**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) | Коновалова Е.А. |
| Проверил: доцент, к.т.н. | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись)  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (дата) | Калентьев А.А. |
|  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (балл) |  |

Томск - 2024

СОДЕРЖАНИЕ

[Введение 3](#_Toc184131632)

[1 UML диаграмма вариантов использования 4](#_Toc184131633)

[2 UML диаграмма классов 5](#_Toc184131634)

[3 Дерево ветвлений Git 8](#_Toc184131635)

[4 Тестирование программы 10](#_Toc184131636)

[4.1 Тестовый случай «Добавить движение» 10](#_Toc184131637)

[4.2 Тестовый случай «Удалить» 14](#_Toc184131638)

[4.3 Тестовый случай «Отменить удаление» 15](#_Toc184131639)

[4.4 Тестовый случай «Настроить фильтр» 17](#_Toc184131640)

[4.5 Тестовый случай «Сброс фильтра» 20](#_Toc184131641)

[4.6 Тестовый случай «Сохранить файл» 21](#_Toc184131642)

[4.7 Тестовый случай «Загрузить файл» 22](#_Toc184131643)

[Список источников 24](#_Toc184131644)

[ПРИЛОЖЕНИЕ А 25](#_Toc184131645)

**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 калорий для различных типов упражнений: бег, плавание, жим штанги. 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.

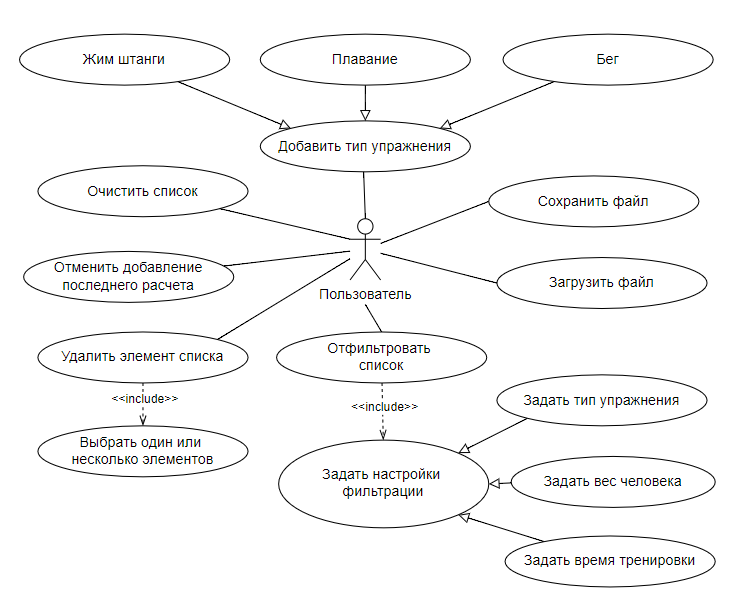


Рисунок 1 – Диаграмма вариантов использования

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.

