

## Оглавление

ВВЕДЕНИЕ .....	2
1 ОБЗОР ИСТОЧНИКОВ .....	4
1.1 Обзор используемой литературы.....	4
1.2 Основные термины.....	5
1.3 Обзор аналогов .....	6
2 СТРУКТУРНОЕ ПРОЕКТИРОВАНИЕ .....	7
3 ФУНКЦИОНАЛЬНОЕ ПРОГРАММИРОВАНИЕ .....	8
4 РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ.....	13
5 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ .....	16
6 ТЕСТИРОВАНИЕ .....	26
ЗАКЛЮЧЕНИЕ.....	31
СПИСОК ЛИТЕРАТУРЫ .....	32
ПРИЛОЖЕНИЕ А.....	33
ПРИЛОЖЕНИЕ Б .....	34
ПРИЛОЖЕНИЕ В .....	35
ПРИЛОЖЕНИЕ Г .....	36
ПРИЛОЖЕНИЕ Д.....	37

## ВВЕДЕНИЕ

Фитнес дома — это отличная идея, если вы хотите похудеть и привести свое тело в порядок, а возможности пойти в зал по тем или иным причинам у вас нет. Для этого и нужна программа «TRAINING PLANNER».

Чтобы заниматься спортом правильно, нужно помнить про следующие вещи, рассмотрим их по порядку:

- **Мотивация**

Пожалуй, это один из ключевых моментов, от которых нужно отталкиваться. Тут не сработает светлая и чистая мечта быть стройным, здоровым, самым красивым на планете. Ставьте конкретную цель, которая будет мотивировать вас работать для ее достижения. Мне, например, всегда хочется выглядеть хорошо и быть в тонусе. А еще нет возможности платить космические, как для студента, суммы за абонемент в зал. Найдите ту самую причину, которая будет заставлять вас каждый день работать над собой.

- **Найти место для тренировок**

Если вы занимаетесь в квартире, то найдите удобное для тренировок место, чтобы во время выполнения упражнений вам не приходилось переставлять мебель и отвлекаться. Вам должно быть удобно ложиться во весь рост, прыгать, делать размахи руками и т.д. Если в комнате не хватает места, то позаботьтесь об этом заранее, отодвиньте стол, расстелите коврик и занимайтесь в свое удовольствие. Утреннюю зарядку в теплое время года можно делать на улице.

- **Установите график**

Дисциплина — самое главное. Выберите удобное для вас время и дни, когда вы наименее загружены, чтобы не пропускать тренировки. Повесьте расписание на видное место, чтобы не забывать о занятиях. Да, поначалу будет сложно заставлять себя тренироваться, когда диван и сериалы так и манят, но самое главное — ввести это в привычку.

- Подберите систему тренировок

Сейчас на просторах интернета любой желающий может найти для себя подходящий онлайн-курс. В Ютубе есть огромное количество видео, которые объяснят вам, как правильно выполнять те, или иные упражнения.

«TRAINING PLANNER» идеально подходит для базового уровня. Не прыгайте выше головы и не начинайте с тяжелых комплексов, иначе можете травмироваться или просто разочароваться в своих возможностях.

- Слушайте свое тело

Не нужно начинать свои тренировки с огромной нагрузкой или марафонского забега, если вы с трудом выполняете разминку (к слову, перед тренировками никогда не забывайте разминаться), начните с постепенных нагрузок, а со временем увеличивайте обороты. Если вам тяжело выполнять определенные упражнения, то выполняйте меньшее количество повторов. Со временем вы привыкнете и сможете побить собственные рекорды.

- Не забывайте о питании

Глупо начинать бороться со своей ленью и при этом питаться по-прежнему. Кушать нужно маленькими порциями 4-5 раз в день вместо 2-3 больших приемов пищи. Отдайте предпочтение белковой еде и клетчатке, а от быстрых углеводов лучше вообще отказаться. Сначала можете записывать все съеденное за день в блокнот или специальную программу на телефоне. Так вы увидите, что делаете лишние перекусы в виде чая с шоколадками или конфеток, принесенных коллегой.

- Найдите единомышленников

Минус занятий дома — соблазн пропустить тренировку. За фитнес-зал вы платите деньги, что само по себе является неплохой мотивацией, да и тренер не даст сачковать. Поэтому лучше всего договориться с другом на совместные занятия или утренние пробежки.

# 1 ОБЗОР ИСТОЧНИКОВ

## 1.1 Обзор используемой литературы

В данной работе использовалась различная литература для проектирования данного приложения. Ниже приведено их краткое описание.

