**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

федеральное государственное автономное образовательное учреждение высшего образования

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**

**ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

Инженерная школа энергетики

Отделение электроэнергетики и электротехники

Направление: 09.04.03 Прикладная информатика

Отчет по лабораторной работе №5

**«Проектная документация»**

По дисциплине:

Основы объектно-ориентированного программирования



|  |  |  |
| --- | --- | --- |
| Выполнил: студент гр. О-5КМ21 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | Краснобаев А.Д. |
| Проверил: доцент, к.т.н. | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись)  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (дата) | Калентьев А.А. |
|  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (балл) |  |

Томск - 2024

**СОДЕРЖАНИЕ**

[Введение 3](#_Toc179995326)

[1 UML диаграмма вариантов использования 4](#_Toc179995327)

[2 UML диаграмма классов 5](#_Toc179995328)

[3 Дерево ветвлений Git 8](#_Toc179995329)

[4 Тестирование программы 9](#_Toc179995330)

[4.1 Тестовый случай «Добавить движение» 9](#_Toc179995331)

[4.2 Тестовый случай «Удалить» 12](#_Toc179995332)

[4.3 Тестовый случай «Очистить» 13](#_Toc179995333)

[4.4 Тестовый случай «Отфильтровать список» 14](#_Toc179995334)

[4.5 Тестовый случай «Сброс фильтра» 20](#_Toc179995335)

[4.6 Тестовый случай «Сохранить файл» 21](#_Toc179995336)

[4.7 Тестовый случай «Загрузить файл» 22](#_Toc179995337)

[Список источников 24](#_Toc179995338)

[ПРИЛОЖЕНИЕ А 25](#_Toc179995339)

**Введение**

Назначение программной документации заключается в детальном описании всех аспектов разрабатываемого программного обеспечения. В контексте данной лабораторной работы, документация играет ключевую роль, предоставляя структурированную информацию о разработке программы, предназначенной для вычисления координат для различных типов движений: равномерное, равноускоренное и колебательное. Эта документация предназначена для облегчения понимания, сопровождения и дальнейшего развития программы.

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

Для достижения поставленной цели должны быть выполнены следующие задачи:

* Составление UML диаграммы вариантов использования для разработанной программы;
* Составление UML диаграммы классов;
* Описание классов, образующих связь типа «общее-частное»;
* Дерево ветвлений Git;
* Тестирование программы;
* Техническое задание.

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].

Диаграмма вариантов использования для разработанного ПО приведена на рисунке 1.