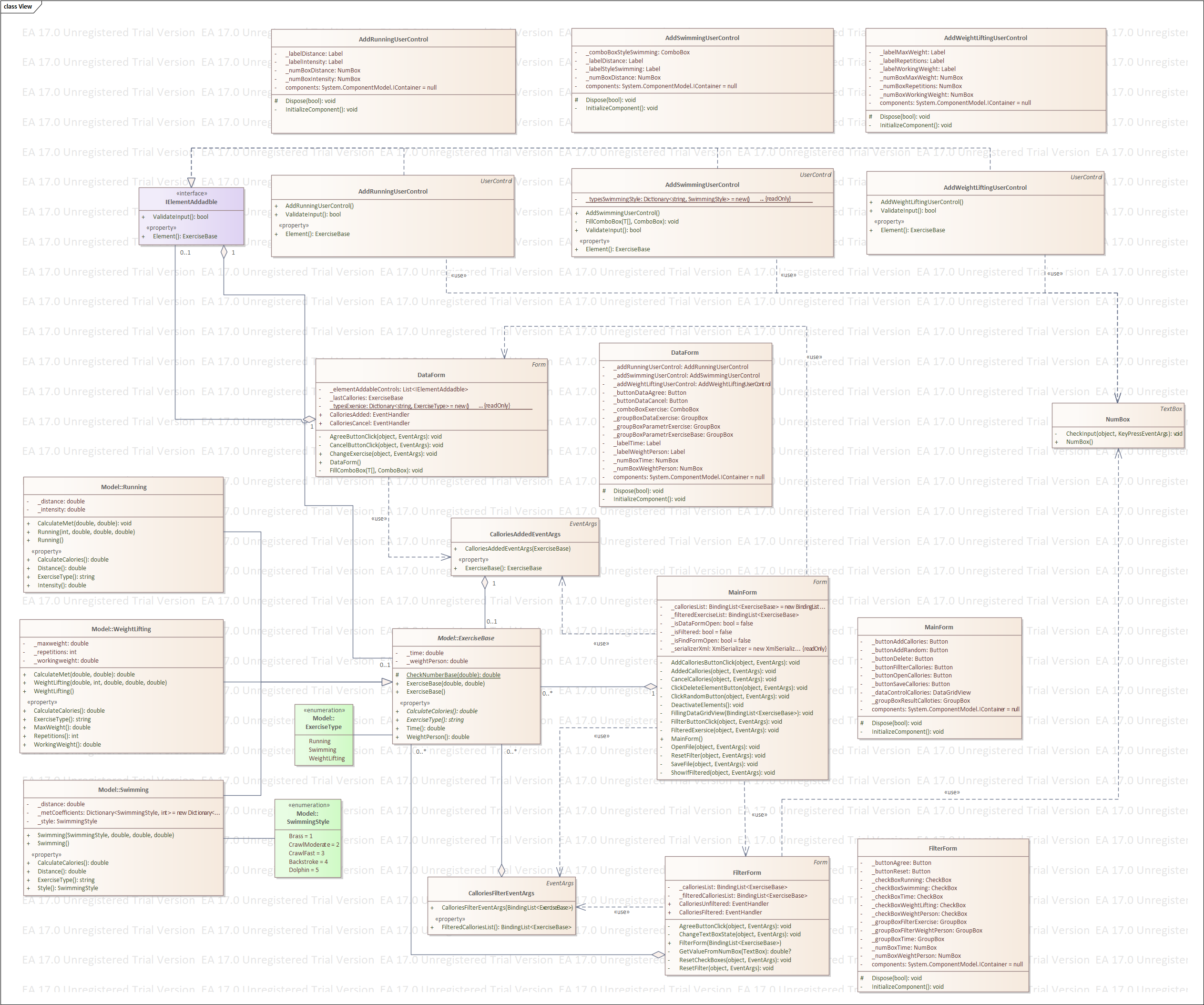


Рисунок 2 – UML диаграмма классов

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

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

Таблица 1 – Описание класса ExerciseBase.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название | Тип | | | Описание |
| Описание класса | | | | |
| Класс *ExerciseBase* – абстрактный базовый класс, описывает общую подход расчета количества калорий, сжигаемых при выполнений разных видов упражнений | | | | |
| Поля | | | | |
| - \_time | | double | Время тренировки | |
| - \_weightPerson | | double | Вес человека | |
| Свойства | | | | |
| + ExerciseType | string | | | Тип упражнения.  Абстрактное свойство, переопределяется в производных классах. |
| + WeightPerson | double | | | Вес человека.  Общее свойство для всех производных классов. |
| + Time | double | | | Время тренировки.  Общее свойство для всех производных классов. |
| Методы | | | | |
| + CalculateCalories | double | | | Расчет количества калорий  Абстрактный метод, переопределяется в производных классах. |
| + CheckNumberBase | double | | | Метод для проверки введенного числа Абстрактный метод, переопределяется в производных классах. |

В таблицах 2–4 приведены описания классов *Running*, *Swimming* и *WeightLifting*, которые наследуются от *ExerciseBase*.

Таблица 2 – Описание класса Running

| Название | Тип | Описание |
| --- | --- | --- |
| Описание класса | | |
| Класс *Running* – тип упражнения бег | | |
| Поля | | |
| – \_intensity | double | Интенсивность |
| – \_distance | double | Дистанция |
| Свойства | | |
| + Intensity | double | Интенсивность |
| + ExerciseType | string | Тип упражнения (бег) |
| + Distance | double | Дистанция |
| + WeightPerson | double | Вес человека |
| + Time | double | Время тренировки |
| Методы | | |
| + CalculateCalories | double | Расчет калорий, саженных при типе упражнения бег |
| + CalculateMet | double | Определение MET |

Таблица 3 – Описание класса Swimming

| Название | | Тип | Описание |
| --- | --- | --- | --- |
| Описание класса | | | |
| Класс *Swimming* – тип упражнения плавание | | | |
| Поля | | | |
| – \_style | | SwimmingStyle | Стиль плавания |
| – \_distance | | double | Дистанция |
| Свойства | | | |
| + Style | | SwimmingStyle | Стиль плавания |
| + Distance | | double | Дистанция |
| + ExerciseType | | string | Тип упражнения (плавание) |
| + WeightPerson | | double | Вес человека |
| + Time | | double | Время тренировки |
| Методы | | | |
| + CalculateCalories | double | | Расчет калорий, саженных при типе упражнения плавание |

Таблица 4 – Описание класса WeightLifting

| Название | | Тип | | Описание |
| --- | --- | --- | --- | --- |
| Описание класса | | | | |
| Класс WeightLifting – тип упражнения жим штанги | | | | |
| Поля | | | | |
| – \_workingweight | | double | | Вес во время тренировки |
| – \_maxweight | | double | | Максимальный вес, который может сделать человек за 1 повторение |
| – \_repetitions; | | int | | Количество повторений |
| Свойства | | | | |
| + WorkingWeight | | double | | Рабочий вес |
| + Repetitions | | double | | Количество повторений |
| + MaxWeight | | double | | Максимальный вес, который может сделать человек за 1 повторение |
| + ExerciseType | | string | | Тип упражнения (жим штанги) |
| + WeightPerson | | double | | Вес человека |
| + Time | | double | | Время тренировки |
| Методы | | | | |
| + CalculateCoordinate | double | | Расчет калорий, саженных при типе упражнения жим штанги | |
| + CalculateMet | double | | Рассчет MET | |

В табл. 5 приведено описание интерфейса *IElementAddable*.

Таблица 5 – Описание интерфейса IElementAddable

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Описание класса | | |
| IElementAddable – Интерфейс добавления расчета калорий по типу упражнения | | |
| Свойства | | |
| + Element | ExerciseBase | Добавляемый тип упражнения |

В табл. 6–8 приведены описания классов AddRunningUserControl, AddSwimmingUserControl и AddWeightLiftingUserControl которые наследуются от интерфейса IElementAddable.

Таблица 6 – Описание класса AddRunningUserControl

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Описание класса | | |
| Класс Running – бег | | |
| Свойства | | |
| + Element | ExerciseBase | Добавляемый тип упражнения - бег |
| Методы | | |
| + ValidateInput | bool | Метод проверки введенных данные |

Таблица 7 – Описание класса AddSwimmingUserControl

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Описание класса | | |
| Класс Swimming – плавание | | |
| Свойства | | |
| + Element | ExerciseBase | Добавляемый тип упражнения - плавание |
| Методы | | |
| + ValidateInput | bool | Метод проверки введенных данные |

Таблица 8 – Описание класса AddWeightLiftingUserControl

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Описание класса | | |
| Класс WeightLifting – жим штанги | | |
| Свойства | | |
| + Element | ExerciseBase | Добавляемый тип упражнения – жим штанги |
| Методы | | |
| + ValidateInput | bool | Метод проверки введенных данные |

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.