1. [1] Документация <https://doc.qt.io/>. Здесь вы найдете документацию для Qt, кросс-платформенной среды разработки программного обеспечения
2. [2] Шлее Макс. Qt 5.10. Профессиональное программирование на C++

Книга посвящена разработке приложений для Windows, Mac OS X, Linux, Android и iOS с использованием библиотеки Qt версии 5.10. Подробно рассмотрены возможности, предоставляемые этой библиотекой, и описаны особенности, выгодно отличающие ее от других библиотек. Описана интегрированная среда разработки Qt Creator и работа с технологией Qt Quick. Книга содержит исчерпывающую информацию о классах Qt 5, и так же даны практические

3. [3] Жасмин Бланшет, Марк Саммерфилд «Qt 4: программирование GUI на C++».

Из данной книги вы узнаете о наиболее эффективных приемах и методах программирования с применением Qt 4 и овладеете ключевыми технологиями в самых различных областях – от архитектуры Qt модель/представление до мощного графического процессора 2D. Авторы вооружают читателей беспрецедентно глубокими знаниями модели событий и системы компоновки Qt.

На реалистических примерах они описывают высокоэффективные методы во всех областях – от разработки основных элементов графического пользовательского интерфейса до передовых методов интеграции с базой данных и XML. Каждая глава содержит актуальный материал.

4. [4] Алексеев Е.Р., Злобин Г.Г., Костюк Д.А. и др.  
Программирование на языке С++ в среде Qt Creator  
Книга, которую открыл читатель, является с одной стороны учебником по алгоритмизации и программированию на С++ , а с другой — пособием по разработке визуальных приложений в среде Qt Creator. В книге описаны среда программирования Qt Creator и редактор Geany. При чтении книги не требуется предварительного знакомства с программированием. И др.

## 1.2 Основные термины

- Интерфейс – совокупность программных и аппаратных средств, обеспечивающих взаимодействие пользователя и компьютерной системы
- Сигналы (*SIGNAL ()*) и слоты (*SLOT ()*) используются для связи между объектами. Механизм сигналов и слотов - это основная особенность Qt и, вероятно, основная часть Qt, которая больше всего отличается по функциональности от других библиотек.
- Объектно-ориентированное программирование (ООП) – это методология программирования, основанная на представлении программы в виде совокупности объектов, каждый из которых является экземпляром определённого класса, а классы образуют иерархию наследования.
- Класс – это элемент ПО, описывающий абстрактный тип данных и его частичную или полную реализацию.
- Поле (класса) – переменная, связанная с классом или объектом.
- Объект – конкретное представление абстракции, имеющее свои свойства и методы.
- Методы – это функции, которые могут выполнять какие-либо действия над данными класса.
- Наследование – это свойство, позволяющее создать новый класс-потомок на основе уже существующего, при этом все характеристики класса родителя присваиваются классу-потомку.
- Qt – кроссплатформенный фреймворк для разработки программного обеспечения на языке программирования С++.
- Библиотека – сборник подпрограмм или объектов, используемых для разработки программного обеспечения.

### 1.3 Обзор аналогов

Аналогами разрабатываемого приложения являются: онлайн приложения для составления тренировок, также мобильные приложения.

Похожие онлайн калькуляторы:

- Калькулятор результата ваших тренировок
  - Определение эффективности домашних тренировок
  - Сколько должна длиться тренировка в тренажёрном зале?
  - Подсчет энергозатрат за одну тренировку
  - Определение уровня силовой подготовки
- И др.

В своем приложении я бы хотел увидеть возможность добавлять план тренировок и сохранять в определенное место на компьютере. Все это должно быть в удобном интерфейсе с необходимыми пунктами меню. Необходимая информация должна храниться в различных файлах, так же сам план тренировок должен содержаться в файле.

## 2 СТРУКТУРНОЕ ПРОЕКТИРОВАНИЕ

В данном разделе описана структурная схема работы приложения. Сама структурная схема представлена в приложении А.

Работа моей программы разбита на четыре основных блока: загрузка данных с файлов, работа с пользовательским интерфейсом, обработка пользовательского ввода и представление информации.

Блок файлов служит для записи и загрузки информации о поле (муж., жен.), весе, росте, возрасте, количестве тренировок в неделю и уровне тренировок.

Блок графического пользовательского интерфейса позволяет создавать, сохранять, открывать тренировку и прочие необходимые действия для минимально работы с тренировочной программой.

Блок обработки пользовательского ввода нужен для корректной обработки введённой информации, а также для её вывода информации. После того, когда данные с файла загружены, пользователь, используя этот блок может производить необходимые действия с тренировочным планом.

