**Министерство науки и высшего образования 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 \_\_\_\_\_\_\_\_\_\_\_ Соболев И.Г.

(Подпись)

\_\_\_\_\_\_\_\_\_\_\_\_\_

(Дата сдачи отчета)

Отчет принят:

Доцент, к.т.н. \_\_\_\_\_\_\_\_\_\_\_ Калентьев А.А.

(Подпись)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Дата проверки отчета)

Томск 2023

**СОДЕРЖАНИЕ**

[Введение 3](#_Toc155426518)

[1 UML-диаграмма вариантов использования 4](#_Toc155426519)

[2. UML-диаграмма классов 5](#_Toc155426520)

[3 Описание классов, образующих связь типа «общее-частное» 7](#_Toc155426521)

[4 Дерево ветвлений Git 10](#_Toc155426522)

[5 Тестирование программы 11](#_Toc155426523)

[5.1 Тестовый случай «Рассчитать калории для тренировки» 11](#_Toc155426524)

[5.2 Тестовый случай «Удалить тренировку» 13](#_Toc155426525)

[5.3 Тестовый случай «Найти тренировку» 15](#_Toc155426526)

[5.4 Тестовый случай «Сохранить файл» 17](#_Toc155426527)

[5.5 Тестовый случай «Загрузить файл» 19](#_Toc155426528)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 22](#_Toc155426529)

[ПРИЛОЖЕНИЕ А 23](#_Toc155426530)

[1 Общие сведения 24](#_Toc155426531)

[1.1 Полное наименование системы и её условное обозначение 24](#_Toc155426532)

[1.2 Наименование предприятий разработчика и заказчика системы 24](#_Toc155426533)

[1.3 Плановые сроки начала и окончания работы по созданию системы 24](#_Toc155426534)

[2 Назначение и цели создания системы 25](#_Toc155426535)

[2.1 Назначение системы 25](#_Toc155426536)

[2.2 Цели создания системы 25](#_Toc155426537)

[3 Характеристика объектов автоматизации 26](#_Toc155426538)

[4 Требования к системе 27](#_Toc155426539)

[4.1 Требования к архитектуре 27](#_Toc155426540)

[4.2 Требования к структуре данных 27](#_Toc155426541)

[4.3 Функциональные требования 27](#_Toc155426542)

[4.4 Требования к пользовательскому интерфейсу 28](#_Toc155426543)

[4.5 Требования к программному обеспечению 28](#_Toc155426544)

[4.6 Требования к аппаратному обеспечению 28](#_Toc155426545)

# Введение

Программная документация, представленная в данном комплекте, играет ключевую роль в поддержке и понимании созданного программного продукта. Назначение данной документации – обеспечить полное и структурированное описание программы, созданной в рамках лабораторной работы №4 по дисциплине: «Основы объектно-ориентированного программирования».

Среди требований к основной части данной документации представлены:

* UML-диаграмма вариантов использования дляразработанной программ
* UML-диаграмма классов;
* Для классов, образующих связь типа «общее-частное» (наследование, реализация) – описание;
* Дерево ветвлений Git;
* Результаты тестирования программы.

# 1 UML-диаграмма вариантов использования

Диаграмма вариантов использования в UML – диаграмма, отражающая отношения между акторами и прецедентами и являющаяся составной частью модели прецедентов.

Для разрабатываемой системы была разработана диаграмма вариантов использования и представлена на рисунке 1.

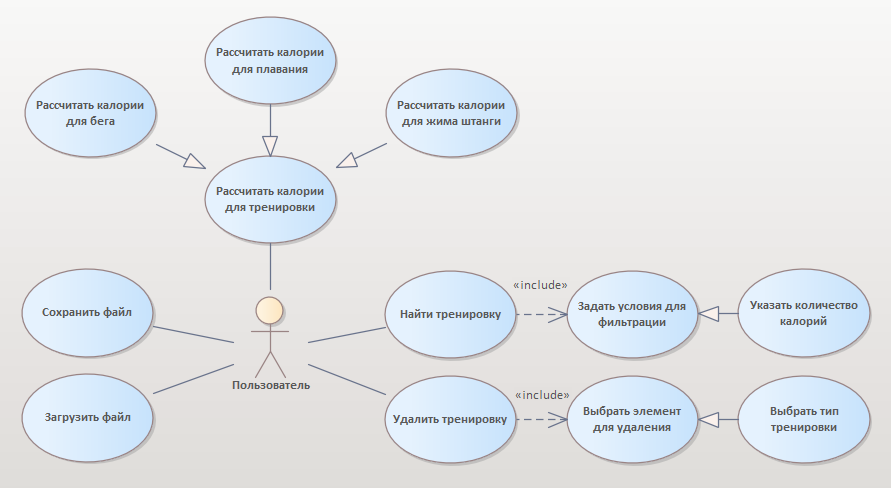


Рисунок 1 – UML-диаграмма вариантов использования

# 2. UML-диаграмма классов

Диаграмма классов – это структурная диаграмма в языке моделирования UML (Unified Modeling Language), предназначенная для визуализации структуры системы и отображения взаимосвязей между классами. Она представляет собой графическое изображение классов, интерфейсов и их взаимосвязей в программной системе.

Диаграмма классов предсатвлена на рисунке 2.