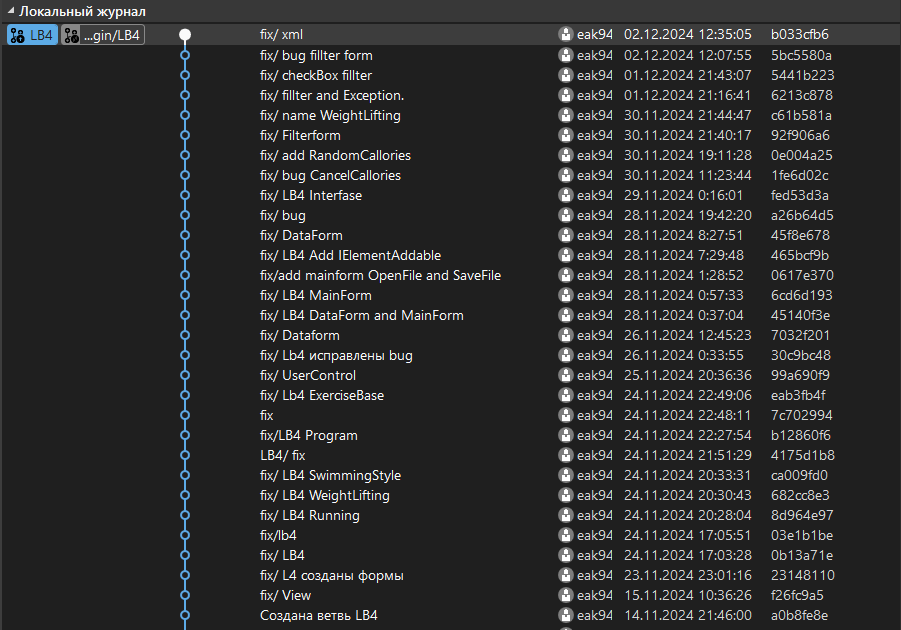


Рисунок 3 – Дерево Git

1. **Тестирование программы**

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

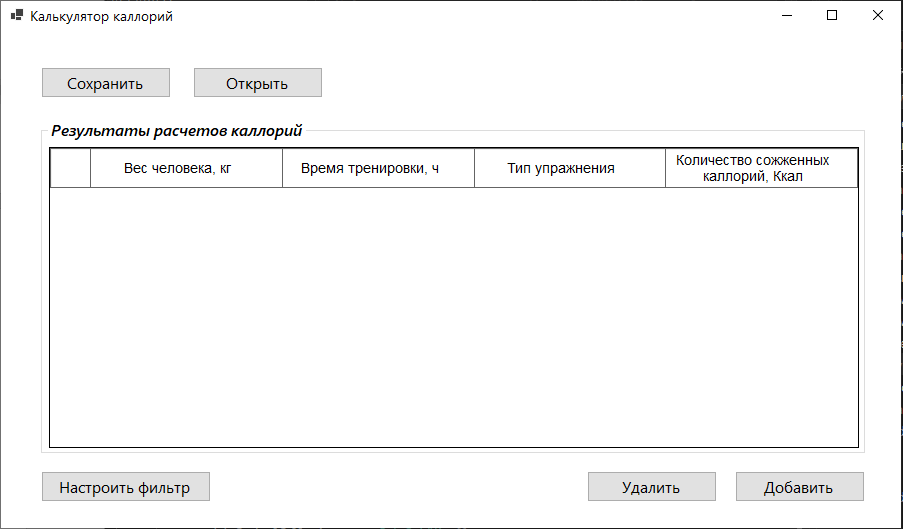


Рисунок 4 – Графический интерфейс пользователя

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

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

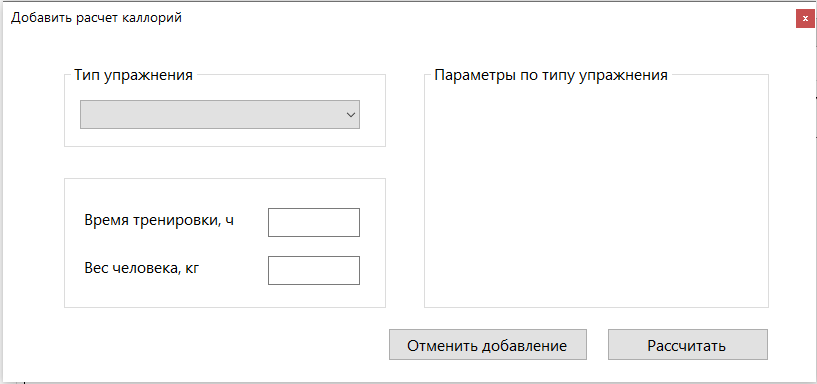
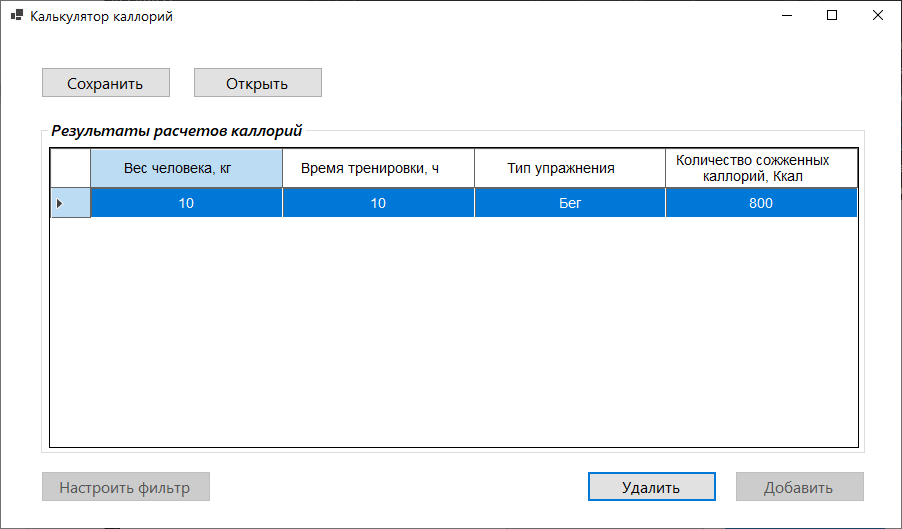


Рисунок 5 – Форма для добавления фигуры

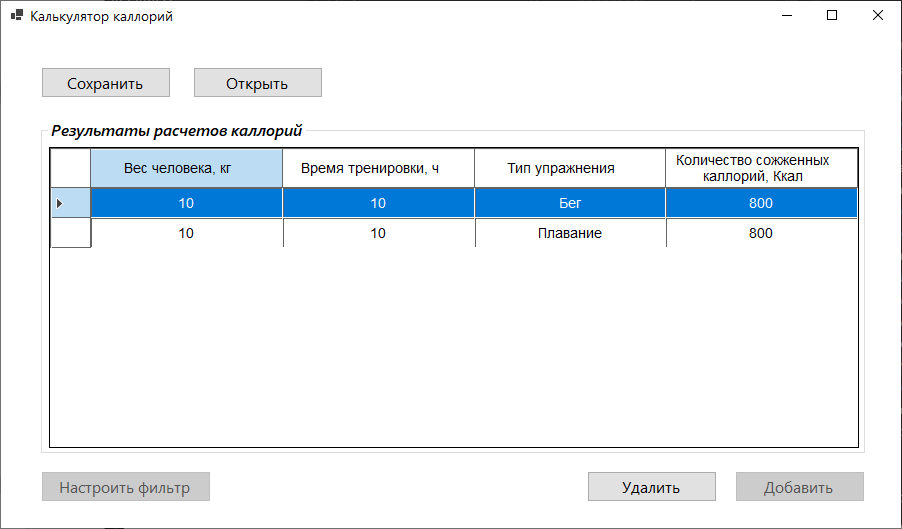
Параметры любого типа упражнения (бег, плавание и жим штанги) можно ввести, отметив соответствующий выбор с помощью кнопки.