Блок предоставления информации обеспечивает пользователя актуальной информацией. Также этот блок предназначен для управления информацией, которая была загружена из файла или были введены пользователем.

### 3 ФУНКЦИОНАЛЬНОЕ ПРОГРАММИРОВАНИЕ

Поскольку в данной работе используется объектно-ориентированный подход программирования, то ниже приведен классы, которые использовались в программе (диаграмма классов представлена в приложении Б):

#### 1. *Human* - класс, описывающий объект человек

Поля:

- height: string - рост человека;
- weight: string - вес человека;
- age: string - возраст человека;
- workoutDiff: string - желаемая сложность тренировки;
- workoutWeekCount: string - количество тренировок в неделю;
- gender: string - пол человека;
- name: string - собственное имя человека (необходимое для названия файла);

Методы:

- GetHeight (): string const - метод для получения height, данный метод ничего не принимает и возвращает значение типа string.
- GetWeight (): string const - метод для получения weight, данный метод ничего не принимает и возвращает значение типа string.



-GetAge (): string const – метод для получения age, данный метод ничего не принимает и возвращает значение типа string.

-GetWorkoutDiff (): string const – метод для получения workoutDiff, данный метод ничего не принимает и возвращает значение типа string.

-GetWorkoutWeekCount (): string const – метод для получения workoutWeekCount, данный метод ничего не принимает и возвращает значение типа string.

-GetGender (): string const – метод для получения gender, данный метод ничего не принимает и возвращает значение типа string.

-GetName (): string const – метод для получения name, данный метод ничего не принимает и возвращает значение типа string.

-SetHeight (string str): void – метод для задания значения полю height, данный метод принимает значение типа string, которое является новым значением для поля, и ничего не возвращает.

-SetWeight (string str): void – метод для задания значения полю weight, данный метод принимает значение типа string, которое является новым значением для поля, и ничего не возвращает.

-SetAge (string str): void – метод для задания значения полю age, данный метод принимает значение типа string, которое является новым значением для поля, и ничего не возвращает.

-SetWorkoutDiff (string str): void – метод для задания значения полю workoutDiff, данный метод принимает значение типа string, которое является новым значением для поля, и ничего не возвращает.

-SetWorkoutWeekCount (string str): void – метод для задания значения полю workoutWeekCount, данный метод принимает значение типа string, которое является новым значением для поля, и ничего не возвращает.

-SetGender (string str): void – метод для задания значения полю gender, данный метод принимает значение типа string, которое является новым значением для поля, и ничего не возвращает.

-SetName (string str): void – метод для задания значения полю name, данный метод принимает значение типа string, которое является новым значением для поля, и ничего не возвращает.

2. MainWindow – класс, который предназначен для определения поведения программы при взаимодействии пользователя с графическим пользовательским интерфейсом.

Методы:

-LevelPerson(int age, int height, int weight, string levelWorkout): int – метод предназначен для вычисления уровня подготовки человека (от 0 - 10) Метод принимает значение типа int(возраст), int(рост), int(вес), string(сложность тренировки), и возвращает int.

-on\_newP\_triggered(): void – метод предназначен для создания нового плана тренировок, в пункте меню → файл → Новая программа. Метод ничего не принимает и ничего не возвращает.

-on\_exit\_triggered(): void – метод предназначен для выхода из программы, в пункте меню → файл → Выход. Метод ничего не принимает и ничего не возвращает.

-on\_about\_triggered(): void – метод предназначен для предоставления информации/помощи о программе, в пункте меню → *файл* → *Помощь*. Метод ничего не принимает и ничего не возвращает.

-on\_pushButtonCreate\_clicked (): void – метод предназначен для создания нового плана тренировок, при нажатии на кнопку «Создать программу». Метод ничего не принимает и ничего не возвращает.

-on\_pushButtonQuit\_clicked (): void – метод предназначен для выхода из программы, при нажатии на кнопку «Выход». Метод ничего не принимает и ничего не возвращает.

-creatingPlan (): void – метод предназначен для основной работы программы, которая заключается в составлении и заполнении файла – в зависимости от пользовательской информации, алгоритм открывает соответствующий файл с упражнениями, которые разделены по уровням и заполняет пустой файл для плана тренировки. Метод ничего не принимает и ничего не возвращает.

-on\_pushButtonOpen\_clicked (): void – метод предназначен для открытия программы, при нажатии на кнопку «Открыть программу». Метод ничего не принимает и ничего не возвращает.

-on\_save\_triggered (): void – метод предназначен для сохранения программы, в определённое место, куда пожелает пользователь, в пункте меню → *файл* → *Сохранить как...*. Метод ничего не принимает и ничего не возвращает.