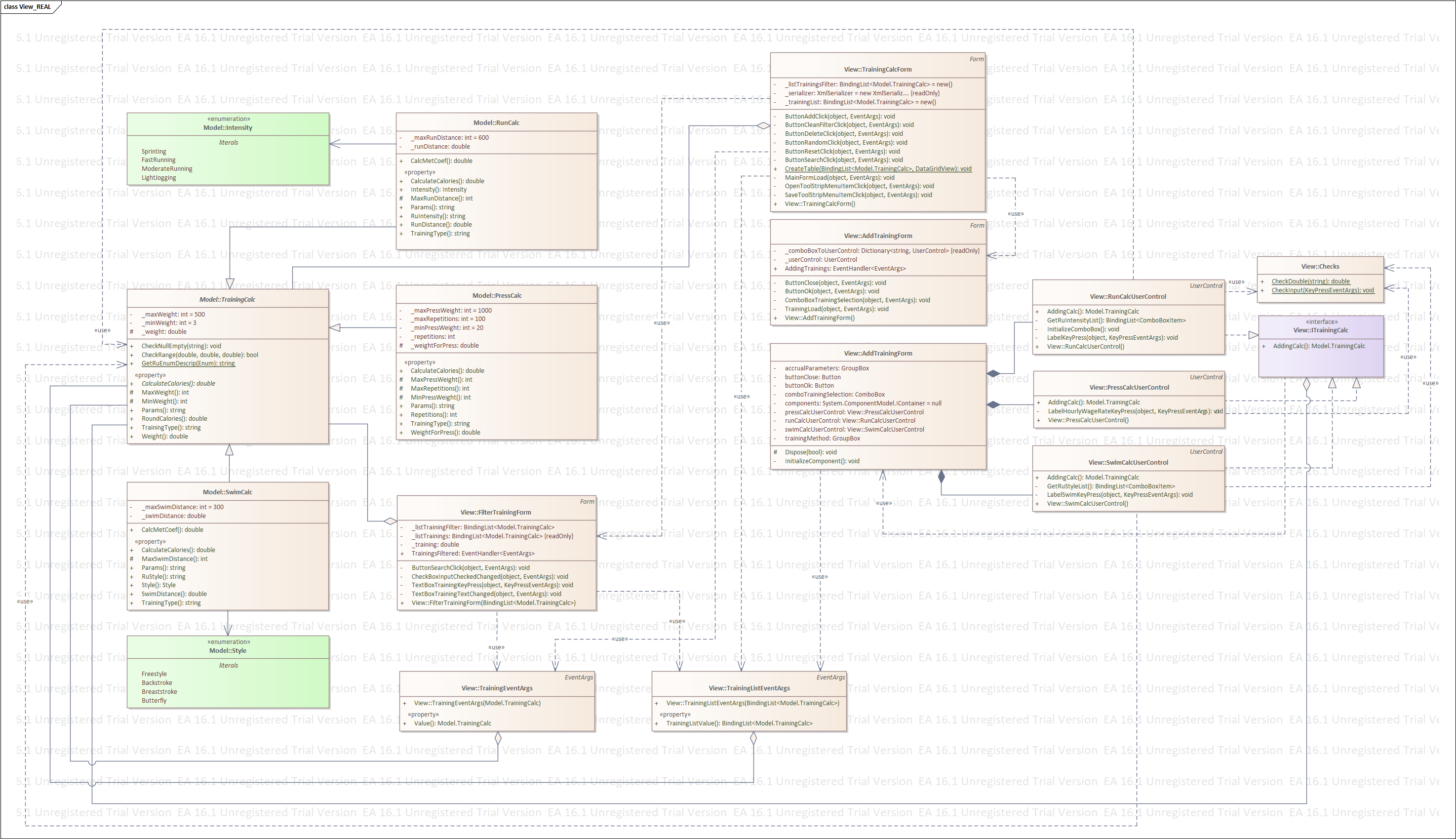


Рисунок 2 – UML-диаграмма классов

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

В таблице 1 приведено описание абстрактного класса «TrainingCalc» с его полями, свойствами и методами.

Таблица 1 – Описание класса «TrainingCalc»

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Описание класса | | |
| Класс TrainingCalc – абстрактный базовый класс для расчёта тренировок | | |
| Поля | | |
|  |  |  |
| – \_minWeight | int | Минимальный вес в 3 кг. |
| – \_maxWeight | int | Максимальный вес в 500 кг. |
| – \_weight | double | Вес в кг. |
| Свойства | | |
| + CalculateCalories | double | Количество затраченных калорий.  Абстрактное свойство, переопределяется  в производных классах. |
| # MaxWeight | int | Свойство для обращения к приватному полю \_maxWeight. |
| # MinWeight | int | Свойство для обращения к приватному полю \_minWeight. |
| + Params | string | Параметры тренировки.  Абстрактное свойство, переопределяется  в производных классах. |
| + RoundCalories | double | Округлённое количество калорий.  Абстрактное свойство, переопределяется  в производных классах. |
| + TrainingType | string | Тип тренировки.  Абстрактное свойство, переопределяется  в производных классах. |
| + Weight | double | Свойство для обращения к приватному полю \_weight. |
| Методы | | |
| + CheckNullEmpty | void | Метод проверки значения на Null- или Empty-значение. |
| + CheckRange | bool | Метод проверки числа на допустимый диапазон. |
| + CheckInput | double | Метод проверки введённого значения. |

В таблице 2 представлено описание класса «RunCalc».