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



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

Рисунок 6 – Добавление упражнения бег



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

Рисунок 7 – Добавление упражнения плавание

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

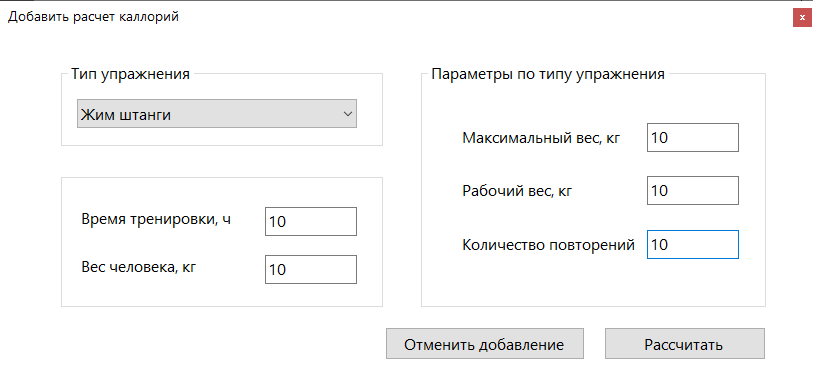


Рисунок 8 – Добавление упражнения жим штанги

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.

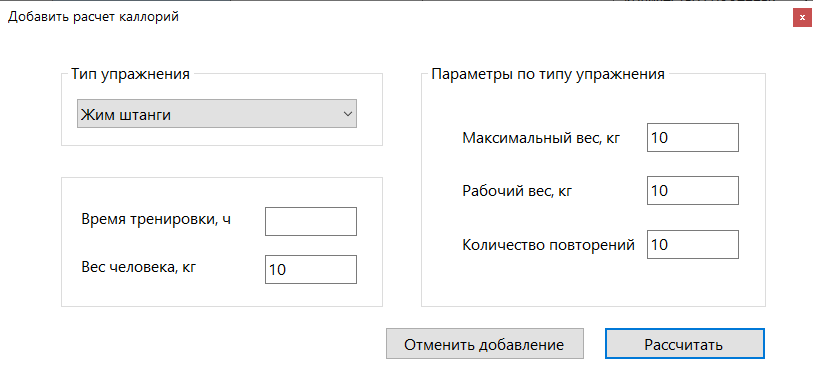


Рисунок 9 – Некорректный ввод

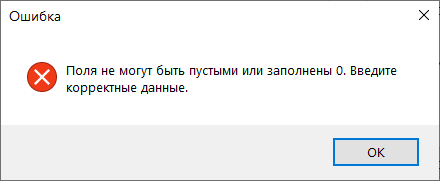


Рисунок 10 – Предупреждение пользователя о некорректном вводе

* 1. **Тестовый случай «Удалить»**

Для удаления одного или нескольких движений необходимо выбрать их в таблице и нажать на кнопку «Удалить», рисунки 11-12.

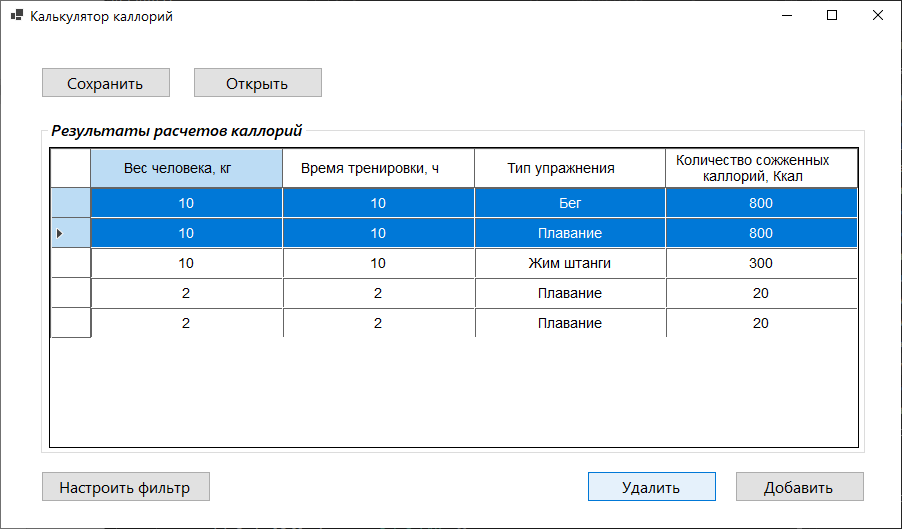


Рисунок 11 – Удаление движений

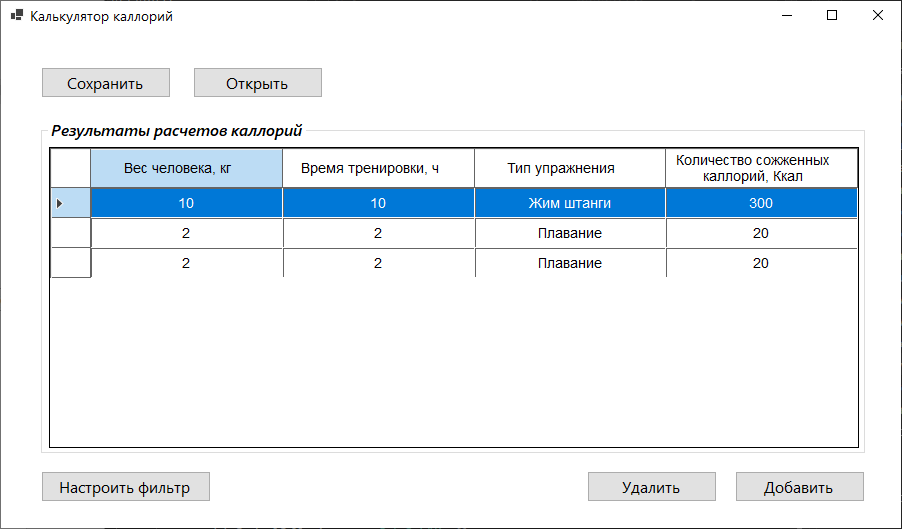


Рисунок 12 – Результат нажатия кнопки «Удалить»

* 1. **Тестовый случай «Отменить удаление»**

Для отмены добавления необходимо нажать кнопку «Отменить добавление» (рисунки 13-14).

