

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение
высшего образования

«КРЫМСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ им. В. И. ВЕРНАДСКОГО»
ФИЗИКО-ТЕХНИЧЕСКИЙ ИНСТИТУТ

Кафедра компьютерной инженерии и моделирования

**ПРИЛОЖЕНИЕ ПО РАСЧЕТУ ЭНЕРГЕТИЧЕСКОЙ ЦЕННОСТИ
ПРОДУКТА**

Курсовая работа

по дисциплине «Программирование»

студента 1 курса группы ИВТ-б-о-191

Чушева Сергея Александровича

направления подготовки 09.03.01 «Информатика и вычислительная техника»

Научный руководитель

старший преподаватель кафедры

компьютерной инженерии и моделирования

(оценка)

(подпись, дата)

Чабанов В.В.

Симферополь, 2020

РЕФЕРАТ

Разработка приложения по расчету энергетической ценности продукта. - Симферополь: ФТИ КФУ им. Вернадского, 2020.

Курсовой проект выполнен на 38 листах, состоит из 4-х глав, в проекте 11 иллюстраций, 2 приложения и 22 литературных источника.

Объект исследования: информационные технологии, культура питания.

Предмет исследования: калькулятор подсчета калорий, белков, жиров и углеводов (Далее КБЖУ), энергетическая ценность продуктов.

Целью курсовой работы является разработка программы расчета энергетической ценности продуктов. Программа производит расчет КБЖУ на дневной или разовый прием пищи. Для разработки программы был использован основной язык программирования C++ и кроссплатформенный фреймворк Qt. C++ - компилируемый язык программирования общего назначения, сочетает свойства как высокоуровневых, так и низкоуровневых языков программирования.

В связи с целью курсовой работы, были поставлены следующие **задачи**:

- Изучить тему правильного питания
- Изучить энергетическую ценность продуктов
- Изучить теоретические аспекты основного языка программирования C++
- Изучение кроссплатформенного фреймворка Qt

Создать программу расчёта КБЖУ на дневной или разовый прием пищи с учетом граммов продукта на языке программирования C++

В результате создана программа расчета энергетической ценности продуктов. Программа производит расчет КБЖУ на дневной или разовый прием пищи. Для разработки программы был использован основной язык программирования C++ и кроссплатформенный фреймворк Qt. C++ - компилируемый язык программирования общего назначения, сочетает свойства как высокоуровневых, так и низкоуровневых языков программирования.

Выполнены следующие **задачи**:

- Изучена тема правильного питания

- Изучена энергетическую ценность продуктов
- Изучена теоретические аспекты основного языка программирования C++
- Изучен кроссплатформенный фреймворк Qt

Перспективы развития приложения:

- Разместить приложение на Android и IOS устройства;
- Расширить ассортимент продуктов;
- Добавить готовые блюда и напитки;
- Добавить встроенные платные покупки;
- Установить напоминание о приеме пищи для контроля спортивного питания.

Добавление предложенного функционала сделает приложение более удобным при использовании мобильного телефона. Расширенный функционал и выбор продуктов даст возможность более корректно составить меню и рассчитать энергетическую ценность продуктов питания и напитков, скорректировать прием пищи при спортивном питании.

Для перспективы монетизации бесплатного приложения необходимо встроить платные дополнения. В качестве платного дополнения может выступать разработка комплекса физических упражнений с различными целями: поддержка массы тела, сушка тела, увеличение мышечной массы тела. Платное дополнение предлагает комплексный подход: расчёт продуктов питания, рекомендуемые физические упражнения, напоминания о приемах пищи и физической нагрузке с ежемесячным отчетом результата изменения массы тела.

Приложение может быть использовано фитнес тренерами для составления программ по питанию, контроля веса, носить рекомендационный характер.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	5
ГЛАВА 1 ПОСТАНОВКА ЗАДАЧИ.....	6
1.1 Цель проекта.....	6
1.1.1 Общая характеристика рационального питания.....	6
1.1.2 Энергетическая ценность продуктов. КБЖУ.....	7
1.2 Существующие аналоги	9
1.3 Основные отличия от аналогов	10
1.4 Техническое задание.....	10
ГЛАВА 2 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ПРИЛОЖЕНИЯ	11
2.1 Анализ инструментальных средств.....	11
2.2 Описание алгоритмов	12
2.3 Описание структур данных	14
2.4 Описание основных модулей.....	14
ГЛАВА 3 ТЕСТИРОВАНИЕ ПРОГРАММЫ	19
3.1 Тестирование исходного кода.....	19
3.2 Тестирование интерфейса пользователя и юзабилити.....	23
ГЛАВА 4 ПЕРСПЕКТИВЫ ДАЛЬНЕЙШЕГО РАЗВИТИЯ ПРОЕКТА.....	24
4.1 Перспективы технического развития.....	24
4.2 Перспективы монетизации.....	24
ЗАКЛЮЧЕНИЕ	25
ЛИТЕРАТУРА	26
ПРИЛОЖЕНИЕ 1 КОД ОСНОВНЫХ МОДУЛЕЙ ПРОЕКТА.....	28
ПРИЛОЖЕНИЕ 2 КОД ТЕСТИРУЮЩЕГО ПРОЕКТА	37

ВВЕДЕНИЕ

Программирование - одновременно наука и искусство создания компьютерных программ и программного обеспечения с помощью языков программирования. Язык программирования - формальная знаковая система, предназначенная для записи компьютерных программ. Программирование сочетает в себе элементы искусства, фундаментальных наук (прежде всего информатика и математика), связана со всеми сферами науки и жизни человека, в том числе с культурой правильного питания.

Объект исследования: информационные технологии, культура питания.

Предмет исследования: калькулятор подсчета калорий, белков, жиров и углеводов (Далее КБЖУ), энергетическая ценность продуктов.

Целью курсовой работы является разработка программы расчета энергетической ценности продуктов. Программа производит расчет КБЖУ на дневной или разовый прием пищи. Для разработки программы был использован основной язык программирования C++ и кроссплатформенный фреймворк Qt. C++ - компилируемый язык программирования общего назначения, сочетает свойства как высокоуровневых, так и низкоуровневых языков программирования.

В связи с целью курсовой работы, были поставлены следующие **задачи**:

- Изучить тему правильного питания
- Изучить энергетическую ценность продуктов
- Изучить теоретические аспекты основного языка программирования C++
- Изучение кроссплатформенного фреймворка Qt
- Создать программу расчёта КБЖУ на дневной или разовый прием пищи с учетом граммов продукта на языке программирования C++

ГЛАВА 1

ПОСТАНОВКА ЗАДАЧИ

1.1 Цель проекта

При создании приложения по контролю энергетической ценности продукта, первоначально необходимо изучить тему правильного питания: влияние на организм калорий, белков, жиров и углеводов. При изучении влияния КБЖУ на организм человека, можно научиться контролировать количество питательных веществ. С помощью контроля КБЖУ станет возможно регулировать или поддерживать массу тела человека, а также его физическое состояние и здоровье.

В связи с этим, была поставлена цель: разработать программу расчета энергетической ценности продуктов. Программа производит расчет КБЖУ на дневной или разовый прием пищи.

Из цели вытекают следующие задачи, первая из которых – изучить тему правильного питания, а также энергетическую ценность продуктов. После изучения темы питания, необходимо обратиться к теоретическим аспектам основного языка программирования C++, а также кроссплатформенного фреймворка Qt.

1.1.1 Общая характеристика рационального питания

Питание является жизненной необходимостью человека. «В настоящее время заметно возрастает понимание того, что пища оказывает на человека значительное влияние. Она даёт энергию, силу, развитие, а при грамотном её употреблении - и здоровье. Можно с определённой уверенностью утверждать, что здоровье человека на 70% зависит от питания.» [20, с.35]

Расход энергии человека на выполненную в течение дня работу зависит от характера производственной деятельности, режима труда и отдыха, объема домашней работы, особенностей использования свободного от труда времени.

Физическое напряжение характеризуется значительным повышением энерготрат по сравнению с умственной работой.

«Рациональное питание должно основываться на теории сбалансированного питания и предусматривать правильный режим потребления пищи. Необходимо знать и соблюдать три принципа рационального питания: умеренность, разнообразие, режим приема пищи.» [11, с.43]

Суточные энерготраты человека зависят от возраста, пола, массы тела, характера трудовой деятельности, климатических условий и индивидуальных особенностей протекания реакций обмена веществ в организме.

«Восполнение всех энерготрат человека осуществляется благодаря питанию. Зная химический состав пищи, легко подсчитать ее энергетическую ценность.» [14, с.45]

Энергетическая ценность рациона питания зависит от входящих в его состав белков, жиров и углеводов.