Таблица 2 – Описание класса «RunCalc»

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Описание класса | | |
| Класс RunCalc – бег | | |
| Поля | | |
| – \_maxRunDistance | int = 600 | Максимальная дистанция бега. |
| –\_runDistance | double | Дистанция для пробежки. |
| Свойства | | |
| + CalculateCalories | double | Затраченные калории при беге. |
| + Intensity | Intensity | Интенсивность бега. |
| # MaxRunDistance | int | Метод для обращения к приватному полю \_maxRunDistance. |
| + Params | string | Параметры для расчёта тренировки. |
| + RuIntensity | string | Получение параметра интенсивности на русском языке. |
| + RunDistanse | double | Свойство доступа к пробегаемой дистанции. |
| + TrainingType | string | Тип тренировки - бег. |
| Методы | | |
| + CalcMetCoef | double | Коэффициент метаболизма. |

В таблице 3 представлено описание класса «SwimCalc».

Таблица 3 – Описание класса «SwimCalc»

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Описание класса | | |
| Класс SwimCalc – плавание | | |
| Поля | | |
| – \_maxSwimDistance | int = 300 | Максимальная дистанция плавания. |
| – \_swimDistance | double | Дистанция плавания. |
| Свойства | | |
| + CalculateCalories | double | Затраченные калории при беге. |
| # MaxSwimDistanse | double | Свойство для обращения к приватному полю \_maxSwimDistance. |
| + Style | Style | Стиль плавания. |
| + Params | string | Параметры для расчёта тренировки. |
| + RuStyle | string | Получение параметра стиля плавания на русском языке. |
| + SwimDistanse | double | Свойство доступа к проплываемой дистанции. |
| + TrainingType | string | Тип тренировки - плавание. |
| Методы | | |
| + CalcMetCoef | double | Коэффициент метаболизма. |

В таблице 4 представлено описание класса «PressCalc».

Таблица 4 – Описание класса «PressCalc»

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Описание класса | | |
| Класс PressCalc – жим штанги | | |
| Поля | | |
| –\_maxPressWeight | int = 500 | Максимальный поднимаемый вес: 1000 кг. |
| – \_maxRepetitions | int = 100 | Максимальное количество повторений. |
| –\_minPressWeight | int = 20 | Минимальный вес штанги (только гриф): 20 кг. |
| – \_repititions | int | Количество повторов |
| # \_weightForPress | double | Вес штанги (с учётом грифа) в килограммах. |
| Свойства | | |
| + CalculateCalories | double | Затраченные калории при жиме штанги. |
| # MaxPressWeight | int | Свойство для обращения к приватному полю \_maxPressWeight. |
| # MaxRepititions | int | Свойство для обращения к приватному полю \_maxRepetitions. |
| # MinPressWeight | int | Свойство для обращения к приватному полю \_minPressWeight. |
| + Params | string | Параметры для расчёта тренировки. |
| + Repititions | int | Свойство доступа к приватному полю \_repititions. |
| + TrainingType | string | Тип тренировки – жим штанги. |
| + WeightForPress | double | Поднимаемый вес. |

# 4 Дерево ветвлений Git

На рисунке 3 представлено дерево ветвлений Git, полученное по результатам работы над проектом.

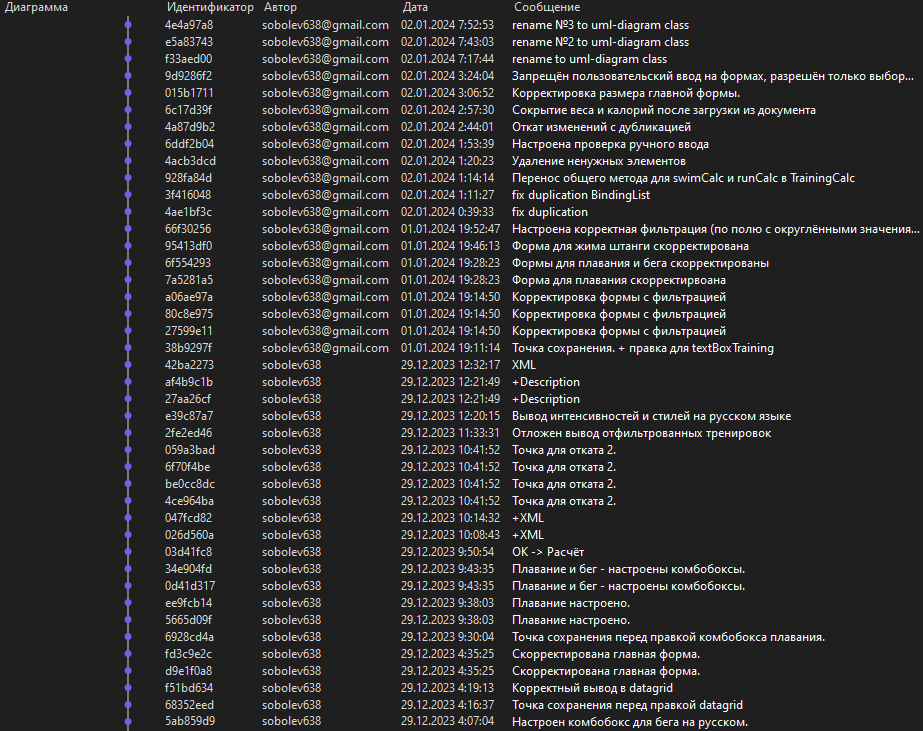


Рисунок 3 – Дерево ветвлений Git

# 5 Тестирование программы

Процесс функционального тестирования программы представлен ниже.