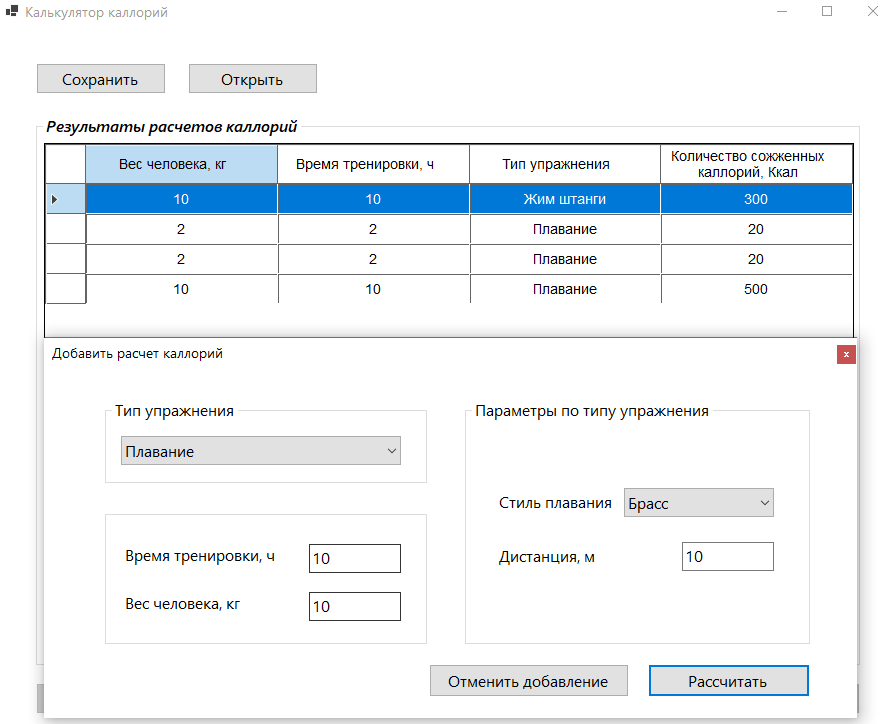


Рисунок 13 – Добавление расчета в таблицу

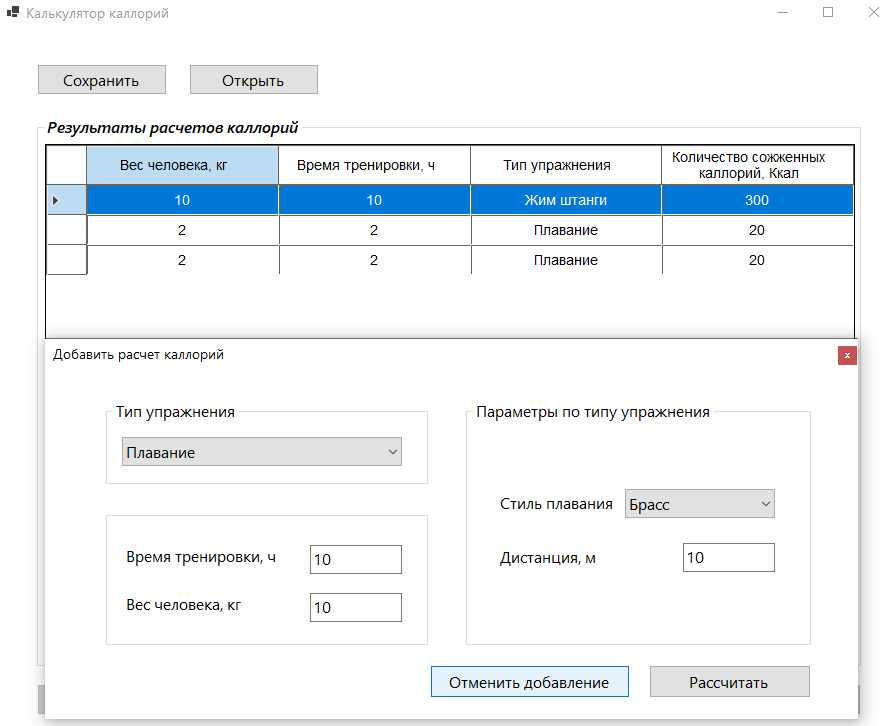


Рисунок 14 – Результат нажатия кнопки «Отменить удаление»

* 1. **Тестовый случай «Настроить фильтр»**

Для фильтрации расчета калорий по определенным критериям необходимо нажать кнопку «Настроить фильтр» (рисунок 15).

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

Рисунок 15 – Вызов формы для фильтрации

Ниже приведены примеры фильтрации списка результатов расчета калорий по разным критериям.

* Тип упражнения «Бег»

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

Рисунок 16 – Фильтрация по типу упражнения «Бег»

* Тип упражнения «Плавание»

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

Рисунок 17 – Фильтрация по типу упражнения «Плавание»

* Тип упражнения «Жим штанги»

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

Рисунок 18 – Фильтрация по типу упражнения «Жим штанги»

* Значение «Вес человека»

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

Рисунок 19 – Фильтрация по значению «Вес человека»

* Значение «Время тренировки»

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

Рисунок 20 – Фильтрация по значению «Время тренировки»

* Значения «Вес человека» и «Время тренировки»

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

Рисунок 21 – Фильтрация по значениям «Вес человека» и «Время»

* Тип «Бег» и «Плавание»

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

Рисунок 22 – Фильтрация по типам упражнения «Бег» и «Плавание»

* Тип «Бег» и значение «Время тренировки»

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

Рисунок 23 – Фильтрация по типу «Бег» и значению «Время тренировки»

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

На рисунках 24 и 25 представлена работа функции сброса фильтра.

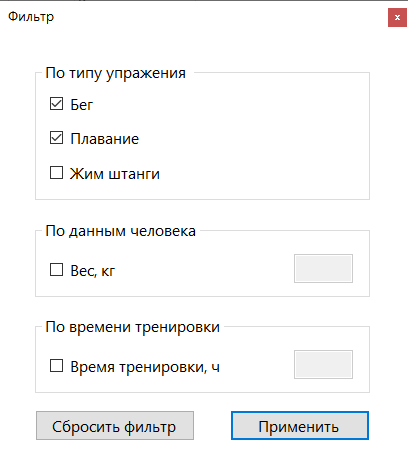
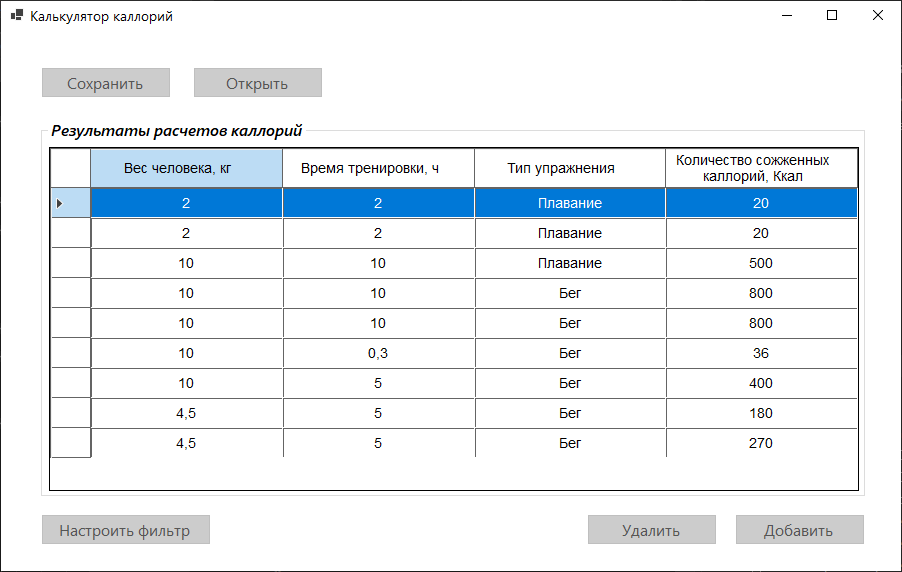


