleBase" >

< xsd:complexType >

< xsd:sequence >

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

< xsd:element name = "Fuel" type = "xsd:string" />

< xsd:element name = "Weight" type = "xsd:decimal" />

< xsd:element name = "Waste" type = "xsd:decimal" />

< xsd:element name = "Power" type = "xsd:decimal" />

< xsd:element name = "Distance" type = "xsd:decimal" />

</ xsd:sequence >

</ xsd:complexType >

</ xsd:element >

</ xsd:sequence >

</ xsd:complexType >

</ xsd:element >

</ xs:schema >

**4.3 Функциональные требования**

**F01.** Система должна рассчитывать объём потребления топлива для следующих типов транспорта:

* машина;
* гибрид;
* вертолёт.

**F01.01.** Расчёт объёма потребления топлива для машины и вертолёта выполняется по следующей формуле:



где *W* – расход топлива, л/км;

*s* – дистанция, км.

**F01.02.** Расчёт объёма потребления топлива для гибрида вычисляется по следующей формуле:



где *k* < 1– коэффициент гибридности;

**F02.** В системе должен быть реализован список типов транспорта.

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

* имя транспорта;
* тип топлива;
* расход топлива;
* вес транспорта;
* мощность транспорта;
* пройденная транспортом дистанция.

**F03.** В системе должна присутствовать функция добавления единиц транспорта в список.

**F04.** В системе должна присутствовать функция удаления единиц транспорта из списка.

**F05.** В системе должна присутствовать функция поиска элементов по типу транспорта и объёму затраченного топлива.

**F06.** В системе должна присутствовать функция сохранения единиц транспорта в файл (**D01**).

**F07.** В системе должна присутствовать функция загрузки списка единиц транспорта 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 Home 21H1. 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.