Оптимальным в рационе практически здорового человека является соотношение белков, жиров и углеводов. Это соотношение наиболее благоприятно для максимального удовлетворения как пластических, так и энергетических потребностей организма человека

1.1.2 Энергетическая ценность продуктов. КБЖУ.

Из-за недостатка физической активности современный человек все больше страдает от проблем с фигурой. Одни постоянно пытаются похудеть, мучая себя различными диетами, другие, наоборот, хотят набрать мышечную массу. Но помочь во всем этом может расчет БЖУ.

«КБЖУ — это общепринятое сокращение от сочетания "калории, белки, жиры, углеводы".» [17, с.97]

Вода играет важную роль в организме человека. Продукты питания с большим содержанием воды обладают меньшей калорийностью и менее способствуют ожирению. К таким продуктам относятся овощи, фрукты. Чем

больше воды в овощах, тем ниже их калорийность. Это касается и круп, и хлеба, и бобовых, и других продуктов растительного происхождения.

В совокупности эти вещества называются нутриентами.

«Нутриенты - основные составляющие нашего организма. От них во многом зависит и внешний вид, и здоровье человека. Каждый из нутриентов обладает своими свойствами и ролью в нашей жизни.» [11, с.74]

Рассмотрим их подробнее:

- Калорийность продуктов питания определяется их энергетической ценностью, которую получает организм при усвоении пищи. Нашим организмам для поддержания жизнедеятельности необходима температура около 37°C. Сжигаемое «топливо», то есть усваиваемая пища, позволяет телу продуцировать энергию для температурного баланса, осуществления обменных процессов. Объем энергии, перераспределяемой из продукта в тело, определяется количеством калорий в съеденном продукте или блюде.
- Белки формируют мышцы, поддерживают иммунитет, а также способствуют росту клеток. Белки содержатся в яйцах, мясе, сыре, рыбе, бобовых и других продуктах. Однако в каждом случае они сопровождаются разными вспомогательными веществами, поэтому не рекомендуется получать белки, например, только из куриного мяса. Кроме того, нужно следить за тем, чтобы в рационе присутствовали и животные, и растительные белки. Хотя последние содержат меньше незаменимых аминокислот, они обладают антисклеротическим действием, то есть нормализуют работу сосудов.
- Жиры. Являются второстепенными, но не менее важными источниками энергии, повышают эластичность стенок сосудов, в результате чего ткани получают больше питательных веществ из крови. Их недостаток в организме приводит к заболеваниям печени и почек, задержке жидкости и проблемам с кожей. Именно жиры влияют на уровень холестерина в нашей крови, поэтому их избыток часто становится причиной атеросклероза.

Содержатся жиры в растительной (подсолнечное, кукурузное и другие масла) и животной (мясо, рыба, сливочное масло) пище. Рекомендуемое соотношение растительных и животных жиров в рационе - 3:7.

- Углеводы. Это первостепенный источник энергии. Диетологами принято делить углеводы на простые (быстрые) и сложные (медленные). К первой группе относятся моносахариды и дисахариды. Самый распространенный пример - обычный столовый сахар, но простые углеводы также содержатся в фруктах и сладостях. Быстрыми они называются из-за скорости расщепления и усвоения в организме. Их функции - регуляция уровня сахара в крови, питание мышц и мозга. А теперь о сложных углеводах: они содержатся в макаронах, крупах, овощах и белом хлебе, долго расщепляются, благодаря чему насыщают организм энергией и практически не влияют на уровень сахара. Избыточные углеводы способны переходить в разряд жиров, из-за чего опять же растет уровень холестерина и развивается атеросклероз. Исключение углеводов из рациона - одна из самых популярных диет: при их недостатке в качестве источника энергии организм станет использовать жировые запасы.

[19, 6, 11, 14]

Чтобы нутриенты правильно выполняли все вышеперечисленные функции, они должны поступать в нужных количествах и пропорциях. Это и является целью расчета КБЖУ - узнать, сколько нутриентов каждого типа нужно употреблять для достижения той или иной цели.

1.2 Существующие аналоги

YAZIO — это виртуальный счетчик калорий, в базе которого содержится более 1000 блюд и продуктов. В приложение внесены рецепты диетических закусок.

Lifesum - приложение синхронизируется с Google Fit, информацию о пользователе вводить необязательно. Программа считает количество км во время

пробежки, сожженные и употребленные калории, показывает статистику похудения.

FatSecret - одно из самых популярных приложений. В приложении информация о БЖУ и водном балансе, рецепты ПП блюд, рекомендации худеющим, статистика весам, история прогресса.

1.3 Основные отличия от аналогов

Основным отличием от аналогов является: кроссплатформенность, уникальный дизайн, а также доступная, бесплатная версия программы, без встроенных покупок и рекламы.

1.4 Техническое задание

Создать приложение для контроля энергетической ценности продукта. Программа должна производит расчет КБЖУ на сутки. Необходимо добавить возможность регистрировать пользователей с индивидуальными рекомендациями по питанию. Добавить в программу индивидуальный дизайн и удобный интерфейс.

ГЛАВА 2

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ПРИЛОЖЕНИЯ

2.1 Анализ инструментальных средств

Программа контроля энергетической ценности продуктов питания с указанием продукта в граммах и полным расчетом КБЖУ на введенные данные разработана с помощью языка C++

«C++ - язык программирования, который поддерживает такие парадигмы программирования как процедурное программирование, объектно-ориентированное программирование, обобщенное программирование, обеспечивает модульность, отдельную компиляцию, обработку исключений, абстракцию данных, объявление типов (классов) объектов, виртуальные функции. Стандартная библиотека включает, в том числе, общеупотребительные контейнеры и алгоритмы. C++ сочетает свойства как высокоуровневых, так и низкоуровневых языков.» [8, с.74]

Интерфейс приложения был разработан с помощью кроссплатформенной свободной, интегрированной среды разработки – Qt Creator.

«Основная задача Qt Creator — упростить разработку приложения с помощью фреймворка Qt на разных платформах.» [15, с.45]

Поэтому среди возможностей, присущих любой среде разработки, есть и специфичные, такие как отладка приложений на QML и отображение в отладчике данных из контейнеров Qt, встроенный дизайнер интерфейсов: как на QML, так и на QtWidgets.

Для создания пользовательского интерфейса был использован модуль Qt GUI. «Модуль подключает базовые классы для компонентов графического пользовательского интерфейса, так же включает в себя OpenGL, что позволяет переносить программу на практически на любые платформы, получая при этом одинаковый результат, будь это графическая станция или суперкомпьютер.» [21, с.78]

2.2 Описание алгоритмов

После открытия программы происходит регистрация пользователя. Для регистрации необходимо указать следующие данные:

- Имя пользователя – имя учетной записи, в которой будут храниться результаты подсчетов. В качестве вводных данных используются цифры и буквы, а также спецсимволы. Пробелы запрещены.
- Возраст – допустим ввод не более двух числовых символов.
- Пол – от данного параметра будет зависеть по какой формуле будет рассчитан конечный результат.
- Вес и рост – вес указывается в килограммах, рост в сантиметрах. Невозможен ввод отрицательных чисел, нуля.
- Степень активности – указывается степень физической активности пользователя.
- Цель – выбор желаемого результата.

Для решения поставленной задачи использован алгоритм.

Рисунок 2.1.1

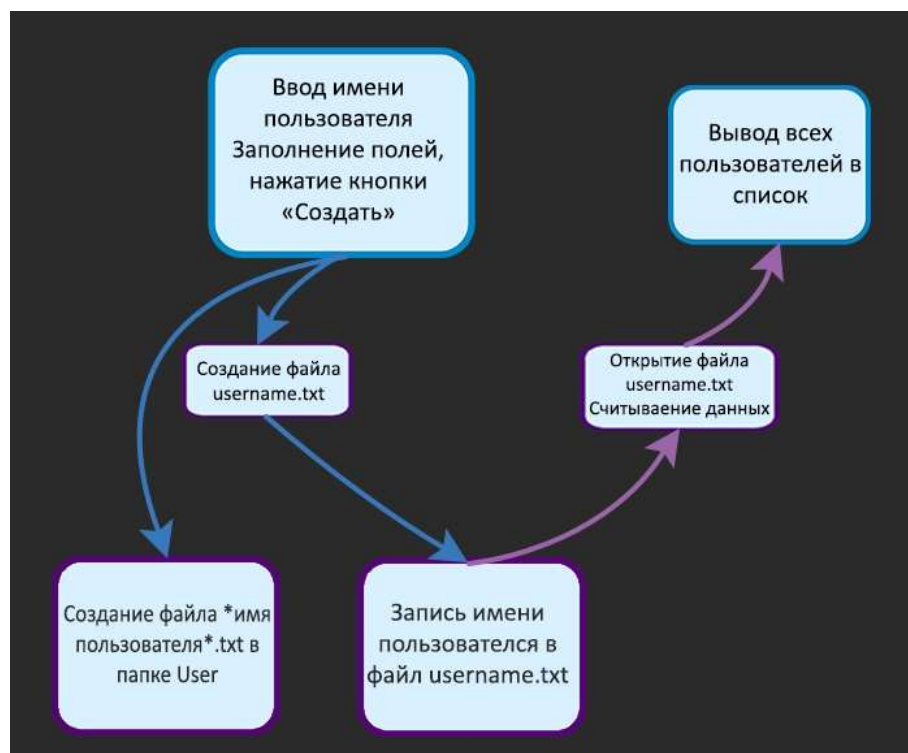


Рис 2.1.1 Алгоритм регистрации пользователя

Удаление учетной записи.