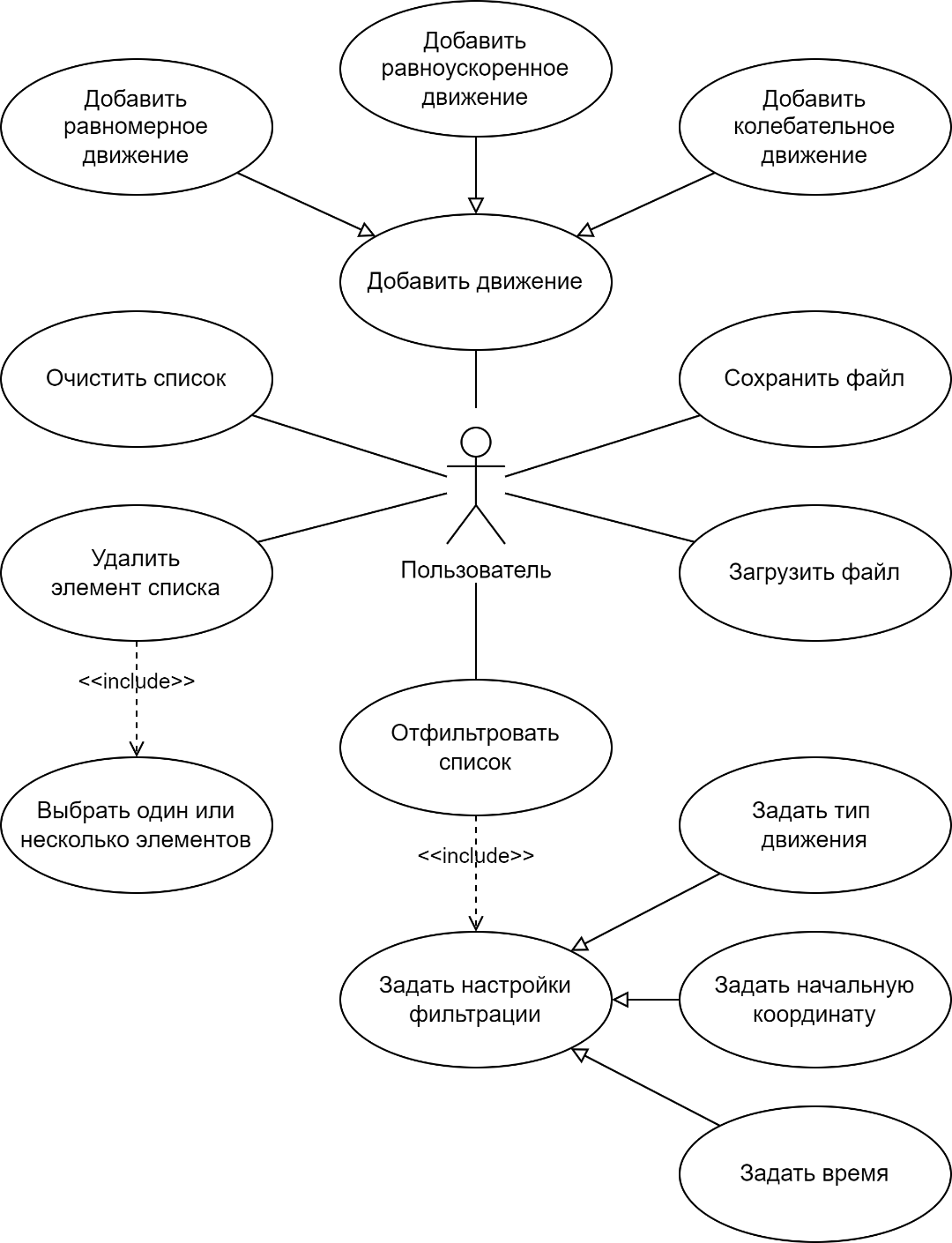


Рисунок 1 – Диаграмма вариантов использования

1. **UML диаграмма классов**

Диаграмма классов — один из видов UML-диаграмм, позволяющий описать статический аспект программной системы за счёт описания классов и их взаимосвязей в системе [1].

Диаграмма классов приведена на рисунке 2.