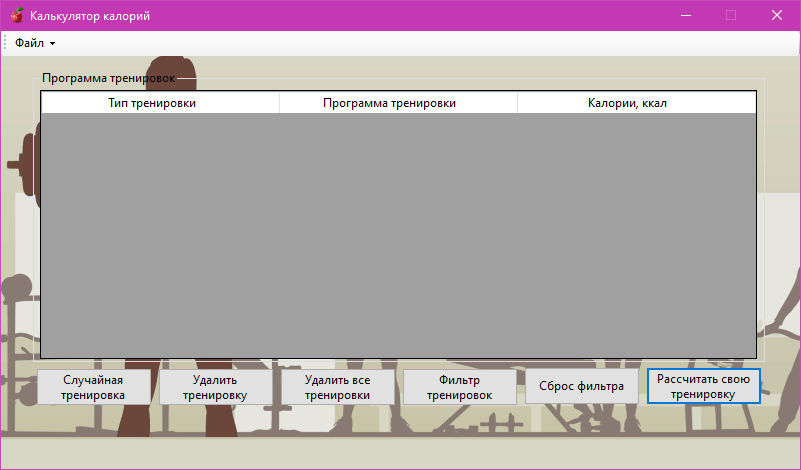


Рисунок 4 – Графический интерфейс пользователя

# 5.1 Тестовый случай «Рассчитать калории для тренировки»

Для добавления элемента необходимо вызвать соответствующую форму путём нажатия кнопки «Рассчитать свою тренировку» (рисунок 5).

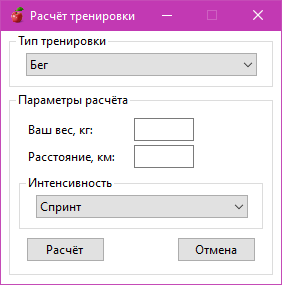


Рисунок 5 – Форма для добавления элемента

После ввода данных необходимо нажать кнопку «Расчёт», элемент появится в таблице главной формы (рисунки 6 и 7).

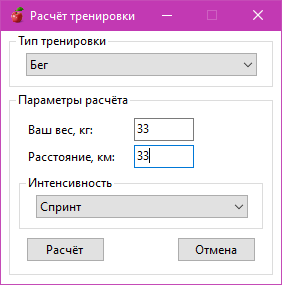


Рисунок 6 – Заполнение полей

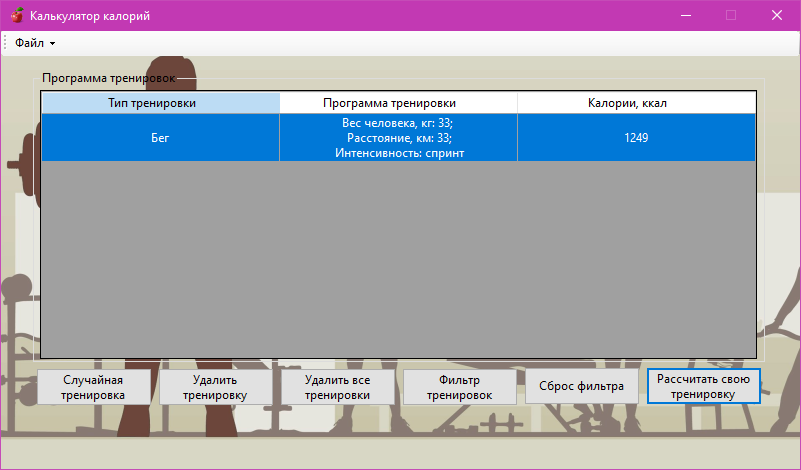


Рисунок 7 – Успешное добавление нового элемента

В программе предусмотрена система обработки некорректного ввода данных пользователем. Например, при попытке ввести число с несколькими запятыми появится соответствующее сообщение об ошибке, рисунки 8-9. Аналогичным образом обрабатывается попытка ввода NaN.