Для решения данной задачи использован алгоритм, рисунок 2.1.2

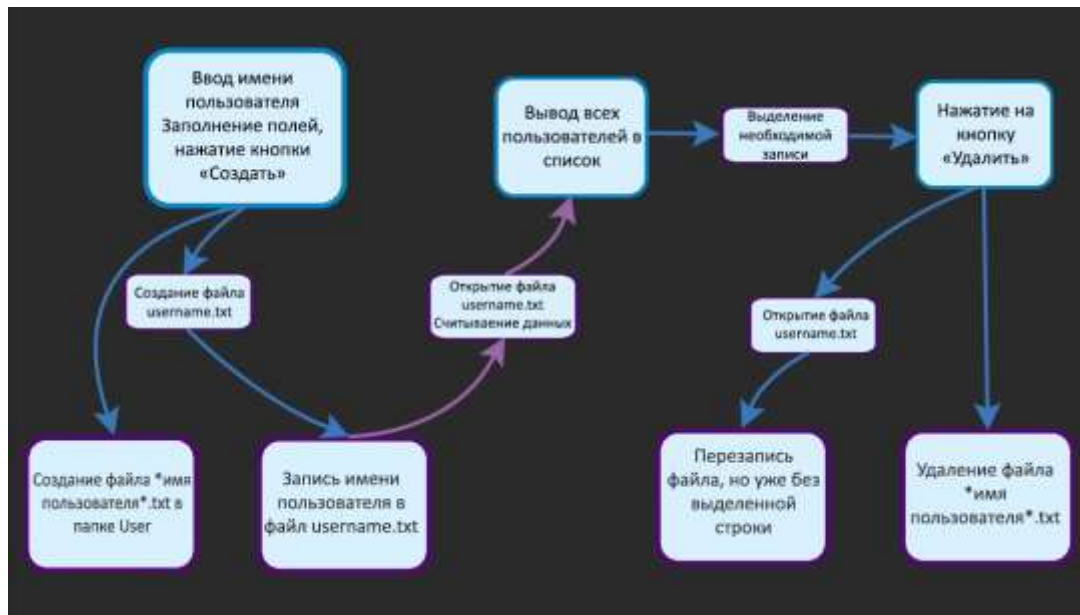


Рис. 2.1.2 Алгоритм удаления пользователя

Для выбора пользователя использован алгоритм, предоставленный на рисунке 2.1.3

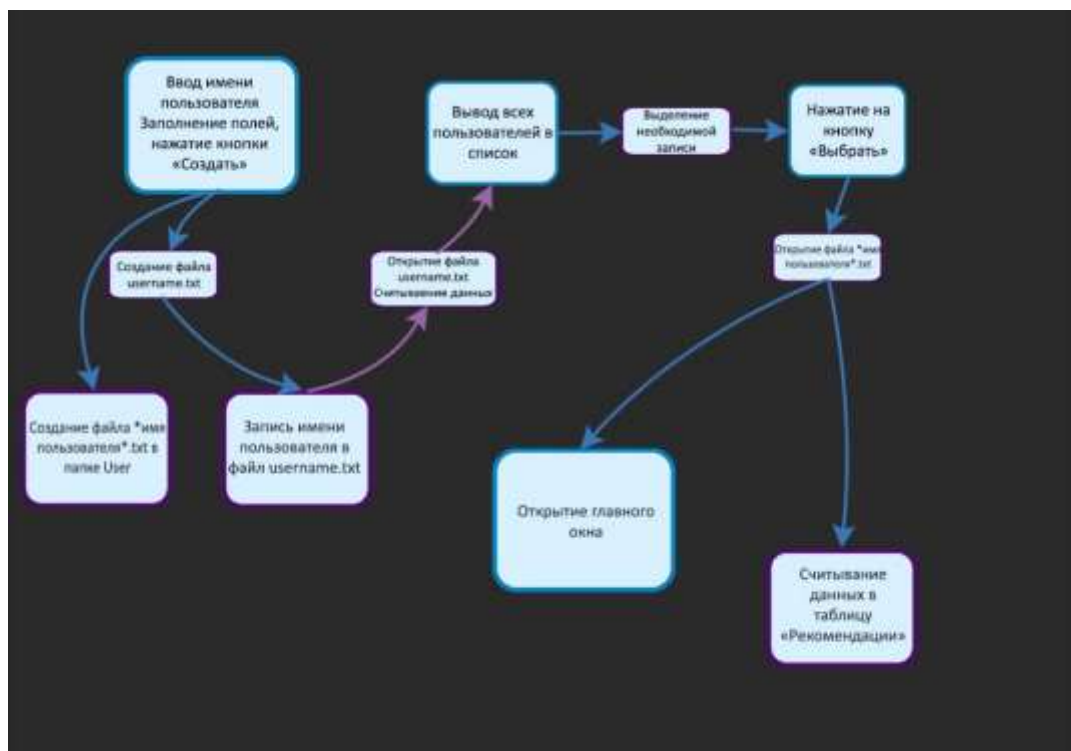


Рис. 2.1.3 Алгоритм выбора пользователя

Для работы с основным окном используем алгоритм, находящийся на рисунке 2.1.4

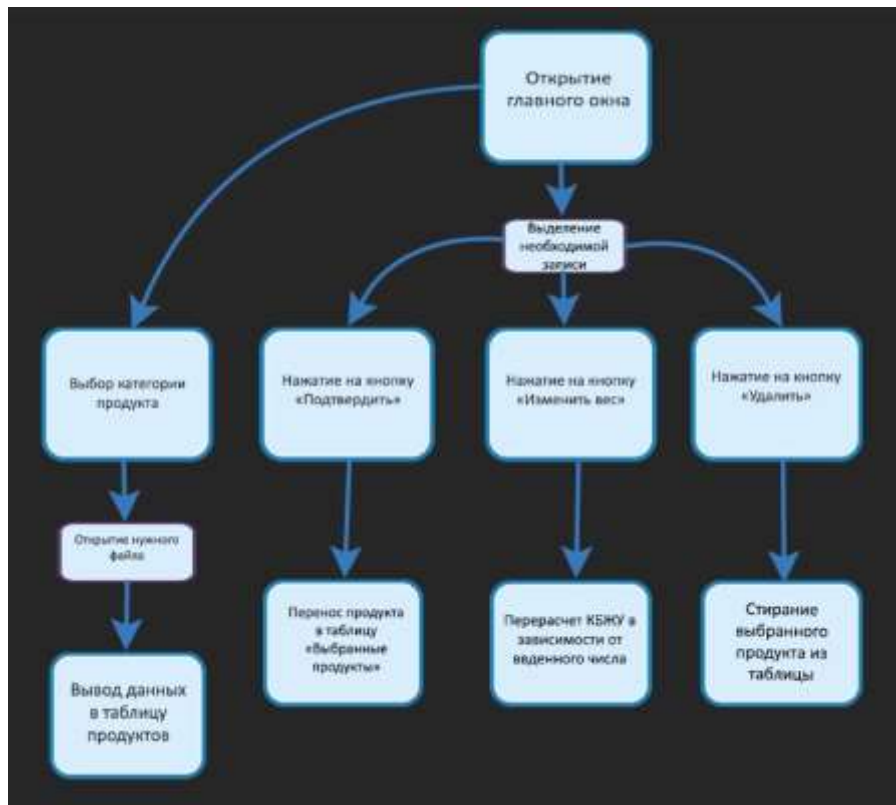


Рис. 2.1.4 Алгоритм работы с основным окном

2.3 Описание структур данных

Для хранения и передачи данных о продукте и пользователях использован формат TXT.

2.4 Описание основных модулей

Модуль QtCore

Описание

Модуль QtCore содержит ядро функциональности, не касающейся GUI.

Все другие модули Qt опираются на этот модуль. Модуль QtCore является частью всех версий Qt. [15,21]

Модуль QtGui

Описание

Модуль QtGui дополняет QtCore функциональностью GUI.