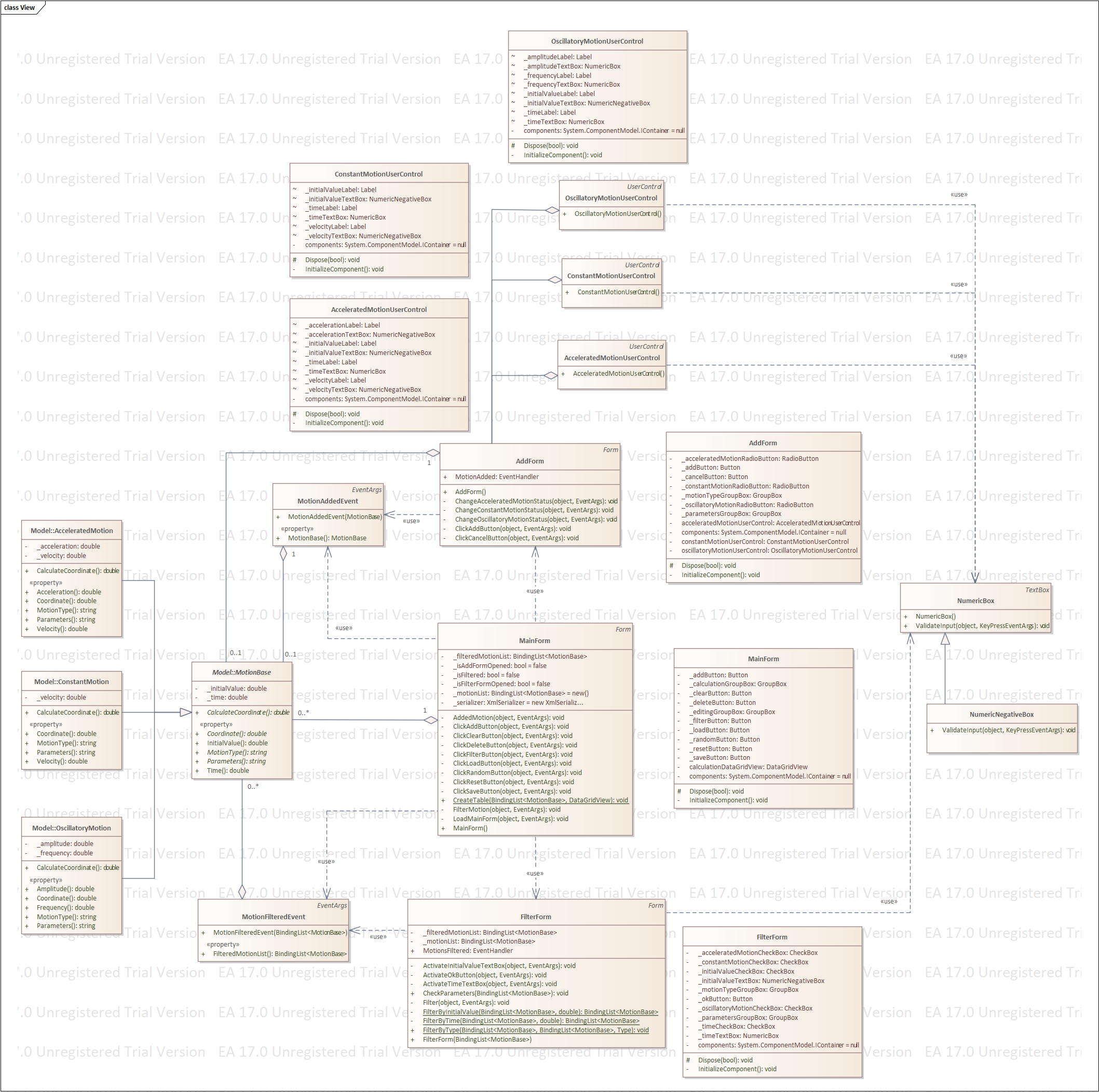


Рисунок 2 – UML диаграмма классов

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

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

Таблица 1 – Описание класса MotionBase

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название | Тип | | | Описание |
| Описание класса | | | | |
| Класс *MotionBase* – абстрактный базовый класс, описывает общую природу движения | | | | |
| Поля | | | | |
| - \_initialValue | | double | Начальная координата | |
| - \_time | | double | Время | |
| Свойства | | | | |
| + Coordinate | double | | | Координата.  Абстрактное свойство, переопределяется в производных классах. |
| + InitialValue | double | | | Начальная координата.  Общее свойство для всех производных классов. |
| + MotionType | string | | | Тип движения.  Абстрактное свойство, переопределяется в производных классах. |
| + Parameters | string | | | Параметры движения  Абстрактное свойство, переопределяется в производных классах. |
| + Time | double | | | Время.  Общее свойство для всех производных классов. |
| Методы | | | | |
| + CalculateCoordinate | double | | | Расчет координаты.  Абстрактный метод, переопределяется в производных классах. |

В таблицах 2–4 приведены описания классов *ConstantMotion*, *AcceleratedMotion* и *OscillatoryMotion*, которые наследуются от *MotionBase*.

Таблица 2 – Описание класса ConstantMotion

| Название | Тип | Описание |
| --- | --- | --- |
| Описание класса | | |
| Класс *ConstantMotion* – равномерное движение | | |
| Поля | | |
| – \_velocity | double | Скорость |
| Свойства | | |
| + Coordinate | double | Координата |
| + MotionType | string | Тип движения (равномерное) |
| + Parameters | string | Параметры движения (скорость) |
| + Velocity | double | Скорость |
| Методы | | |
| + CalculateCoordinate | double | Расчет координаты для равномерного движения |

Таблица 3 – Описание класса AcceleratedMotion

| Название | | Тип | | Описание |
| --- | --- | --- | --- | --- |
| Описание класса | | | | |
| Класс *AcceleratedMotion* – равноускоренное движение | | | | |
| Поля | | | | |
| – \_acceleration | | double | | Ускорение |
| – \_velocity | | double | | Скорость |
| Свойства | | | | |
| + Acceleration | | double | | Ускорение |
| + Coordinate | | double | | Координата |
| + MotionType | | string | | Тип движения (равноускоренное) |
| + Parameters | | string | | Параметры движения (скорость, ускорение) |
| + Velocity | | double | | Скорость |
| Методы | | | | |
| + CalculateCoordinate | double | | Расчет координаты для равноускоренного движения | |

Таблица 4 – Описание класса OscillatoryMotion

| Название | | Тип | | Описание |
| --- | --- | --- | --- | --- |
| Описание класса | | | | |
| Класс *OscillatoryMotion* – колебательное движение | | | | |
| Поля | | | | |
| – \_amplitude | | double | | Амплитуда |
| – \_frequency | | double | | Частота |
| Свойства | | | | |
| + Amplitude | | double | | Амплитуда |
| + Coordinate | | double | | Координата |
| + Frequency | | double | | Частота |
| + MotionType | | string | | Тип движения (колебательное) |
| + Perimeter | | double | | Периметр треугольника |
| + Parameters | | string | | Параметры движения (амплитуда, частота) |
| Методы | | | | |
| + CalculateCoordinate | double | | Расчет координаты для колебательного движения | |

1. **Дерево ветвлений Git**

Git [2] — распределённая система управления версиями файлов. Система спроектирована как набор утилит командной строки, специально разработанных с учётом их использования в скриптах. Git обеспечивает управление версиями кода и способствует эффективной совместной работе над проектом.