Рисунок 24 – Нажатие кнопки «Применить»

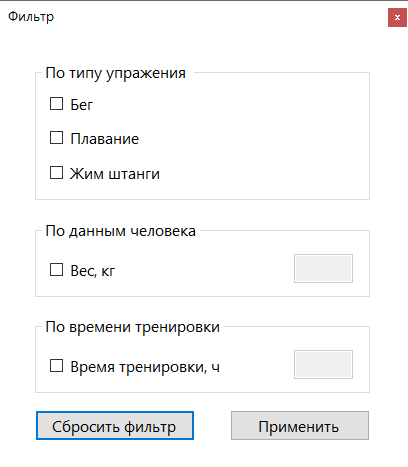
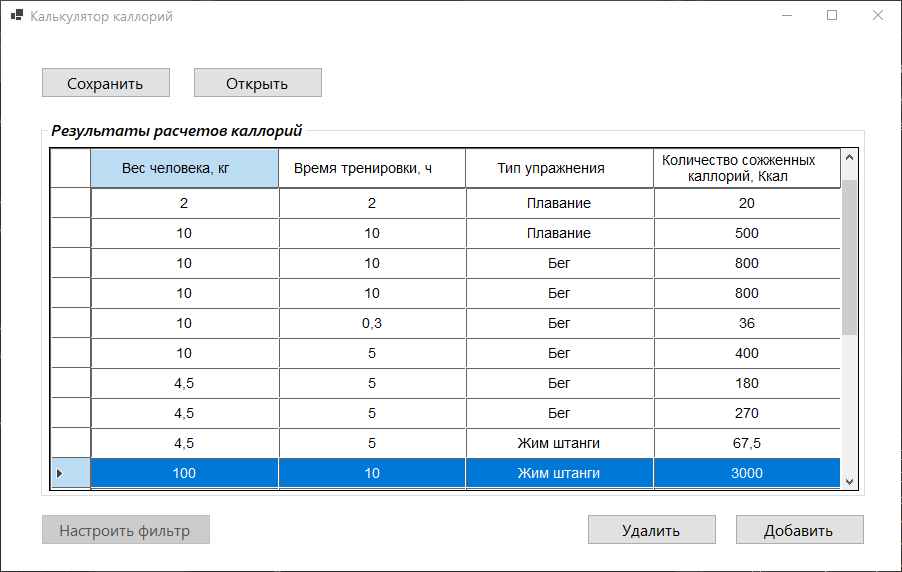


Рисунок 25 – Нажатие кнопки «Сбросить фильтр»

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

Для сохранения данных в файл необходимо нажать на кнопку «Сохранить», рисунок 26. Откроется системный диалог сохранения файла, где пользователь выбирает директорию и указывает имя файла, рисунок 27.

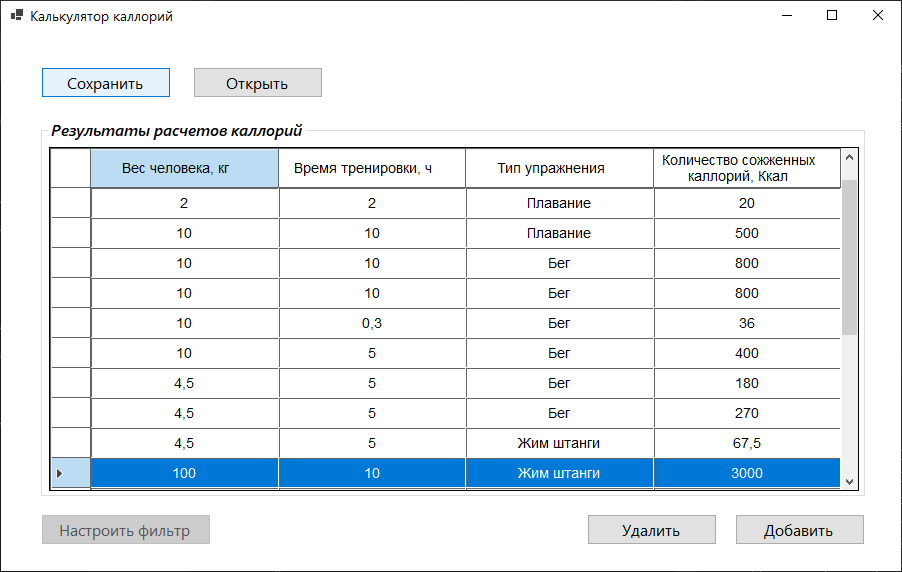


Рисунок 26 – Сохранение данных

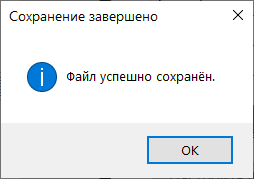
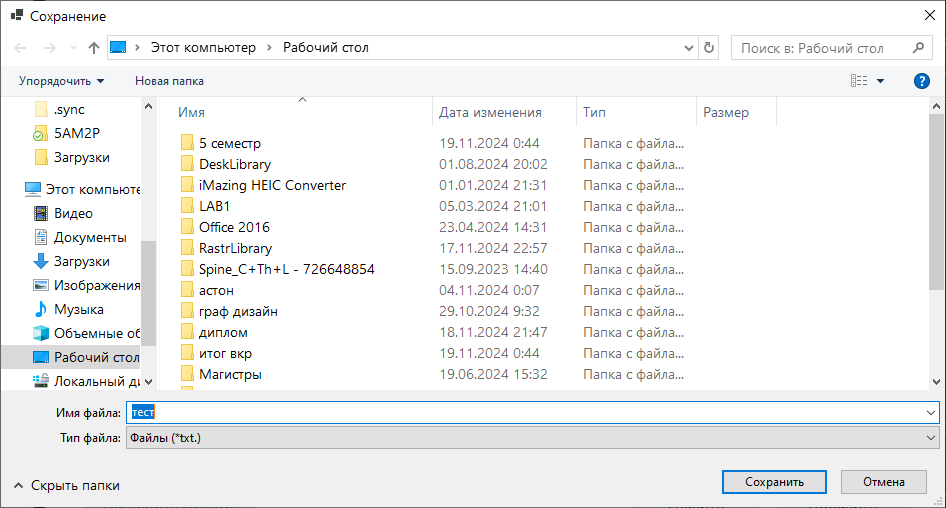