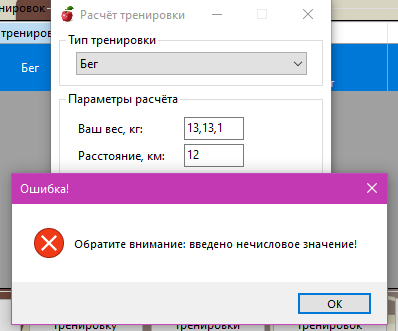


Рисунок 8 – Некорректный ввод и сообщение об ошибке

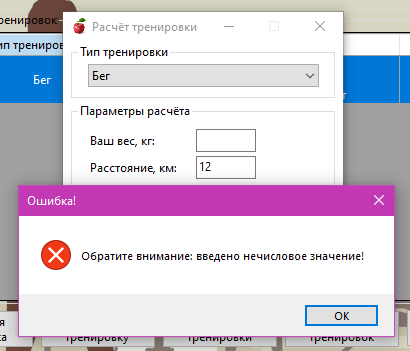


Рисунок 9 – Некорректный ввод и сообщение об ошибке

# 5.2 Тестовый случай «Удалить тренировку»

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

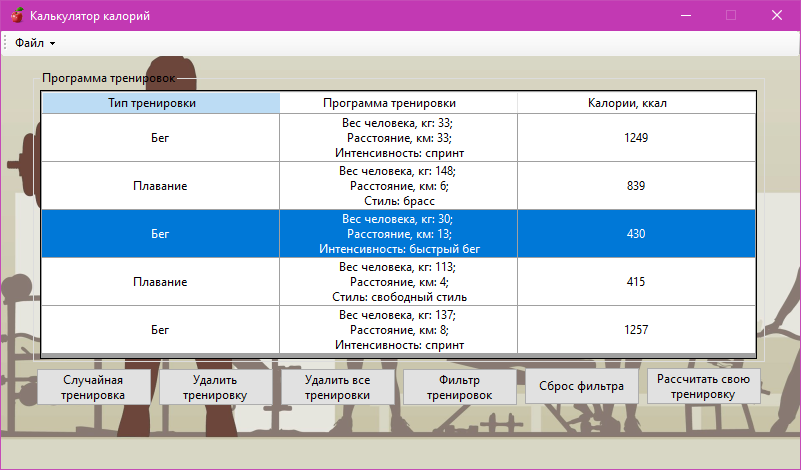


Рисунок 10 – Выбор элемента в таблице

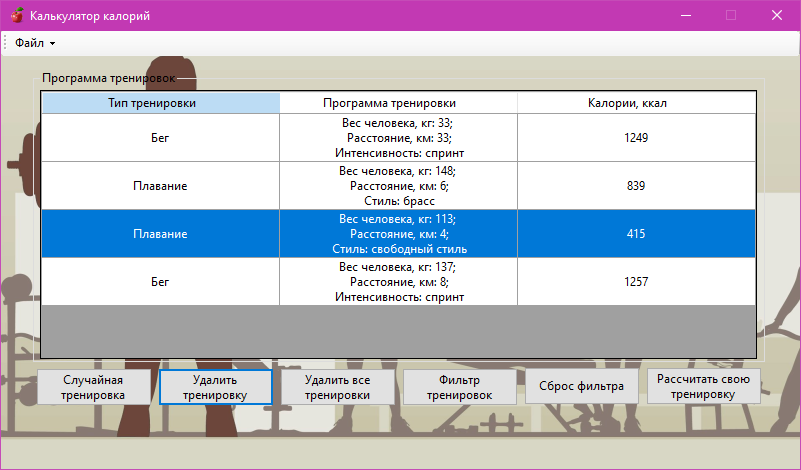


Рисунок 11 – Результат нажатия кнопки «Удалить тренировку»

# 5.3 Тестовый случай «Найти тренировку»

Для поиска элементов в списке необходимо нажать кнопку «Фильтр тренировок» (рисунок 12). Откроется соответствующая форма для поиска (рисунок 13).

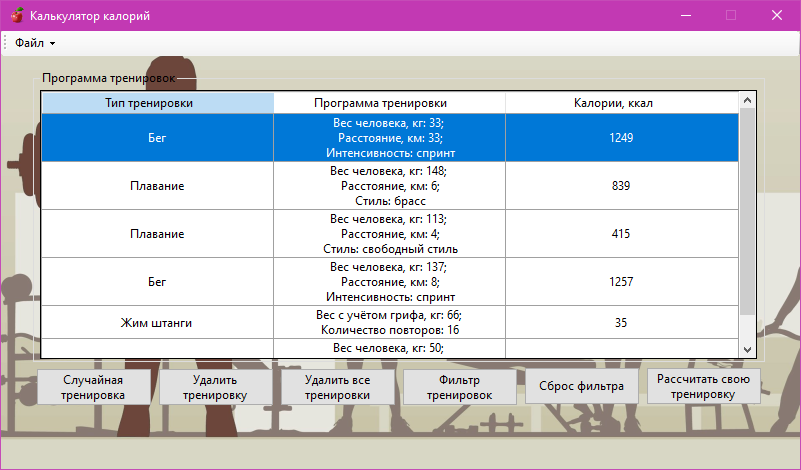


Рисунок 12 – Список рассчитанных тренировок

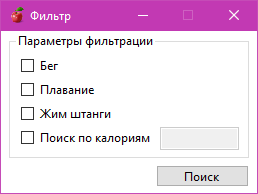


Рисунок 13 – Форма для поиска

Далее пользователь выбирает параметры, по которым требуется найти элемент, вводит значения этих параметров и нажимает кнопку «Поиск» (рисунки 14 и 15).