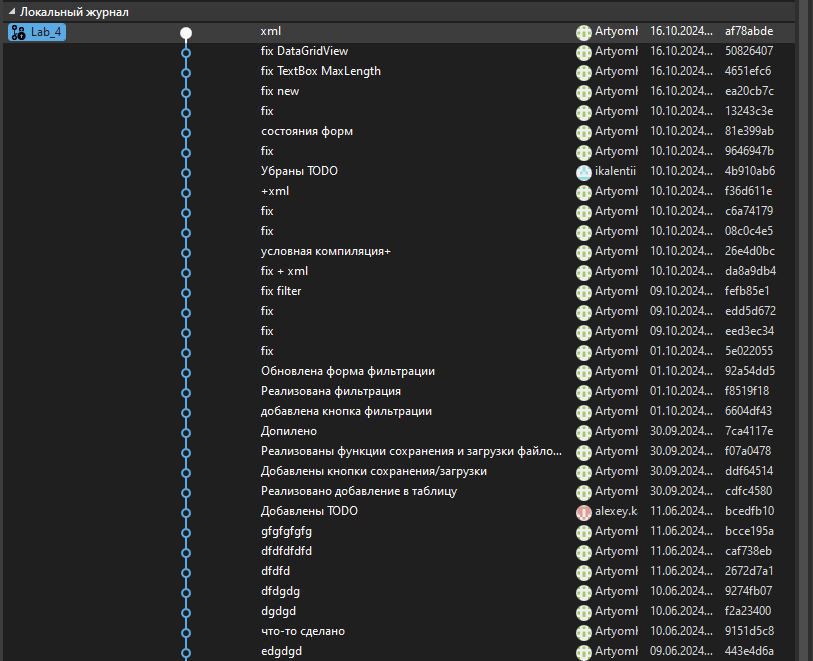


Рисунок 3 – Дерево Git

1. **Тестирование программы**

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

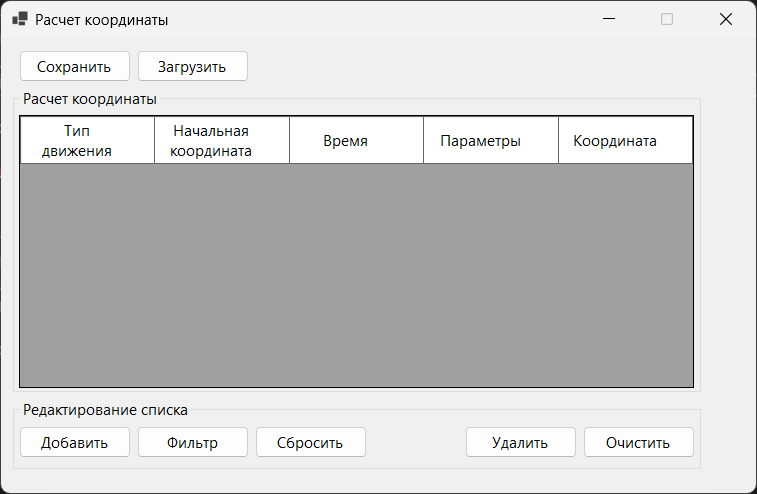


Рисунок 4 – Графический интерфейс пользователя

* 1. **Тестовый случай «Добавить движение»**

Для добавления движения необходимо вызвать соответствующую форму путём нажатия кнопки «Добавить», рисунок 5.

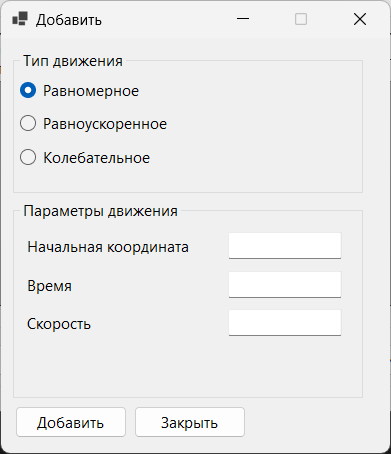


Рисунок 5 – Форма для добавления фигуры

Параметры любого движения (равномерного, равноускоренного, колебательного) можно ввести, отметив соответствующий выбор с помощью кнопки.

После ввода данных необходимо нажать кнопку «Добавить», фигура появится в таблице главной формы, рисунки 6-8).

|  |  |
| --- | --- |
|  |  |

Рисунок 6 – Добавление равномерного движения

|  |  |
| --- | --- |
|  |  |

Рисунок 7 – Добавление равноускоренного движения

|  |  |
| --- | --- |
|  |  |

Рисунок 8 – Добавление колебательного движения

В программе предусмотрена система обработки некорректного ввода данных пользователем. Например, пользователь не сможет ввести несколько точек в числе или отрицательное число, также программа не позволит ввести любой символ кроме точки или цифры. Если пользователь оставляет поле незаполненным и нажимает «Добавить», то появится соответствующее сообщение об ошибке, рисунки 9–10.

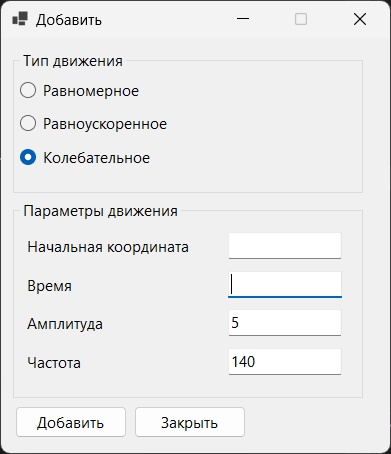


Рисунок 9 – Некорректный ввод

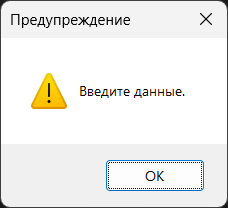


Рисунок 10 –Предупреждение пользователя о некорректном вводе

* 1. **Тестовый случай «Удалить»**

Для удаления одного или нескольких движений необходимо выбрать их в таблице и нажать на кнопку «Удалить», рисунки 11–14.