Модуль QtGui является частью Выпуска Qt GUI Framework, Выпуска Qt Full Framework и Версий Open Source Qt. [15,21]

Модуль QWidget

Класс QWidget является базовым для всех объектов пользовательского интерфейса. Виджет — это элементарный объект пользовательского интерфейса: он получает события мыши, клавиатуры и другие события от оконной системы и рисует свое изображение на экране. Каждый виджет имеет прямоугольную форму, и все они отсортированы в порядке наложения. Виджет ограничен своим родителем и другими виджетами, расположенными перед ним.

Виджет, не встроенный в родительский виджет, называется окном. Обычно окно имеет рамку и полосу с заголовком, хотя, используя соответствующие флаги окна можно создать окно без такого внешнего оформления. В Qt QMainWindow и разнообразные подклассы QDialog являются наиболее распространёнными типами окон. [15,21]

Модуль QTableWidgetItem

Описание

Класс QTableWidgetItem предоставляет табличное представление на основе элементов с моделью по умолчанию. наследует QTableWidgetItemView. [15,21]

Виджеты таблиц предоставляют стандартные средства отображения таблиц для приложений. Элементы в QTableWidgetItem предоставляются QTableWidgetItemItem.

Если вам нужна таблица, которая использует вашу собственную модель данных, вы должны использовать QTableWidgetItemView, а не этот класс. [15,21]

Табличные виджеты могут быть построены с необходимым количеством строк и столбцов:

Элементы создаются вне таблицы (без родительского виджета) и вставляются в таблицу с помощью QTableWidgetItem (). [15,21]

Описание используемых функций

setItem

Устанавливает элемент для данной строки (row) и столбца (column) в значение item.

item

Возвращает элемент для данной строки и столбца, если он был установлен; в противном случае возвращает 0.

setRowCount

Данное свойство содержит количество строк в таблице.

По умолчанию для таблицы, построенной без подсчетов строк и столбцов, это свойство содержит значение 0.

setColumnCount

Свойство содержит количество столбцов в таблице.

По умолчанию для таблицы, построенной без подсчетов строк и столбцов, это свойство содержит значение 0.

currentRow

Возвращает строку текущего элемента.

removeRow

Удаляет строку (row) и все ее элементы из таблицы.

Модуль QMessageBox**Описание**

Класс QMessageBox предоставляет модальный диалог для информирования пользователя или для того, чтобы задать ему вопрос и получить ответ.

В окне сообщения отображается основной текст для предупреждения пользователя о ситуации, информативный текст для дальнейшего объяснения предупреждения или для того, чтобы задать пользователю вопрос, а также необязательный подробный текст для предоставления еще большего количества данных, если пользователь запросит его. Окно сообщения также может отображать значок и стандартные кнопки для принятия ответа пользователя.

[15,21]

Модуль QLineEdit

QLineEdit - это однострочный текстовый редактор.

Редактирование строк позволяет пользователю вводить и редактировать одну строку обычного текста с помощью полезной коллекции функций редактирования, включая отмену и повтор, вырезание и вставку, а также перетаскивание. Он также может быть использован в качестве поля "только для записи" для входных данных, таких как пароли. [15,21]

Модуль mainWindow

Класс QMainWindow предоставляет главное окно приложения.

Конструктор

Отображение фона приложения. Задаёт количество строк и столбцов таблиц. Устанавливает названия заголовков таблиц. Открывает промежуточный файл, считывает имя пользователя, происходит поиск файла с необходимым именем, после открытия считывает данные и записывает их в таблицу «Рекомендации». После операции промежуточный файл удаляется.

Слот Choice_of_products

Отвечает за копирование продукта из главной таблицы в таблицу «Выбранные продукты». Вывод запроса на количество продукта.

При повторном добавлении одного и того же продукта, выводится предупреждающее сообщение.

Слот Delete_Button

Удаляет выбранную строку в таблице «Выбранные продукты».

Слот to_Change

Изменяет количество грамм продукта в таблице «Выбранные продукты». При отсутствии каких-либо значений в таблице, выводится соответствующее сообщение.

Слот comboBox

В качестве параметра принимает индекс выбранного значения. Значение передается в switch (), выбирается case. Происходит открытие файла .txt с нужными пользователю данными о продукте. Затем построчное считывание и запись данных в таблицу.

Слот action

Вызывает окно, в котором написан принцип работы приложения.

Функция sum

Суммирует КБЖУ в таблице «Результат КБЖУ». Если значение из таблицы результата равно (с погрешностью 10%) значению из таблицы «Рекомендации», ячейка в таблице «Результат КБЖУ» выделяется зеленым цветом. При несоблюдении рекомендуемых норм цвет ячейки остается красным.

Модуль new_user

Содержит окно регистрации пользователя.

Конструктор

Запрещает использование пробела, выгружает данные из файла в ListWidget.

Слот pushButton_2

Создает файл username.txt. Считывает значения, которые были выбраны пользователем, проверяет их на правильность. В файл username записывается имя пользователя. Создается файл .txt с названием, полностью соответствующим имени созданного пользователя, в который записываются все расчетные данные.

Слот VIBOR

Создается промежуточный файл, в который передается имя выбранного пользователя для последующего использования в модуле mainwindow.

Слот Delete_Button

Удаляет выбранную учетную запись из папки User, стирает имя из файла username.txt.

ГЛАВА 3

ТЕСТИРОВАНИЕ ПРОГРАММЫ

3.1 Тестирование исходного кода

Шаг /Step No.	Действие (операция) /Process (Actions)	Ожидаемый результат /Expected Results	Результат /Actual Results	Пройден /не пройден/ не доступен*	Комментарии/ Notes
Запуск exe-файла					
1	Разархивировать скачанный архив в произвольное место	В выбранном месте появляется папка с исполняемым exe-файлом.	Найден .exe файл	Пройден	
2	Открыть и прочитать readme.txt файл	Понята механика проекта	В документе всё доступно расписано	Пройден	
3	Открыть папку с исполняемым файлом и запустить exe-файл "fitfat"	Открывается приложение. Видно окно регистрации пользователя с полями для заполнения. Имеется список пользователей.	Приложение открылось, описанные элементы присутствуют	Пройден	

Рисунок 3.1.1 Тестирование запуска приложения

Шаг /Step No.	Действие (операция) /Process (Actions)	Ожидаемый результат /Expected Results	Результат /Actual Results	Пройден /не пройден/ не доступен*	Комментарии/ Notes
Создание пользователя					
1	Разархивировать скачанный архив в произвольное место	В выбранном месте появляется папка с исполняемым exe-файлом.	Найден .exe файл	Пройден	
3	Открыть папку с исполняемым файлом и запустить exe-файл "fitfat"	Открывается приложение. Видно окно регистрации пользователя с полями для заполнения. Имеется список пользователей.	Приложение открылось, описанные элементы присутствуют	Пройден	
2	Заполнить поля для ввода и выбрать пол. Нажать кнопку "Создать"	В левом списке появится только что созданный пользователь	Соответствует ожиданию	Пройден	

Рисунок 3.1.2 Тестирование создания пользователя

Шаг /Step No.	Действие (операция) /Process (Actions)	Ожидаемый результат /Expected Results	Результат /Actual Results	Пройден /не пройден/ не доступен*	Комментарии/ Notes
Удаление пользователя					
1	Разархивировать скачанный архив в произвольное место	В выбранном месте появляется папка с исполняемым exe-файлом.	Найден .exe файл	Пройден	
3	Открыть папку с исполняемым файлом и запустить exe-файл "fitfat"	Открывается приложение. Видно окно регистрации пользователя с полями для заполнения. Имеется список пользователей.	Приложение открылось, описанные элементы присутствуют	Пройден	
2	Заполнить поля для ввода и выбрать пол. Нажать кнопку "Создать"	В левом списке появится только что созданный пользователь	Соответствует ожиданию	Пройден	
4	В списке пользователей выделить пользователя и нажать кнопку "Удалить"	Выделенный пользователь удаляется из списка	Шарик передвигается в примерную область, куда направлен взгляд пользователя	Пройден	

Рисунок 3.1.3 Тестирование удаления пользователя

Шаг /Step No.	Действие (операция) /Process (Actions)	Ожидаемый результат /Expected Results	Результат /Actual Results	Пройден /не пройден/ не доступен*	Комментарии/ Notes
Выбор пользователя					
1	Разархивировать скачанный архив в произвольное место	В выбранном месте появляется папка с исполняемым exe-файлом.	Найден .exe файл	Пройден	
3	Открыть папку с исполняемым файлом и запустить exe-файл "fitfat"	Открывается приложение. Видно окно регистрации пользователя с полями для заполнения. Имеется список пользователей.	Приложение открылось, описанные элементы присутствуют	Пройден	
2	Заполнить поля для ввода и выбрать пол. Нажать кнопку "Создать"	В левом списке появится только что созданный пользователь	Соответствует ожиданию	Пройден	
4	Кликнуть ЛКМ на учетную запись. Нажать кнопку "Выбрать"	Учетная запись выделяется. После нажатия кнопки, окно закрывается. Открывается новое окно.	Соответствует ожиданию	Пройден	