Рисунок 27 – Сохранение файла

В случае, если таблица пуста, сохранение не производится (рисунок 28).

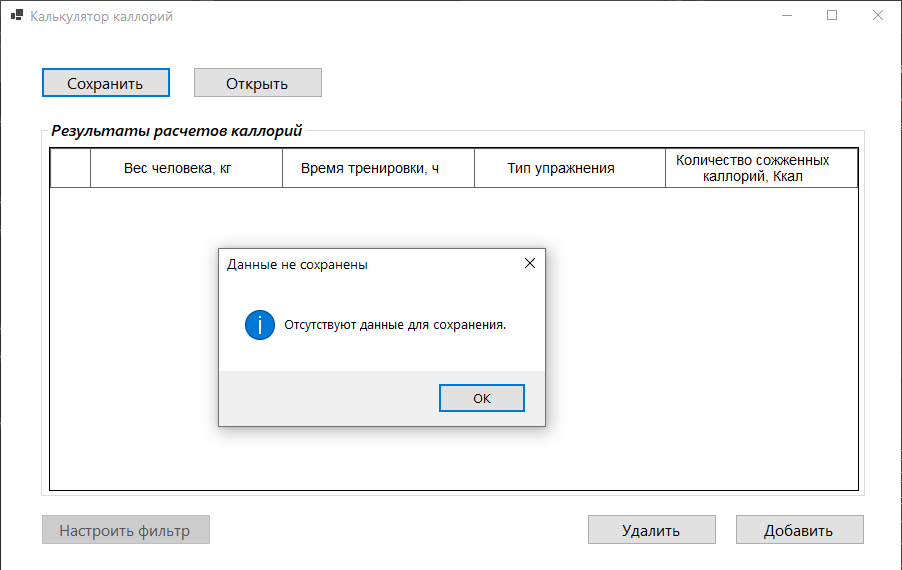


Рисунок 28 – Результат нажатия на кнопку «Сохранить» при пустой таблице

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

Для загрузки данных в таблицу необходимо нажать на кнопку «Открыть», рисунок 29.

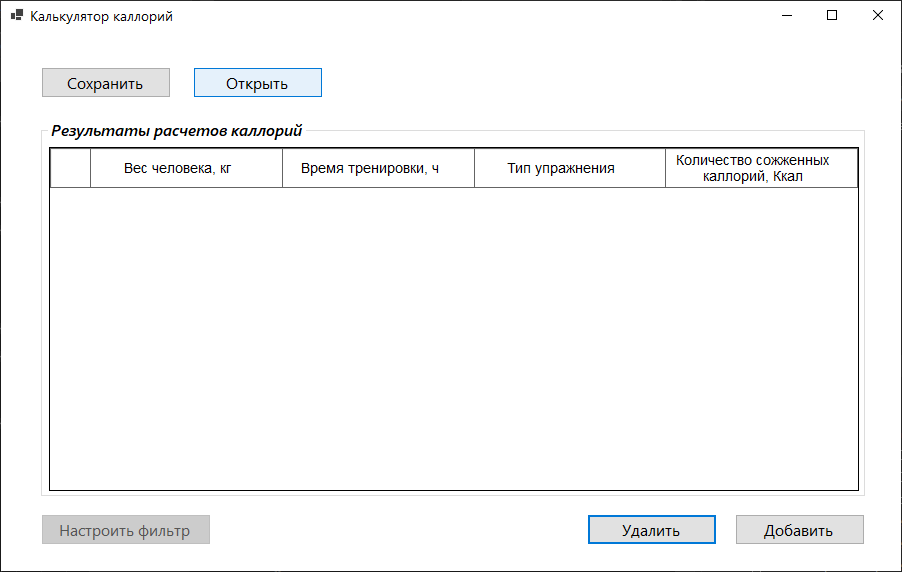


Рисунок 29 – Загрузить данные в таблицу

Далее откроется системный диалог загрузки файла, рисунок 30.

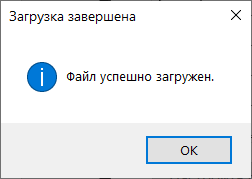
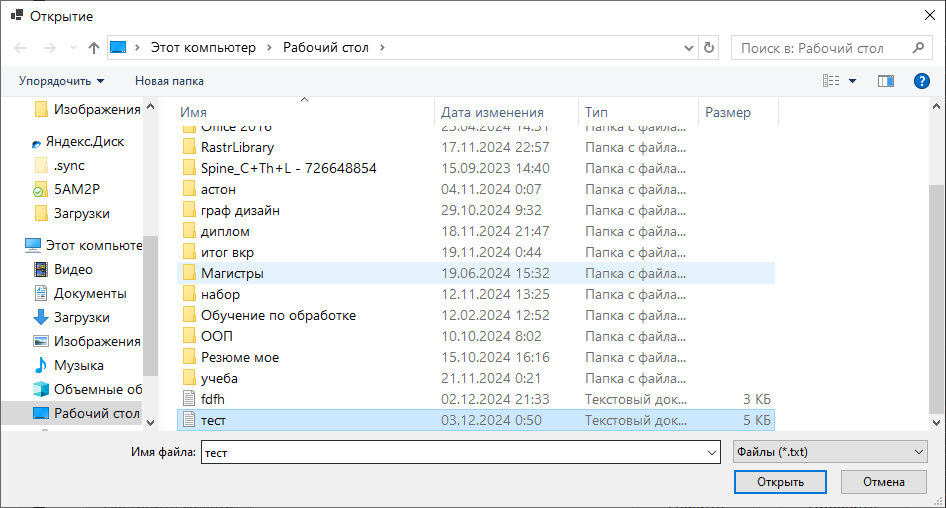


Рисунок 30 – Выбор файла для загрузки

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 31.