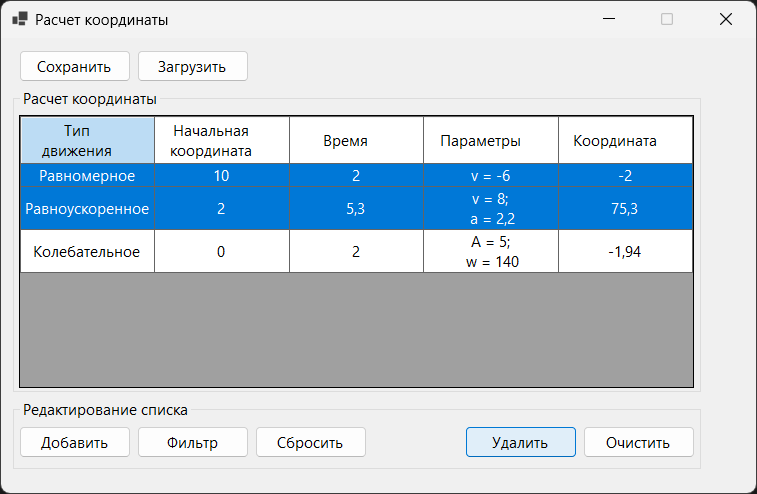


Рисунок 11 – Удаление движений

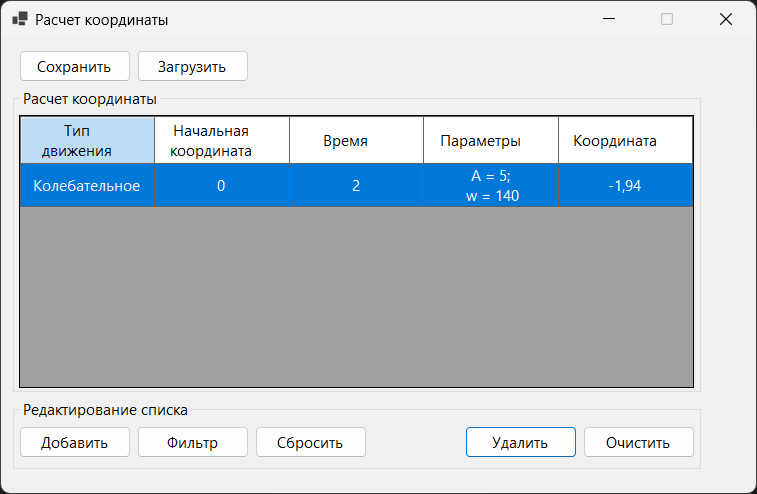


Рисунок 12 – Результат нажатия кнопки «Удалить»

* 1. **Тестовый случай «Очистить»**

Для полного очищения списка необходимо нажать кнопку «Очистить» (рисунки 13-14).

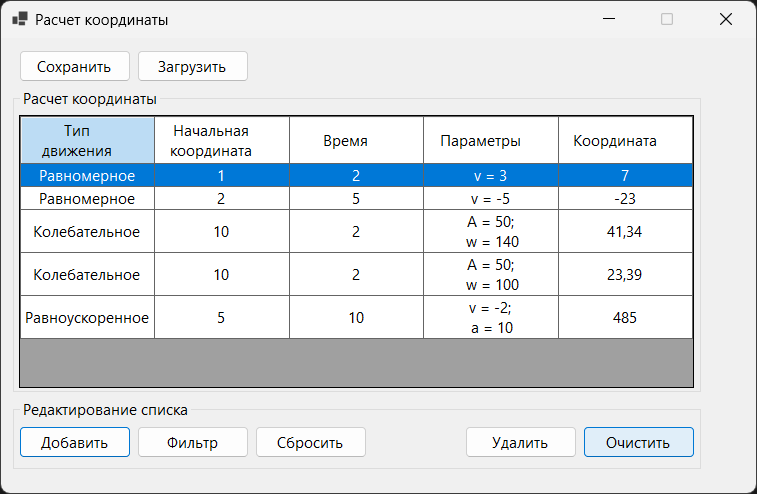


Рисунок 13 – Результат нажатия кнопки «Удалить»

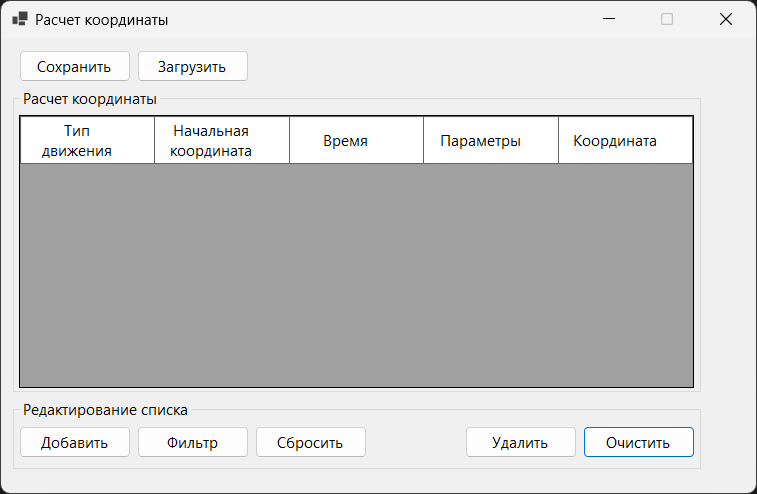


Рисунок 14 – Результат нажатия кнопки «Удалить»

* 1. **Тестовый случай «Отфильтровать список»**

Для фильтрации фигур по определенным критериям необходимо нажать кнопку «Фильтр» (рисунок 15).

|  |  |
| --- | --- |
|  |  |

Рисунок 15 – Вызов формы для фильтрации

Ниже приведены примеры фильтрации списка фигур по разным критериям.

* Тип движения «Равномерное»

|  |  |
| --- | --- |
|  |  |

Рисунок 16 – Фильтрация по типу движения «Равномерное»

* Тип движения «Равноускоренное»

|  |  |
| --- | --- |
|  |  |

Рисунок 17 – Фильтрация по типу движения «Равноускоренное»

* Тип движения «Колебательное»

|  |  |
| --- | --- |
|  |  |

Рисунок 18 – Фильтрация по типу движения «Колебательное»

* Значение «Начальная координата»

|  |  |
| --- | --- |
|  |  |

Рисунок 19 – Фильтрация по значению «Начальная координата»

* Значение «Время»

|  |  |
| --- | --- |
|  |  |

Рисунок 20 – Фильтрация по значению «Время»

* Значения «Начальная координата» и «Время»

|  |  |
| --- | --- |
|  |  |

Рисунок 21 – Фильтрация по значениям «Начальная координата» и «Время»

* Тип «Равномерное» и «Колебательное»

|  |  |
| --- | --- |
|  |  |

Рисунок 22 – Фильтрация по типам движения «Равномерное» и «Колебательное»

* Тип «Колебательное» и значение «Начальная координата»

|  |  |
| --- | --- |
|  |  |

Рисунок 23 – Фильтрация по типу «Колебательное» и значению «Начальная координата»

На рисунках 24 и 25 представлена реакция системы на некорректное поведение пользователя.