Рисунок 3.1.4 Тестирование выбора пользователя

Шаг /Step No.	Действие (операция) /Process (Actions)	Ожидаемый результат /Expected Results	Результат /Actual Results	Пройден /не пройден/ не доступен*	Комментарии/ Notes
Открытие главного окна					
1	Разархивировать скачанный архив в произвольное место	В выбранном месте появляется папка с исполняемым exe-файлом.	Найден .exe файл	Пройден	
3	Открыть папку с исполняемым файлом и запустить exe-файл "fitfat"	Открывается приложение. Видно окно регистрации пользователя с полями для заполнения. Имеется список пользователей.	Приложение открылось, описанные элементы присутствуют	Пройден	
2	Заполнить поля для ввода и выбрать пол. Нажать кнопку "Создать"	В левом списке появится только что созданный пользователь	Соответствует ожиданию	Пройден	
4	Кликнуть ЛКМ на учетную запись. Нажать кнопку "Выбрать"	Учетная запись выделяется. После нажатия кнопки, окно закрывается. Открывается новое окно.	Соответствует ожиданию	Пройден	
4	Кликнуть ЛКМ на учетную запись. Нажать кнопку "Выбрать"	Учетная запись выделяется. После нажатия кнопки, окно закрывается. Открывается новое окно.	Соответствует ожиданию	Пройден	
5	Выполнить 4 пункт	Появляется окно. На нем присутствуют: Таблицы "Рекомендации", "Выбранные продукты", "Результат КБЖУ". Выбор категории продуктов. Таблица с заполненными продуктами. Таблица "Рекомендации" заполнена цифровыми значениями	Приложение открылось, описанные элементы присутствуют	Пройден	

Рисунок 3.1.5 Тестирование прохождения регистрации

Шаг /Step No.	Действие (операция) /Process (Actions)	Ожидаемый результат /Expected Results	Результат /Actual Results	Пройден /не пройден/ не доступен*	Комментарии/ Notes
Выбор продукта					
1	Выполнить тест 5			Пройден	
2	В таблице продуктов выбрать любой продукт и кликнуть на кнопку "Подтвердить"	Появление окна "Вес"	Соответствует ожиданиям	Пройден	
3	Ввести числовое значение в поле для ввода в окне "Вес". Нажать кнопку "ОК"	Появление продукта с указанными граммами в таблице "Выбранные продукты". В зависимости от введенных грамм, происходит расчет КБЖУ	Соответствует ожиданиям. Расчет производится корректно	Пройден	

Рисунок 3.1.6 Тестирование выбора продукта

Шаг /Step No.	Действие (операция) /Process (Actions)	Ожидаемый результат /Expected Results	Результат /Actual Results	Пройден /не пройден/ не доступен*	Комментарии/ Notes
Изменение веса/ удаление продукта					
1	Выполнить тест 6			Пройден	
2	В таблице "Выбранные продукты" выделить продукт и нажать кнопку "Изменить вес"	Появление окна "Вес"	Соответствует ожиданиям	Пройден	
3	Ввести числовое значение в поле для ввода в окне "Вес". Нажать кнопку "ОК"	В таблице "Выбранные продукты". В зависимости от введенных грамм, происходит расчет КБЖУ	Соответствует ожиданиям. Расчет производится корректно	Пройден	
4	В таблице "Выбранные продукты" выделить продукт и нажать кнопку "Удалить"	Удаление выделенной строки из таблицы	Соответствует ожиданиям	Пройден	
5	Добавление/ изменение продукта в таблице "Выбранные продукты"	суммируются продукты из таблицы "Выбранные продукты"	Расчеты корректны	Пройден	
6	Добавление/ изменение продукта в таблице "Выбранные продукты. При соответствии ячейки из таблицы рекомендации с ячейкой таблицы результата (допустимая погрешность 10%).	Смена цвета ячейки (красный/ зеленый)	Расчеты корректны	Пройден	

Рисунок 3.1.7 Тестирование изменения веса/ удаления продукта

Белое тестирование

Файл mainwindow.cpp

Строка 107: отсутствует закрытие файла с именем пользователя.

Строка 274: дважды написан оператор break.

Общее замечание: трудночитаемость, неудобная структура кода.

Файл new_user.cpp

Строка 48: в операторе while отсутствует завершение. Возможен бесконечный цикл.

Строка 98: вместо многочисленных if else, рекомендуется использовать оператор switch.

Вывод

Приложение выполняет поставленные задачи. Основное внимание необходимо уделить структурному оформлению кода, стилю наименования переменных для повышения читаемости кода.

3.2 Тестирование интерфейса пользователя и юзабилити

Окно регистрации пользователя успешно прошло тестирование. При удалении или выборе пользователя не возникает ошибок. В графу «Имя пользователя», как и было указано, запрещен ввод пробелов. В остальные поля запрещен ввод не числовых символов. После заполнения полей и нажатия кнопки «Создать» в списке пользователей показывается только что созданная ученая запись.

Главное окно успешно прошло тестирование интерфейса. Все кнопки работают правильно и без ошибок. Подсчет результатов производится корректно.

ГЛАВА 4

ПЕРСПЕКТИВЫ ДАЛЬНЕЙШЕГО РАЗВИТИЯ ПРОЕКТА

4.1 Перспективы технического развития

Перспективы развития приложения:

- Разместить приложение на Android и IOS устройства;
- Расширить ассортимент продуктов;
- Добавить готовые блюда и напитки;
- Добавить встроенные платные покупки;
- Установить напоминание о приеме пищи для контроля спортивного питания.

Добавление предложенного функционала сделает приложение более удобным при использовании мобильного телефона. Расширенный функционал и выбор продуктов даст возможность более корректно составить меню и рассчитать энергетическую ценность продуктов питания и напитков, скорректировать прием пищи при спортивном питании.

4.2 Перспективы монетизации

Для перспективы монетизации бесплатного приложения необходимо встроить платные дополнения. В качестве платного дополнения может выступать разработка комплекса физических упражнений с различными целями: поддержка массы тела, сушка тела, увеличение мышечной массы тела. Платное дополнение предлагает комплексный подход: расчёт продуктов питания, рекомендуемые физические упражнения, напоминания о приемах пищи и физической нагрузке с ежемесячным отчетом результата изменения массы тела.

ЗАКЛЮЧЕНИЕ

В результате создана программа расчета энергетической ценности продуктов. Программа производит расчет КБЖУ на дневной или разовый прием пищи. Для разработки программы был использован основной язык программирования C++ и кроссплатформенный фреймворк Qt. C++ - компилируемый язык программирования общего назначения, сочетает свойства как высокоуровневых, так и низкоуровневых языков программирования.

Выполнены следующие задачи:

- Изучена тема правильного питания
- Изучена энергетическая ценность продуктов
- Изучены теоретические аспекты основного языка программирования C++
- Изучен кроссплатформенный фреймворк Qt

Рассмотрены перспективы дальнейшего развития проекта, а также перспективы монетизации проекта.

ЛИТЕРАТУРА

1. Ашарина, И.В. Основы программирования на языках С и С++: Курс лекций для высших учебных заведений / И.В. Ашарина. — М.: Гор. линия-Телеком, 2018. — 208 с.
2. Барановский А.Ю. Диетология: руководство/А.Ю. Барановский. - СПб.: Питер, 2008. - 960 с.
3. Бланшет, Ж. М. Саммерфилд «QT 4: программирование GUI на С++», КУДИЦ-Пресс, 2008
4. Бобровский, С. П. Самоучитель программирования на языке С++/ С. П. Бобровский. - КУДИЦ-Пресс, 2013 – 238 с.
5. Бьярн Страустрап, Введение в язык СИ++ 1995г.
6. Воробьев Р.И. Питание и здоровье. - М.: Медицина. 2010. - 156 с.
7. Герберт Шилдт Полный справочник по С++ /Издание 4. 2015 год.
8. Герберт Шилдт. С + + для начинающих. Пер. с англ. М: ЭКОМ Паблишерз 2007.
9. Горохов В.А., Горохова С.Н. Лечебно-сбалансированное питание - путь к здоровью и долголетию. - СПб.: Питер. 2011. - 278 с.
10. Джарод Холингвэрт, Дэн Баттерфилд, Боб Сворт, Джэйми Оллсоп С++. Руководство разработчика. С++
11. Дрожжина Н.А. Особенности формирования пищевого поведения в студенческой среде /Дрожжина Н.А., Максименко Л.В. // Вопр. диетол. - 2012. - Т.2, № 2. - С.27.
12. Дроздова Т. М., Влощинский П.Е., Поздняковский В.М.. Физиология питания. Учебник / Т.М. Дроздова, П.Е. Влощинский, В.М. Поздняковский. - Новосибирск: Сибирское университетское издательство, 2012. - 352 с.
13. Либерти, Д. Д. Освой самостоятельно С++ за 21 день/ Символ-Плюс, 2010
14. Ренате Цельтнер. Раздельное питание. Справочник здоровья/ - Сигма-пресс: Феникс, 2009. - 96 с.