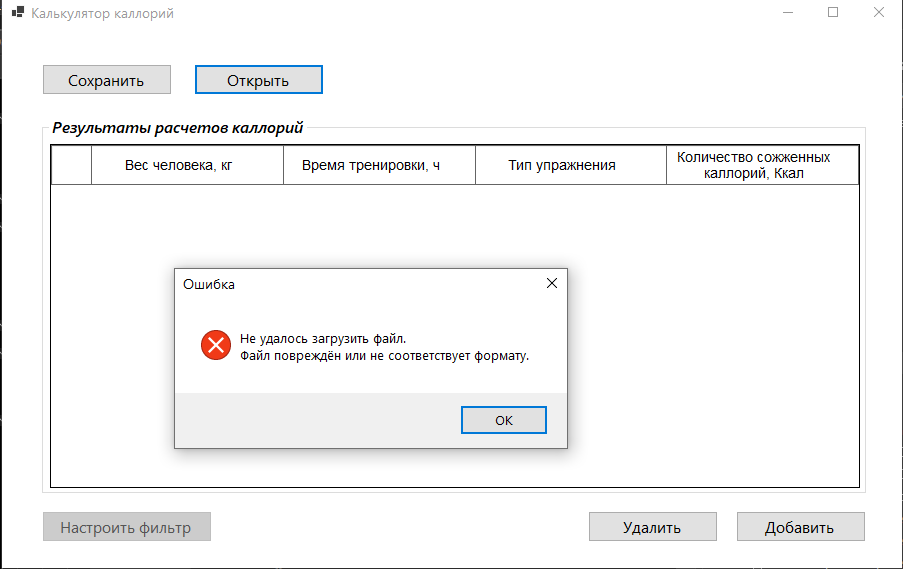


Рисунок 31 – Загрузка некорректного файла

**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 расчета калорий

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

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 калорий».

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 Коновалова Е.А,

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

Начало работ: 17 мая 2024 г.

Окончание работ: 09 ноября 2024 г.

**2 Назначение и цели создания системы**

**2.1 Назначение системы**

Система предназначена для расчёта калорий сожжённых при выполнении видов упражнений: бег, плавание, жим штанги.

**2.2 Цели создания системы**

Система создаётся в целях сокращения трудозатрат людей, которым необходимо быстро рассчитывать количество калорий сожжённых при выполнении видов упражнений: бег, плавание, жим штанги~~.~~

**3 Характеристика объектов автоматизации**

Расчет калорий, сожженных при выполнении различных видов упражнений, таких как бег, плавание, жим штанги, является ключевой функцией, которая может оказаться полезной в различных сферах человеческой деятельности. Особенно это актуально для тех, кто активно занимается спортом, осуществляет контроль за своим рационом питания и следит за общим количеством потребляемых и сожженных калорий.

Поскольку подобные расчеты требуются с определенной регулярностью и их выполнение может быть довольно трудоемким и время затратным, целесообразно внедрить автоматизированный процесс для проведения подобных вычислений. Автоматизация данной задачи сократит затраты времени и повысит точность результатов за счет исключения возможных человеческих ошибок при ручном подсчете.

**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 расчета калорий должны храниться в XML-файле с расширением \*.tran.

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

**<ArrayOfExerciseBase>** <!-- Корневой элемент -->

**<ExerciseBase** **>** <!—Элемент, представляющий расчет калорий -->

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

**</ExerciseBase** **>**

<!-- Возможно, другие элементы ExerciseBase, представляющие другие упражнения-->

**</ ArrayOfExerciseBase >**

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

* Для типа *Swimming*:

<**ExerciseBase** xsi:type="Swimming">

<**WeightPerson**> <!--Вес человека--> </**WeightPerson**>

<**Time**> <!--Время тренировки--> </**Time**>

<**Style**> <!--Стиль плавания --> </**Style**>

<D**istance**> <!--Дистанция--> </**Distance**>

</**ExerciseBase**>

* Для типа*Running:*

<ExerciseBase xsi:type="Running">

<WeightPerson> <!--Вес человека--> </WeightPerson>

<Time> <!--Время тренировки--> </Time>

<Intensity> <! --Интенсивность--> </Intensity>

<Distance> <!--Дистанция--></Distance>

</ExerciseBase>

* Для типа*WeightLifting:*

<ExerciseBase xsi:type="WeightLifting">

<WeightPerson> <!--Вес человека---> </WeightPerson>

<Time> <!--Время тренировки---> </Time>

<WorkingWeight> <!--Рабочий вес---> </WorkingWeight>

<Repetitions> <!--Количество повторений---> </Repetitions>

<MaxWeight> <!--Максимальный вес---> </MaxWeight>

</ExerciseBase

**4.3 Функциональные требования**

**F01.** Система должна рассчитывать количество сожжённых калорий для типа упражнений:

- бег;

- плавание;

- жим штанги.

**F01.01.** Количество сожжённых калорий при беге должно определяться по выражению:

Интенсивность = МЕТ = Дистанция / Время тренировки

где метаболический эквивалент (MET):

- Бег медленным темпом (около 8 км/ч) — 8;

- Бег в умеренном темпе (около 10 км/ч) — 10;

- Бег быстрым темпом (более 12 км/ч) — 12-15;

**F01.02.** Количество сожжённых калорий при плавании должно определяться по выражению:

Интенсивность = МЕТ = Стиль плавания

**Метаболический эквивалент (MET)**

- Брасс: 5

- Кроль на груди (умеренный темп): 6

- Кроль на груди (быстрый темп): 8

- На спине: 4.5

- Дельфин: 10

**F01.03.** Количество сожжённых калорий при жиме штанги должно определяться по выражению:

МЕТ зависит от процента отношения рабочего веса к максимальному весу:

MET = Рабочий вес / Максимальный вес 100 (%)

Если процент рабочего веса равен или превышает 80% или составляет менее 60%, уровень MET устанавливается 3, в противном случае устанавливается 5.

**F02.** В системе должен быть реализован список расчета количества калорий соженных при разных видах упражнений.

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

* Тип упражнения;
* Параметр в соответствии с типом упражнения.

**F03.** В Системе должна присутствовать функция добавления расчета в список.

**F04.** В Системе должна присутствовать функция удаления расчета из списка.

**F05.** В Системе должна присутствовать функция поиска расчета по параметрам, указанным в **F02.01**.

**F06.** В Системе должна присутствовать функция сохранения списка расчетов в файл (**D01**).

**F07.** В Системе должна присутствовать функция загрузки списка расчетов 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, 64 бит. 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.