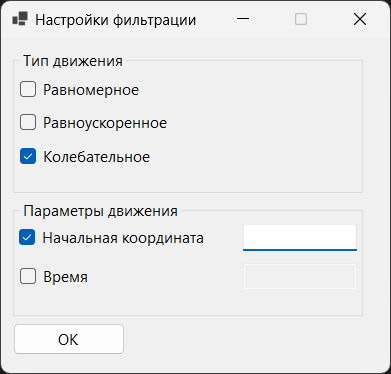


Рисунок 24 – Нажатие кнопки «ОК» при незаполненном поле «Начальная координата»

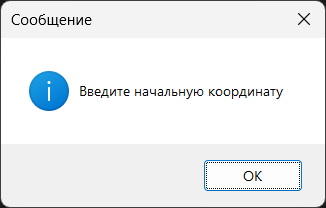


Рисунок 25 – предупреждение пользователя

* 1. **Тестовый случай «Сброс фильтра»**

На рисунках 26 и 27 представлена работа функции сброса фильтра.

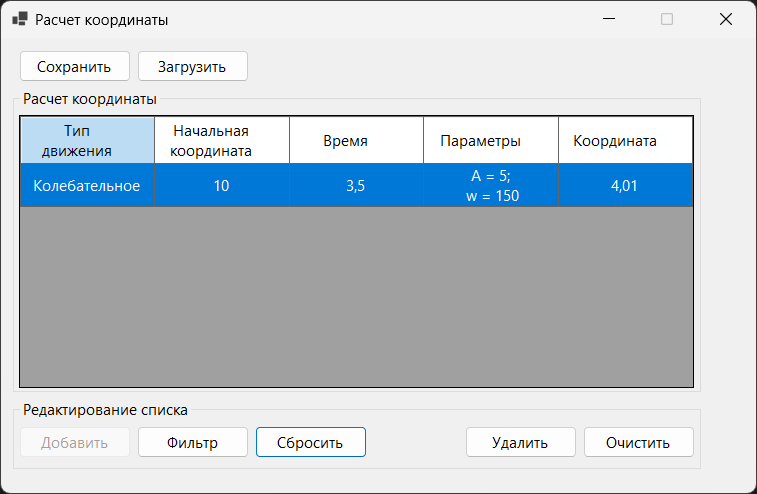


Рисунок 26 – Нажатие кнопки Сброс

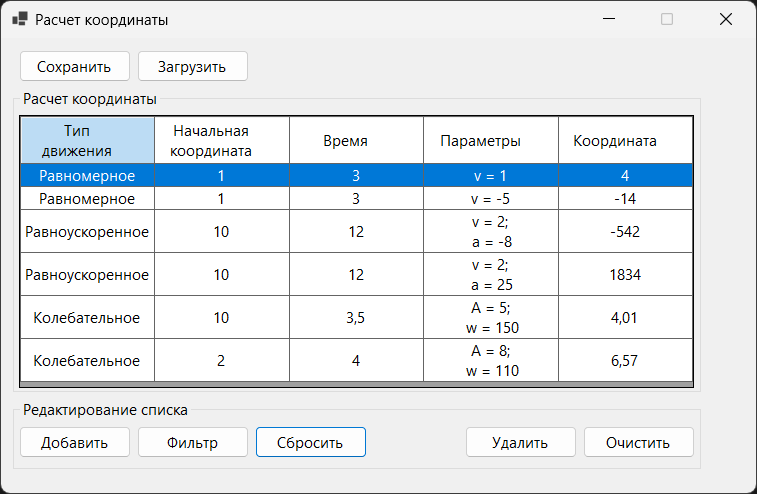


Рисунок 27 – Отображение исходного списка после сброса фильтра

* 1. **Тестовый случай «Сохранить файл»**

Для сохранения данных в файл необходимо нажать на кнопку «Сохранить», рисунок 28. Откроется системный диалог сохранения файла, где пользователь выбирает директорию и указывает имя файла, рисунок 29.