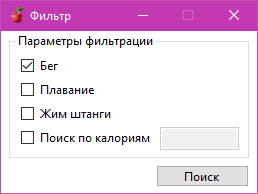


Рисунок 14 – Поиск элемента по названию

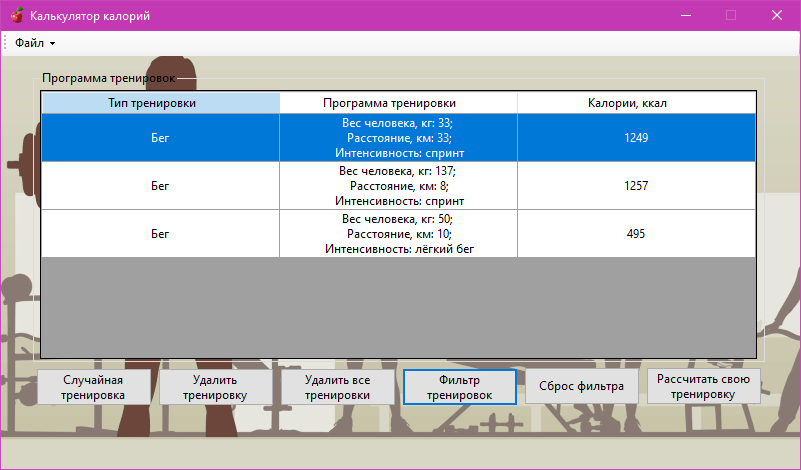


Рисунок 15 – Результат поиска элемента

Для того, чтобы сбросить фильтр, предусмотрена соответствующая кнопка «Сброс фильтра» (рисунок 16).

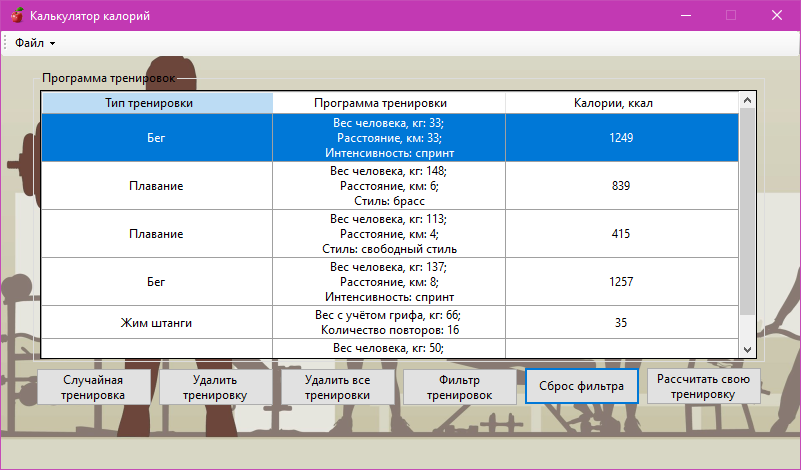


Рисунок 16 – Форма после сброса фильтра тренировок

# 5.4 Тестовый случай «Сохранить файл»

Для сохранения данных в таблице необходимо нажать на панели «Файл\Сохранить», рисунок 17. Откроется системный диалог сохранения файла, где пользователь выбирает директорию и указывает имя файла (рисунок 18).

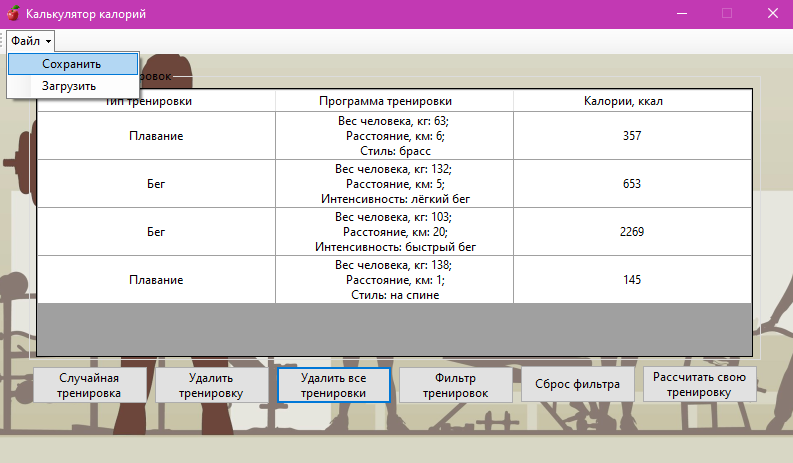


Рисунок 17 – Сохранение данных

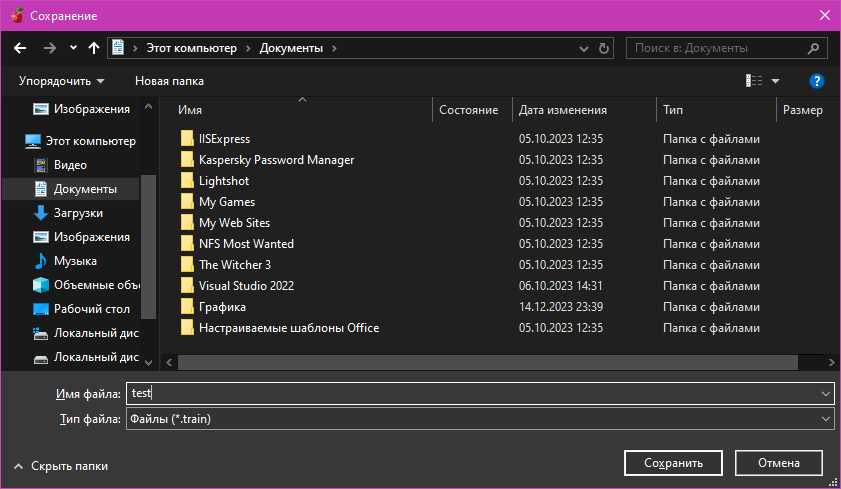


Рисунок 18 – Сохранение файла

После сохранения данных в файл появится соответствующее сообщение, рисунок 19.