- 15.Саммерфилд М. «Qt. Профессиональное программирование. Разработка кроссплатформенных приложений на C++», Символ-Плюс, 2011
- 16.Скальный А.В. Основы здорового питания: пособие по общей нутрициологии/ А.В. Скальный. - Оренбург: ГОУ ОГУ, 2011. - 258 с.
- 17.Сорока Н.Ф. Питание и здоровье/ - Минск, 2010. - 242 с.
- 18.Страуструп, Б. Язык программирования C++: Специальное издание / Б. Страуструп; Пер. с англ. Н.Н. Мартынов. — М.: БИНОМ, 2017. — 1136 с.
- 19.Шевченко В.П. Питание и долголетие/ - М.: Университетская книга, Логос, 2012. - 320 с.
- 20.Шилов В. Н., Мицьо В. П. Здоровое питание./В.Н. Шилов, В.П. Мицьо. - М.: Парус, 2011. - 224 с.
- 21.Шлее М. «Qt 5.3. Профессиональное программирование на C++», БХВ-Петербург, 2015
- 22.Эддар А. Трактат о питании/ - М.: Просвещение, 2009. - 157 с.

ПРИЛОЖЕНИЕ 1

КОД ОСНОВНЫХ МОДУЛЕЙ ПРОЕКТА

Код модуля mainwindow.cpp:

```
#include "mainwindow.h"
#include "ui_mainwindow.h"
#include <QTableWidget>
#include <QtGui>
#include <QMessageBox>
#include <QtWidgets>
#include "help.h"
#include "new_user.h"
// https://youtu.be/kOVUovnj4lg
double plus(double item){//допустимая погрешность в 10%
    return item + (item * 0.1);
}
double minus (double item){
    return item - (item * 0.1);
}

void MainWindow::sum() {
    QString water;
    double water_d;
    for(int j = 1; j!=7; j++){
        water_d=0;
        for(int i=0; i<10; i++){
            if(ui->tableWidget_2->item(i,j)==0)
                break;
            water=ui->tableWidget_2->item(i,j)->text();
            water_d+=water.toDouble();
        }
        water = QString::number(water_d);
        ui->tableResult->setItem(0,j-1,new QTableWidgetItem(water));
    }
    QString recommendation_protein = ui->table_recom->item(0,0)->text(),
    result_protein = ui->tableResult->item(0,1)->text(),
        recommendation_fat = ui->table_recom->item(0,1)->text(), result_fat =
    ui->tableResult->item(0,2)->text(),
        recommendation_carbohydrates = ui->table_recom->item(0,2)->text(),
    result_carbohydrates = ui->tableResult->item(0,3)->text(),
        recommendation_calories = ui->table_recom->item(0,3)->text(),
    result_calories = ui->tableResult->item(0,4)->text();
    double Drecommendation_protein = recommendation_protein.toDouble(),
    Dresult_protein = result_protein.toDouble(),
        Drecommendation_fat = recommendation_fat.toDouble(), Dresult_fat =
    result_fat.toDouble(),
        Drecommendation_carbohydrates =
    recommendation_carbohydrates.toDouble(), Dresult_carbohydrates =
    result_carbohydrates.toDouble(),
        Drecommendation_calories = recommendation_calories.toDouble(),
    Dresult_calories = result_calories.toDouble();
    if(Drecommendation_protein > plus(Dresult_protein) || Drecommendation_protein
    < minus(Dresult_protein)){
        ui->tableResult->item(0,1)->setBackground(Qt::red);
    }
    else{ui->tableResult->item(0,1)->setBackground(Qt::green);}
    if(Drecommendation_fat > plus(Dresult_fat) || Drecommendation_fat <
    minus(Dresult_fat)){
        ui->tableResult->item(0,2)->setBackground(Qt::red);
    }
```

```

    }
    else{ui->tableResult->item(0,2)->setBackground(Qt::green);}
    if(Drecommendation_carbohydrates > plus(Dresult_carbohydrates) ||
Drecommendation_carbohydrates < minus(Dresult_carbohydrates)){
        ui->tableResult->item(0,3)->setBackground(Qt::red);
    }
    else{ui->tableResult->item(0,3)->setBackground(Qt::green);}
    if(Drecommendation_calories > plus(Dresult_calories) ||
Drecommendation_calories < minus(Dresult_calories)){
        ui->tableResult->item(0,4)->setBackground(Qt::red);
    }
    else{ui->tableResult->item(0,4)->setBackground(Qt::green);}
}
MainWindow::MainWindow(QWidget *parent)
: QMainWindow(parent)
, ui(new Ui::MainWindow)
{
    ui->setupUi(this);
    QPalette Pal(palette());
    QImage background2(":/IMG/background.png");
    Pal.setBrush(QPalette::Background, background2);
    this->setAutoFillBackground(true);
    this->setPalette(Pal);
    New_User window;
    window.setModal(true);
    window.exec();
    ui->table_recom->setColumnCount(4);
    ui->table_recom->setRowCount(1);
    ui->table_recom-
>setHorizontalHeaderLabels(QStringList()<<"Белки"<<"Жиры"<<"Углеводы"<<"Калории")
;
    ui->tableWidget->horizontalHeader()->setDefaultSectionSize(106);
    ui->tableWidget_2->horizontalHeader()->setDefaultSectionSize(94);
    ui->tableResult->horizontalHeader()->setDefaultSectionSize(80);
    ui->tableWidget->setColumnCount(6);
    ui->tableWidget-
>setHorizontalHeaderLabels(QStringList()<<"Продукт"<<"Вода"<<"Белки"<<"Жиры"<<"Уг
леводы"<<"Калории");
    ui->tableWidget_2->setColumnCount(7);
    ui->tableWidget_2-
>setHorizontalHeaderLabels(QStringList()<<"Продукт"<<"Вода"<<"Белки"<<"Жиры"<<"Уг
леводы"<<"Калории"<<"Грамм");
    ui->tableResult->setColumnCount(6);
    ui->tableResult->setRowCount(1);
    ui->tableResult-
>setHorizontalHeaderLabels(QStringList()<<"Вода"<<"Белки"<<"Жиры"<<"Углеводы"<<"К
алории"<<"Грамм");
    ui->Choice_of_products->setEnabled(false);
    ui->comboBox->addItem("Хлебобулочные изд.");
    ui->comboBox->addItem("Мясо");
    ui->comboBox->addItem("Морепродукты");
    ui->comboBox->addItem("Молочные продукты");
    ui->comboBox->addItem("Фрукты и ягоды");
    ui->comboBox->addItem("Овощи");
    ui->comboBox->addItem("Крупы");
    ui->comboBox->addItem("Грибы и орехи");
    QFile inf(QApplication::applicationDirPath () + "MANGAL.txt");
    if(!inf.open(QIODevice::ReadOnly)){
        QMessageBox::warning(0,"Предупреждение", " Не был выбран
пользователь");
    }
    else{
        QTextStream stream(&inf);

```

```

    QString name_name;
    stream>>name_name;
    ui->gest->setText(name_name);
    inf.close();
    QFile::remove(QApplication::applicationDirPath () + "MANGAL.txt");
    QFile Result(QApplication::applicationDirPath () +
"/User/"+name_name+".txt");
    if(!Result.open(QIODevice::ReadOnly)){
        ui->statusbar->showMessage("Файл не открыт");
    };
    QTextStream streama(&Result);
    QString sleep;
    for(int j=0;j!=4;j++){
        streama>>sleep;
        QTableWidgetItem *ittt=new QTableWidgetItem;
        ittt->setData(Qt::EditRole,sleep);
        ui->table_recom->setItem(0,j,ittt);
    }
    Result.close();
}
}
MainWindow::~MainWindow()
{
    delete ui;
}

void MainWindow::on_Choice_of_products_clicked()
{
    QString str = QInputDialog::getText(this,"Вес","Введите вес продукта (гр)
",QLineEdit::Normal,"");
    double std =str.toDouble();
    if(std<=0){QMessageBox::warning(0,"Предупреждение", "Неккоректный ввод");}
    else{
        int i= ui->tableWidget->currentRow();//перенос целой строки
        int temp=0;
        bool flag = true,loop = true;
        QString textic,str_gramm;
        double dack, num_gramm;
        for(int val=0;val!=100;val++){
            QString tmp1, tmp2;
            tmp1 = ui->tableWidget->item(i,0)->text();
            if (ui->tableWidget_2->item(val,0) != 0){
                tmp2 = ui->tableWidget_2->item(val,0)->text();
                if(tmp1==tmp2){
                    QMessageBox::critical(this,"Внимание","Этот продукт уже
выбран");
                    loop=false;
                    break;
                }
            }
        }
        if(loop){
            for(int j=0;j!=6;j++){
                while(flag){
                    if(ui->tableWidget_2->item(temp,0)== 0){
                        flag = false;
                    }else temp++;
                }
                num_gramm = str.toDouble();
                textic =ui->tableWidget->item(i,j)->text();
                dack=textic.toDouble()/100*num_gramm;
                if(j>0){
                    textic = QString::number(dack);