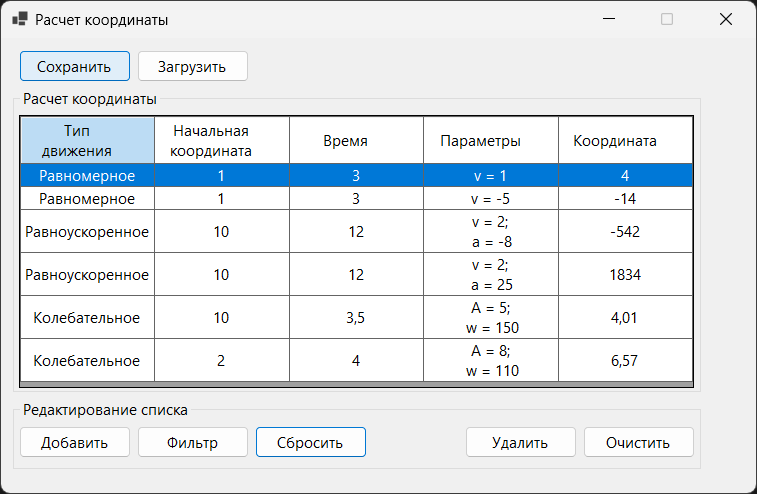


Рисунок 28 – Сохранение данных

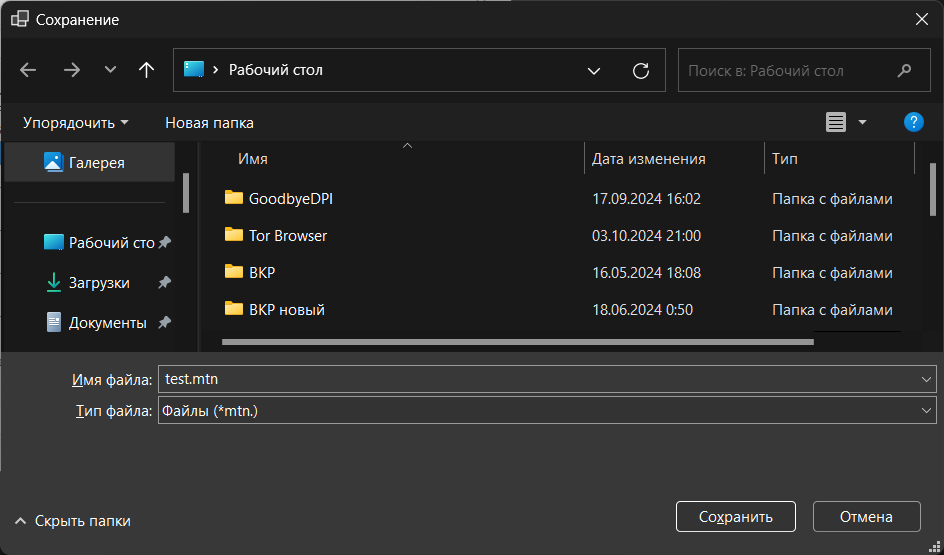


Рисунок 29 – Сохранение файла

В случае, если таблица пуста, сохранение не производится (рисунок 30).

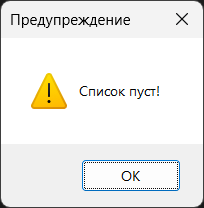


Рисунок 30 – Результат нажатия на кнопку «Сохранить» при пустой таблице

* 1. **Тестовый случай «Загрузить файл»**

Для загрузки данных в таблицу необходимо нажать на кнопку «Загрузить», рисунок 31.

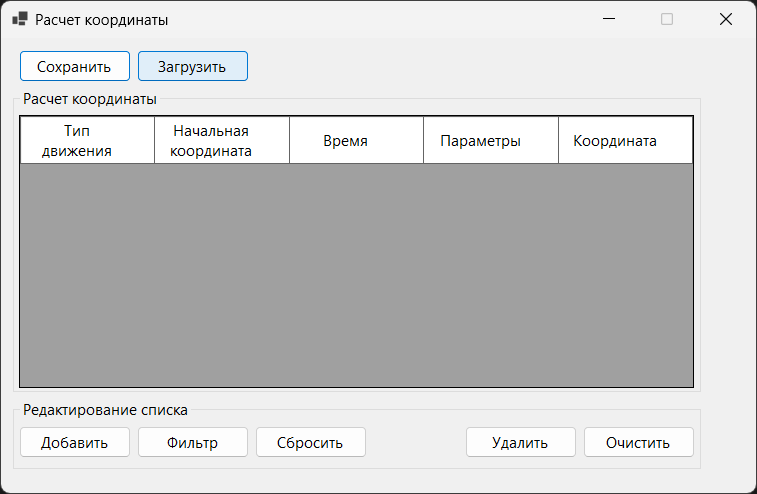


Рисунок 31 – Загрузить данные в таблицу

Далее откроется системный диалог загрузки файла, рисунок 32.

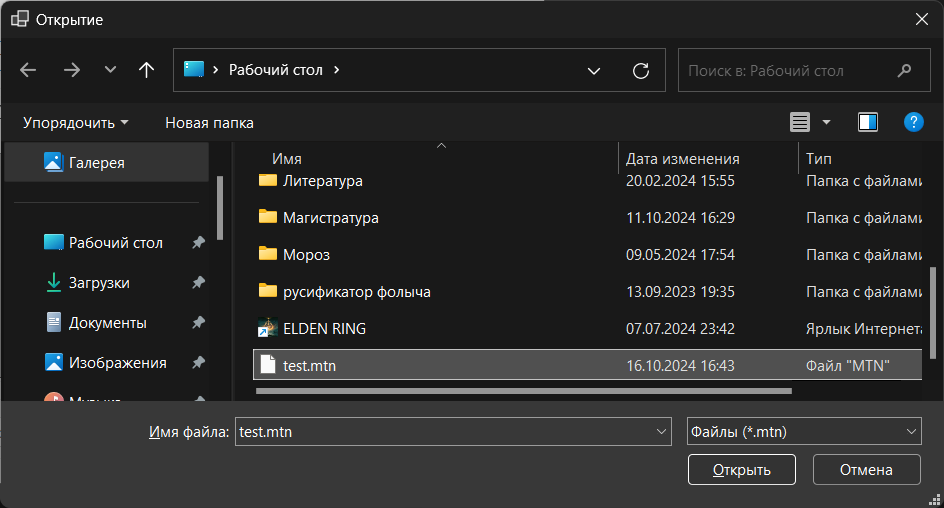


Рисунок 32 – Выбор файла для загрузки

В случае, если структура загружаемого файла не распознана, в случае если в файле присутствуют некорректные значения параметров или значения отсутствуют, появится соответствующее сообщение, представленное на рисунке 33.