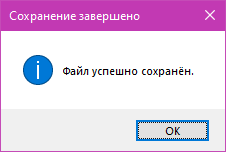


Рисунок 19 – Сообщение о сохранении файла

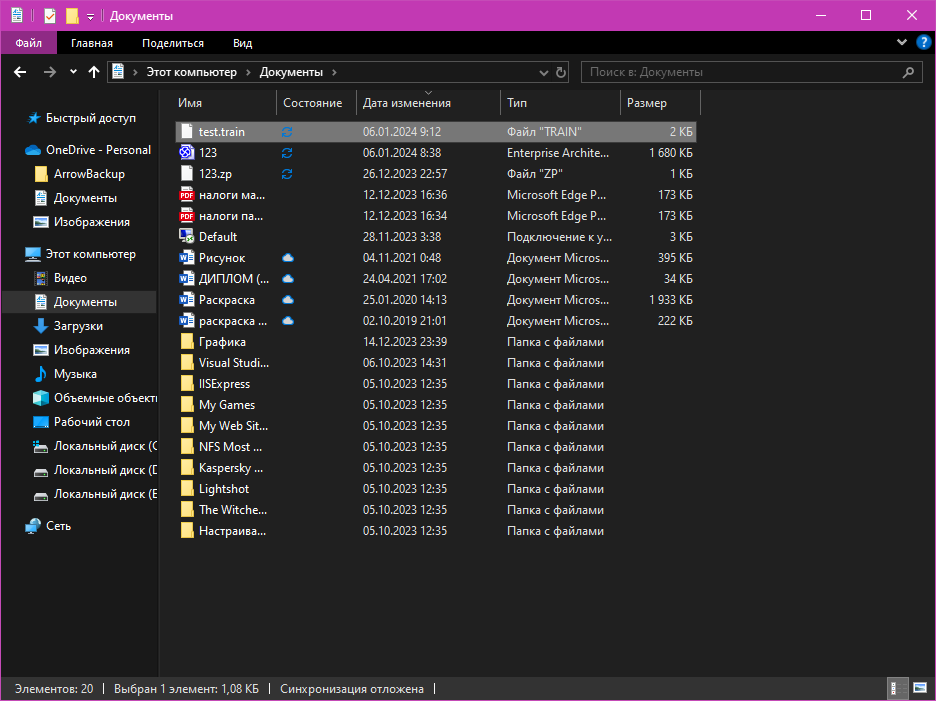


Рисунок 20 – Результат сохранения файла

Далее пользователь выбирает параметры, по которым требуется найти элемент, вводит значения этих параметров и нажимает кнопку «Поиск» (рисунки 14 и 15).

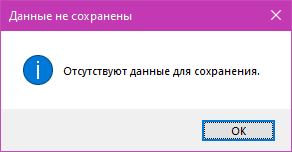


Рисунок 21 – Результат попытки сохранения при отсутствии элементов

# 5.5 Тестовый случай «Загрузить файл»

Для сохранения данных в таблице необходимо нажать на панели «Файл\Загрузить», рисунок 22. Откроется системный диалог открытия файла, где пользователь выбирает директорию и указывает имя файла, рисунок 23.