```

```

        }
        ui->tableWidget_2->setRowCount(temp+1);
        ui->tableWidget_2->setItem(temp,j,new QTableWidgetItem(textic));
        ui->tableWidget_2->setItem(temp,6,new QTableWidgetItem(str));////!!
    }
}
sum();
}

}

void MainWindow::on_Delete_Button_clicked()
{
    int i= ui->tableWidget_2->currentRow();
    ui->tableWidget_2->removeRow(i);
    sum();
}

void MainWindow::on_to_Change_clicked()
{
    ui->tableWidget_2->setFocus();
    if(ui->tableWidget_2->item(0,0)==0){
        QMessageBox::warning(0,"Предупреждение", "В таблице нет продуктов!");
    }
    else{
        QString str = QInputDialog::getText( /*0*/this, "Вес", "Введите вес
продукта (гр) ", QLineEdit::Normal, "");
        double std =str.toDouble();
        if(std<=0){QMessageBox::warning(0,"Предупреждение", "Неккоректный
ввод!");}
        else{
            int i= ui->tableWidget_2->currentRow();//row
            QString tex_G = ui->tableWidget_2->item(i,6)->text(),tex_T;
            double num_G = tex_G.toDouble(),num_T,result;
            for(int j=1;j!=6;j++){
                tex_T=ui->tableWidget_2->item(i,j)->text();
                num_T=tex_T.toDouble();
                result=num_T/num_G*std;
                tex_T = QString::number(result);
                ui->tableWidget_2->setItem(i,j,new QTableWidgetItem(tex_T));
                ui->tableWidget_2->setItem(i,6,new QTableWidgetItem(str));
            }
        }
        sum();
    }
}

void MainWindow::on_comboBox_currentIndexChanged(int index)
{
    switch (index) {
    case 0: {
        ui->Choice_of_products->setEnabled(true);
        QFile f(QApplication::applicationDirPath () +
"/Products_Lists/bread.txt");
        if(!f.open(QIODevice::ReadOnly)){
            ui->statusbar->showMessage("Файл не открыт");
        };
        QTextStream stream(&f);
        QString s;
        for(int i=0;i!=9;i++){
            ui->tableWidget->setRowCount(i+1);
            for(int j=0;j!=6;j++){
                stream>>s;
                QTableWidgetItem *it=new QTableWidgetItem;
                it->setData(Qt::EditRole,s);
                ui->tableWidget->setItem(i,j,it);
            }
        }
    }
}

```

```

    }
}
f.close();
ui->tableWidget->show();break;
}
case 1: {ui->Choice_of_products->setEnabled(true);
    QFile f(QApplication:: applicationDirPath () +
"/Products_Lists/meat.txt");
    if(!f.open(QIODevice::ReadOnly)){ ui->statusbar->showMessage("Файл не
открыт");};
    QTextStream stream(&f);
    QString s;
    for(int i=0;i!=11;i++){
        ui->tableWidget->setRowCount(i+1);
        for(int j=0;j!=6;j++){
            stream>>s;
            QTableWidgetItem *it=new QTableWidgetItem;
            it->setData(Qt::EditRole,s);
            ui->tableWidget->setItem(i,j,it);
        }
    }
    f.close();
    ui->tableWidget->show();} ;break;
case 2: {ui->Choice_of_products->setEnabled(true);
    QFile f(QApplication:: applicationDirPath () +
"/Products_Lists/fish.txt");
    if(!f.open(QIODevice::ReadOnly)){ ui->statusbar->showMessage("Файл не
открыт");};
    QTextStream stream(&f);
    QString s;
    for(int i=0;i!=44;i++){
        ui->tableWidget->setRowCount(i+1);
        for(int j=0;j!=6;j++){
            stream>>s;
            QTableWidgetItem *it=new QTableWidgetItem;
            it->setData(Qt::EditRole,s);
            ui->tableWidget->setItem(i,j,it);
        }
    }
    f.close();
    ui->tableWidget->show();} ;break;
case 3: {ui->Choice_of_products->setEnabled(true);
    QFile f(QApplication:: applicationDirPath () +
"/Products_Lists/molocnie_prodykt.txt");
    if(!f.open(QIODevice::ReadOnly)){ ui->statusbar->showMessage("Файл не
открыт");};
    QTextStream stream(&f);
    QString s;
    for(int i=0;i!=14;i++){
        ui->tableWidget->setRowCount(i+1);
        for(int j=0;j!=6;j++){
            stream>>s;
            QTableWidgetItem *it=new QTableWidgetItem;
            it->setData(Qt::EditRole,s);
            ui->tableWidget->setItem(i,j,it);
        }
    }
    f.close();
    ui->tableWidget->show();} ;break;break;
case 4: {ui->Choice_of_products->setEnabled(true);
    QFile f(QApplication:: applicationDirPath () +
"/Products_Lists/Fruits_and_yagodi.txt");

```



```

        if(!f.open(QIODevice::ReadOnly)){ ui->statusbar->showMessage("Файл не
открыт");};
        QTextStream stream(&f);
        QString s;
        for(int i=0;i!=38;i++){
            ui->tableWidget->setRowCount(i+1);
            for(int j=0;j!=6;j++){
                stream>>s;
                QTableWidgetItem *it=new QTableWidgetItem;
                it->setData(Qt::EditRole,s);
                ui->tableWidget->setItem(i,j,it);
            }
        }
        f.close();
        ui->tableWidget->show();} ;break;
    case 5: {ui->Choice_of_products->setEnabled(true);
        QFile f(QApplication:: applicationDirPath () +
"/Products_Lists/VEg.txt");
        if(!f.open(QIODevice::ReadOnly)){ ui->statusbar->showMessage("Файл не
открыт");};
        QTextStream stream(&f);
        QString s;
        for(int i=0;i!=30;i++){
            ui->tableWidget->setRowCount(i+1);
            for(int j=0;j!=6;j++){
                stream>>s;
                QTableWidgetItem *it=new QTableWidgetItem;
                it->setData(Qt::EditRole,s);
                ui->tableWidget->setItem(i,j,it);
            }
        }
        f.close();
        ui->tableWidget->show();} ;break;
    case 6: {ui->Choice_of_products->setEnabled(true);
        QFile f(QApplication:: applicationDirPath () +
"/Products_Lists/krypi.txt");
        if(!f.open(QIODevice::ReadOnly)){ ui->statusbar->showMessage("Файл не
открыт");};
        QTextStream stream(&f);
        QString s;
        for(int i=0;i!=10;i++){
            ui->tableWidget->setRowCount(i+1);
            for(int j=0;j!=6;j++){
                stream>>s;
                QTableWidgetItem *it=new QTableWidgetItem;
                it->setData(Qt::EditRole,s);
                ui->tableWidget->setItem(i,j,it);
            }
        }
        f.close();
        ui->tableWidget->show();} ;break;
    case 7: {ui->Choice_of_products->setEnabled(true);
        QFile f(QApplication:: applicationDirPath () +
"/Products_Lists/mashroom_and_orexi.txt");
        if(!f.open(QIODevice::ReadOnly)){ ui->statusbar->showMessage("Файл не
открыт");};
        QTextStream stream(&f);
        QString s;
        for(int i=0;i!=10;i++){
            ui->tableWidget->setRowCount(i+1);
            for(int j=0;j!=6;j++){
                stream>>s;
                QTableWidgetItem *it=new QTableWidgetItem;

```

```

        it->setData(Qt::EditRole,s);
        ui->tableWidget->setItem(i,j,it);
    }
}
f.close();
ui->tableWidget->show();} ;break;
default:ui->statusbar->showMessage("Error");
}
}

void MainWindow::on_action_triggered()
{
    Help window;
    window.setModal(true);
    window.exec();
}