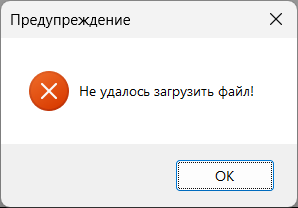


Рисунок 33 – Загрузка некорректного файла

**Список 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.
2. git [Электронный ресурс]. — URL : http://git-scm.com/ (Дата обращения: 11.06.2024)

**ПРИЛОЖЕНИЕ А**

**Техническое задание на создание автоматизированной системы**

Система расчёта координат

Разработчик: студент гр. [О-5КМ21](https://iso.tpu.ru/magistracy/rasp/rasp_5am0r/) НИ ТПУ Краснобаев А.Д.

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

Томск 2024

**1 Общие сведения**

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

Полное наименование: «Система расчета координат».

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-5КМ21](https://iso.tpu.ru/magistracy/rasp/rasp_5am0r/) НИ ТПУ Краснобаев А.Д.

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

Начало работ: 10 мая 2024 г.

Окончание работ: 20 июня 2024 г.

**2 Назначение и цели создания системы**

**2.1 Назначение системы**

Система предназначена для расчёта площадей плоских фигур: окружности, прямоугольника, треугольника.

**2.2 Цели создания системы**

Система создаётся в целях сокращения трудозатрат людей, которым необходимо быстро рассчитывать площади плоских поверхностей~~.~~

**3 Характеристика объектов автоматизации**

Расчет координат при различных типах движения может пригодиться в любой сфере жизнедеятельности человека.

Поскольку такие расчёты выполняются повсеместно и довольно часто, представляется целесообразным автоматизировать этот процесс.

**4 Требования к системе**

Таблица А.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 фигур должны храниться в XML-файле с расширением \*.mtn.

Структура XML-документа представляет собой иерархическую организацию элементов, начиная от корневого элемента и включая вложенные элементы. В случае данного XML-документа структура будет следующей:

**<ArrayOfMotionBase>** <!-- Корневой элемент -->

**<MotionBase>** <!—Элемент, представляющий движение -->

<!-- Дополнительные элементы, зависящие от типа движения -->

**</MotionBase>**

<!-- Возможно, другие элементы MotionBase, представляющие другие движения -->

**</ ArrayOfMotionBase >**

Для каждого типа геометрической фигуры внутри элемента *MotionBase* будут различные дочерние элементы, содержащие информацию о характеристиках каждой фигуры.

* Для типа*ConstantMotion*:

**<MotionBase xsi:type="ConstantMotion">**

**<InitialValue>** <!--Начальная координата--> **</InitialValue>**

**<Time>** <!--Время--> **</Time>**

**<Velocity>** <!--Скорость--> **</Velocity>**

**</MotionBase>**

* Для типа*AcceleratedMotion*:

**<MotionBase xsi:type="AcceleratedMotion">**

**<InitialValue>** <!--Начальная координата--> **</InitialValue>**

**<Time>** <!--Время--> **</Time>**

**<Velocity>** <!--Скорость--> **</Velocity>**

**<Acceleration>** <!--Ускорение--> **</Acceleration>**

**</MotionBase>**

* Для типа*OscillatoryMotion*:

**<MotionBase xsi:type="OscillatoryMotion">**

**<InitialValue>** <!--Начальная координата--> **</InitialValue>**

**<Time>** <!--Время--> **</Time>**

**<Amplitude>** <!--Амплитуда--> **</Amplitude>**

**<Frequency>** <!--Частота--> **</Frequency>**

**</MotionBase>**

**4.3 Функциональные требования**

**F01.** Система должна рассчитывать координату для следующих типов движения:

* равномерное;
* равноускоренное;
* колебательное.

**F01.01.** Координата при равномерном движении должна определяться по выражению:

где – начальная координата, м;

– скорость, м/с;

– время, с

**F01.02.** Координата при равноускоренном движении должна определяться по выражению

где – ускорение, м/с2;

**F01.03.** Координата при колебательном движении должна определяться по выражению:

где – амплитуда колебаний, м;

– частота, Гц.

**F02.** В системе должен быть реализован список элементов фигур.

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

* тип движения;
* начальная координата;
* время;
* параметры, характерные для определенного типа движения.

**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 фильтрации 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 64-разрядной операционной системе Windows 10 версии 22H2 и выше. Работоспособность на других выпусках и версиях не гарантируется.

**C02.** На рабочей станции должен быть установлен .NET Framework версии 8.0.

**4.6 Требования к аппаратному обеспечению**

**C03.** Частота процессора не менее 2500 МГц.

**C04.** Объём оперативной памяти не менее 2 ГБ.

**C05.** Объём HDD/SSD памяти не менее 1 ГБ.