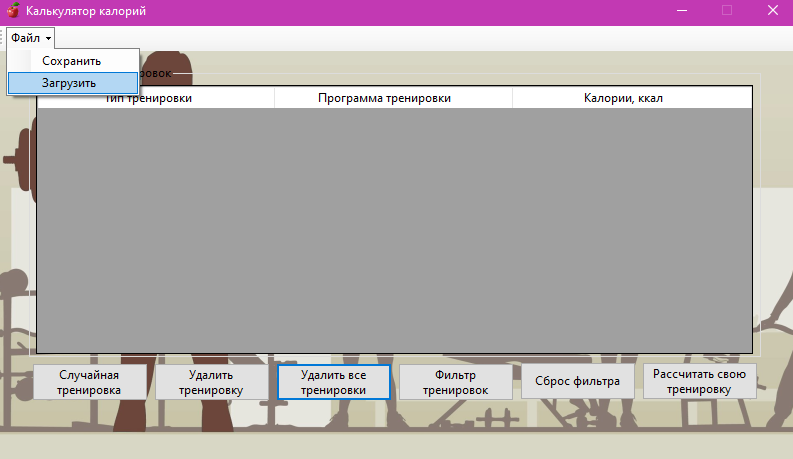
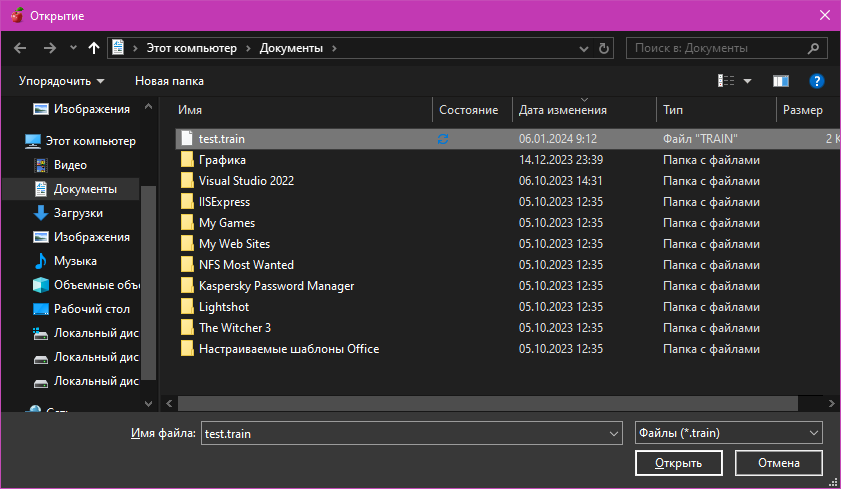
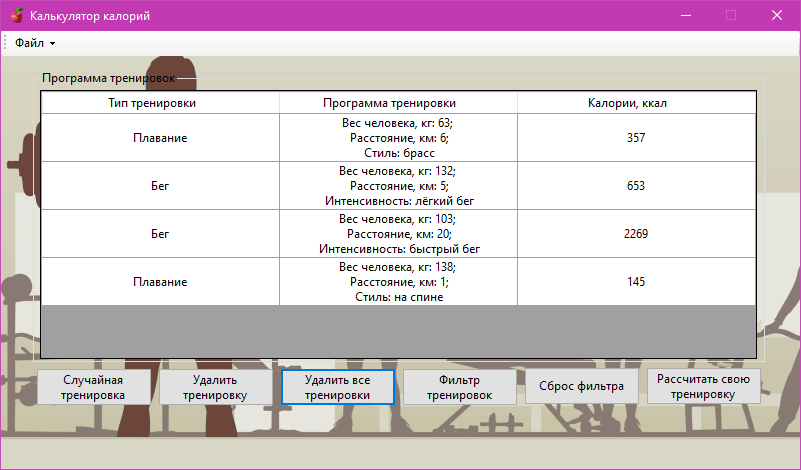


Рисунок 22 – Загрузить данные





# 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 затрачиваемых калорий

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

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

Томск 2023

# 1 Общие сведения

# 1.1 Полное наименование системы и её условное обозначение

Полное наименование: «Калькулятор 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-5КМ11 НИ ТПУ Соболев И.Г.

# 1.3 Плановые сроки начала и окончания работы по созданию системы

Начало работ: 1 мая 2023 г.

Окончание работ: 20 декабря 2023 г.

# 2 Назначение и цели создания системы

# 2.1 Назначение системы

Система предназначена для расчёта затраченный калорий при различных типах тренировок: бег, плавание и жим штанги.

# 2.2 Цели создания системы

Система создаётся в целях сокращения трудозатрат тренеров при расчётах затрачиваемых калорий.

# 3 Характеристика объектов автоматизации

Вычисление затраченных калорий может пригодиться:

1. Фитнес и здоровье:

1.1. Самостоятельные тренировки: Люди, занимающиеся фитнесом индивидуально, могут использовать программу для отслеживания своих тренировок и оценки затрат энергии.

1.2. Фитнес-приложения: Тренеры и пользователи фитнес-приложений могут интегрировать такую программу для предоставления более точной информации о затрате калорий.

2. Спортивные мероприятия:

2.1. Тренировки спортсменов: Тренеры и спортсмены могут использовать программу для анализа затрат калорий во время тренировок и адаптации планов тренировок для достижения определенных целей.

2.2. Мониторинг физической активности: В спортивных мероприятиях, таких как марафоны или плавание, программа может использоваться для отслеживания физической активности участников.

3. Врачебные назначения:

3.1. Лечение ожирения: Врачи могут рекомендовать пациентам, страдающим ожирением, использование подобных программ для контроля за затратами калорий и разработки персонализированных планов по снижению веса.

3.2. Физическая реабилитация: Врачебные программы для восстановления после травм или хирургических вмешательств могут включать мониторинг физической активности и затрат калорий.

Поскольку расчёты затрачиваемых калорий выполняются специалистами при формировании планировании программ тренировок, представляется целесообразным автоматизировать этот процесс.

# 4 Требования к системе

Таблица 4.1 – Префиксы мнемонических идентификаторов требований и их расшифровка

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

# 4.1 Требования к архитектуре

**А01.** Система должна быть реализована в виде настольного приложения.

# 4.2 Требования к структуре данных

**D01.** Данные о параметрах заработной платы должны храниться в XML-файле с расширением \*.train.

# 4.3 Функциональные требования

**F01.** Система должна рассчитывать затрачиваемые калории для следующих типов тренировок:

* бег;
* плавание;
* жим штанги.

**F02.** В системе должен быть реализован вывод списка тренировок.

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

* тип тренировки;
* программа тренировки
* калории, ккал.

**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:

* Тип тренировки: бег, плавание, жим штанги;
* Калории.

**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, 64-bit. Работоспособность на других выпусках и версиях не гарантируется.

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

# 4.6 Требования к аппаратному обеспечению

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

**C04.** ОЗУ – не менее 4 ГБ.