```

Код модуля new_user.cpp:

```

#include "new_user.h"
#include "ui_new_user.h"
#include "ui_mainwindow.h"
#include "mainwindow.h"
#include <QtGui>
#include <QMessageBox>
#include <iostream>
using namespace std;
double New_User::coefficient(double num, int i){
    switch (i) {
        case 0: num*=1.2;break;
        case 1: num*=1.3;break;
        case 2: num*=1.5;break;
        case 3: num*=1.7;break;
        case 4: num*=1.9;break;
    }
    return num;
}
New_User::New_User(QWidget *parent) :
    QDialog(parent),
    ui(new Ui::New_User)
{
    ui->setupUi(this);
    QRegExp rx = QRegExp("[^\\x0020]*");
    QRegExpValidator * LoginValidator = new QRegExpValidator(rx, this);
    ui->user_name->setValidator(LoginValidator);
    QFile f( "username.txt");
    if(!f.open(QIODevice::ReadOnly))
        return;
    while(true){
        QString line = f.readLine();
        line.chop(2);
        if(line==""){
            break;
        }
        ui->listWidget->addItem(line);
    }
    f.close();
}
New_User::~New_User()
{
    delete ui;
}
void New_User::on_pushButton_2_clicked()

```

```

{
QFile Pop("username.txt");
int n = 0;
Pop.open(QIODevice::ReadOnly | QIODevice::Text);
n = QTextStream(&Pop).readAll().split('\n').count();
if(n==6){
    QMessageBox::warning(0,"Предупреждение", "Превышен лимит пользователей");
}
else{
    QString str_age = ui->age->text(),
            str_weight = ui->weight->text(),
            str_height = ui->height->text();
    double std_age =str_age.toDouble(),
            std_weight = str_weight.toDouble(),
            std_height = str_height.toDouble();
    if (ui->men->isChecked()==false && ui->women->isChecked()==false){QMessageBox::warning(0,"Предупреждение", "Выберите пол");}else
        if (std_age <= 0 || std_age > 100){QMessageBox::warning(0,"Предупреждение", "Введен некорректный возраст");}else
            if (std_weight <= 0 || std_weight>200 )
                {QMessageBox::warning(0,"Предупреждение", "Введен некорректный вес");}else
                if (std_height <= 0 || std_height>2500
                ){QMessageBox::warning(0,"Предупреждение", "Введен некорректный рост");}
            else {
                QString file_name = ui->user_name->text();
                QFile user_name("username.txt");
                if(user_name.open(QIODevice::Append | QIODevice::Text)){
                    QTextStream fout(&user_name);
                    fout<<file_name<<endl;
                    user_name.close();
                }
                QString file_name = ui->user_name->text();
                QFile user(QApplication::applicationDirPath () +"/User/"+file_name + ".txt");
                if(user.open(QIODevice::WriteOnly | QIODevice::Text)){
                    QTextStream out(&user);
                }
                double protein,fats,carb;
                if(ui->men->isChecked()){
                    int index=ui->activ->currentIndex();
                    int index_reason=ui->reason->currentIndex();
                    double need_men;
                    need_men = 66.47+13.75*ui->weight->text().toDouble()+5*ui->height->text().toDouble()-6.74*ui->age->text().toDouble();
                    switch(index_reason){
                        case 0:
                            protein = 1.3*ui->weight->text().toDouble();
                            fats = 1.1*ui->weight->text().toDouble();
                            carb = 6.8*ui->weight->text().toDouble();
                            ;break;
                        case 1:
                            protein = 2.3*ui->weight->text().toDouble();
                            fats = 0.9*ui->weight->text().toDouble();
                            carb = 6.6*ui->weight->text().toDouble();
                            ;break;
                        case 2:
                            protein = 1.5*ui->weight->text().toDouble();
                            fats = 0.7*ui->weight->text().toDouble();
                            carb = 2.7*ui->weight->text().toDouble();
                            ;break;
                    }
                }
                out<<protein<<endl;
                out<<fats<<endl;
                out<<carb<<endl;
            }
        }
}

```

```

        out<<coefficient(need_men,index)<<endl;
    }else
        if(ui->women->isChecked()){
            int index=ui->activ->currentIndex();
            double need_women;
            need_women = 655.1 + 9.6*ui->weight->text().toDouble()+1.85*ui-
>height->text().toDouble()-4.68*ui->age->text().toDouble();
            int index_reason=ui->reason->currentIndex();
            switch(index_reason){
                case 0:
                    protein = 1.3 * ui->weight->text().toDouble();
                    fats = 1.1 * ui->weight->text().toDouble();
                    carb = 6.8 * ui->weight->text().toDouble();
                    ;break;
                case 1:
                    protein = 2.3 * ui->weight->text().toDouble();
                    fats = 0.9 * ui->weight->text().toDouble();
                    carb = 6.6 * ui->weight->text().toDouble();
                    ;break;
                case 2:
                    protein = 1.5 * ui->weight->text().toDouble();
                    fats = 0.7 * ui->weight->text().toDouble();
                    carb = 2.7 * ui->weight->text().toDouble();
                    ;break;
            }
            out<<protein<<endl;
            out<<fats<<endl;
            out<<carb<<endl;
            out<<coefficient(need_women,index)<<endl;
        }
        user.close();
    }
}

ui->listWidget->clear();
QFile f( "username.txt");
if(!f.open(QIODevice::ReadOnly))
    return;
for(int i = 0 ;i!=5;i++){
    QString line = f.readLine();
    line.chop(2);
    ui->listWidget->addItem(line);
}
f.close();
}

void New_User::on_VIBOR_clicked()
{
    QString tex_G = ui->listWidget->currentItem()->text();
    QFile user_tmp(QApplication::applicationDirPath ()+"MANGAL.txt");
    if( user_tmp.open((QIODevice::WriteOnly | QIODevice::Text))){
        QTextStream hex(&user_tmp);
        hex<<tex_G;
    }
    close();
}

void New_User::on_Delete_Button_clicked()
{
    if(ui->listWidget->count()==0){
        QMessageBox::critical(0,"Ошибка", "Создайте пользователя");
    }else{
        QString delta = ui->listWidget->currentItem()->text();

```

```

    QFile::remove(QApplication:: applicationDirPath () + "/User/" + delta
+ ".txt");
    QListWidgetItem *it = ui->listWidget->item(ui->listWidget-
>currentRow());
    delete it;
    QFile f( "username.txt");
    if(f.open(QIODevice::WriteOnly | QIODevice::Text)){
        QTextStream ret(&f);
        int betta = ui->listWidget->count();
        for(int i = 0; i<betta; i++){
            ret<<ui->listWidget->item(i)->text();
        }
    }
}
}

```

ПРИЛОЖЕНИЕ 2

КОД ТЕСТИРУЮЩЕГО ПРОЕКТА

Тестируемые модули, в которых были выявлены неполадки:

Модуль mainwindow.cpp

Ошибка: отсутствует закрытие файла «Result»

```

QFile inf(QApplication:: applicationDirPath () + "MANGAL.txt");
if(!inf.open(QIODevice::ReadOnly)){
    QMessageBox::warning(0, "Предупреждение", " Не был выбран
пользователь");
}
else{
    QTextStream stream(&inf);
    QString name_name;
    stream>>name_name;
    ui->gest->setText(name_name);
    inf.close();
    QFile::remove(QApplication:: applicationDirPath () + "MANGAL.txt");
    QFile Result(QApplication:: applicationDirPath () +
"/User/"+name_name+".txt");
    if(!Result.open(QIODevice::ReadOnly)){
        ui->statusbar->showMessage("Файл не открыт");
    };
    QTextStream streama(&Result);
    QString sleep;
    for(int j=0;j!=4;j++){
        streama>>sleep;
        QTableWidgetItem *ittt=new QTableWidgetItem;
        ittt->setData(Qt::EditRole,sleep);
        ui->table_recom->setItem(0,j,ittt);
    }
}

```

Ошибка: повторное использование оператора break.

```

case 3: {ui->Choice_of_products->setEnabled(true);
    QFile f(QApplication:: applicationDirPath () +
"/Products_Lists/molocnie_prodykt.txt");
    if(!f.open(QIODevice::ReadOnly)){ ui->statusbar->showMessage("Файл не
открыт");};};

```

```

QTextStream stream(&f);
QString s;
for(int i=0;i!=14;i++){
    ui->tableWidget->setRowCount(i+1);
    for(int j=0;j!=6;j++){
        stream>>s;
        QTableWidgetItem *it=new QTableWidgetItem;
        it->setData(Qt::EditRole,s);
        ui->tableWidget->setItem(i,j,it);
    }
}
f.close();

ui->tableWidget->show();};break;break;

```

Модуль new_user.cpp

Ошибка: бесконечный цикл while.

```

while(true){
    QString line = f.readLine();
    line.chop(2);
    ui->listWidget->addItem(line);
}

```