-on\_SaveFile\_clicked (): void – метод предназначен для сохранения программы, в определённое место, куда пожелает пользователь, при нажатии на кнопку «Сохранить программу». Метод ничего не принимает и ничего не возвращает.

`-on_open_triggered (): void` – метод предназначен для открытия программы, в пункте меню → *файл* → *Открыть...*. Метод ничего не принимает и ничего не возвращает.

3. *Newprog* – класс, который предназначен для определения поведения программы при взаимодействии пользователя со вторым графическим пользовательским интерфейсом.

Методы:

`-okClicked (): void` – метод предназначен для корректной работы кнопки «Готово» в окне «Application form». Кнопку невозможно нажать, если не заполнены все поля. Метод ничего не принимает и ничего не возвращает.

`-on_pushButton_clicked (): void` – метод предназначен для корректной работы кнопки «Выйти» в окне «Application form». Метод ничего не принимает и ничего не возвращает.

`-dataRecordingText (): void` – метод предназначен для записи информации о данных пользователя в файл «TempDataRecord.txt» Метод ничего не принимает и ничего не возвращает.

## 4 РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ

В данном разделе рассмотрим пошаговую структуру трех произвольно выбранных алгоритмов. Это делается для того, чтобы лучше понять внутреннее устройство программы. Приступим к рассмотрению первого алгоритма.

В качестве первого алгоритма выберем, например, алгоритм вычисления уровня подготовки человека. Этот метод нужен для составления плана тренировок по определенному уровню подготовки человека. Ниже приведен пошаговый разбор данного метода (алгоритма).

1. Начало
2. Если пол мужской, то выполняется следующее:
  - 2.1. По формуле  $\left( \frac{Вес}{Рост^2} \cdot 100 \cdot Возраст \right)$  производится вычисление.
  - 2.2. Если возраст больше, чем 40 лет, то уменьшаем уровень сложности до 4.
  - 2.3. Если уровень сложности, который выбирает пользователь, нормальный, то к уровню добавляем 2.
  - 2.4. Если уровень сложности, который выбирает пользователь, сложный, то к уровню добавляем 4.
  - 2.5. Если уровень сложности, который выбирает пользователь, то уровень увеличиваем до максимального (10).
3. Иначе (т.е, если женщина), то выполняется следующее:
  - 3.1. По формуле  $\left( \frac{Вес}{Рост^2} \cdot 100 \cdot Возраст \right)$  производится вычисление.
  - 3.2. Если возраст больше, чем 40 лет, то уменьшаем уровень сложности до 3.
  - 3.3. Если уровень сложности, который выбирает пользователь, легкий, то от уровня отнимаем 1.
  - 3.4. Если уровень сложности, который выбирает пользователь, нормальный, то к уровню добавляем 1.
  - 3.5. Если уровень сложности, который выбирает пользователь, сложный, то к уровню добавляем 3.

- 3.6. Если уровень сложности, который выбирает пользователь, то уровень увеличиваем до максимального (7).
4. Возвращаем результат.
5. Конец.

Блок-схема алгоритма представлена в приложении В.

В качестве второго алгоритма рассмотрим, алгоритм, который заполняет план тренировок и выводит его на экран. Этот метод нужен для того, чтобы пользователь мог увидеть свой план тренировок на экране.

Примечание!

- 1) Изначально были созданы файлы с набором упражнений отсортированы, по сложности, с первого, по 10 уровень (таких файлов пять). Проверку на открытие файла, не рассматривается в блок-схеме, т.к. и без этого очень большой объем. Начало разбора метода начинается с цикла, который будет открывать каждый из пяти, выше упомянутых файлов и заносить упражнение, которое выбирается в зависимости от уровня подготовки человека (уровень рассчитывался в предыдущем алгоритме).
- 2) Также в программе есть пункт – количество тренировок в неделю (2 или 3), т.к. в цикле *while*, который реализует этот пункт, вложен цикл *for*, который реализует рассматриваемый алгоритм, то в блок-схеме он так же не будет учитываться.

Ниже приведен пошаговый разбор данного метода (алгоритма).

1. Начало
2. Создаем массив (типа `string`) названий файлов с упражнениями (WarmUp.txt, Press.txt, Squats.txt, HorizontParallelBars.txt, Press.txt)
3. Заходим в цикл от первого файла с упражнениями, до последнего
4. Создаем и открываем по очереди файлы с упражнениями (WarmUp.txt, Press.txt, Squats.txt, HorizontParallelBars.txt, Press.txt)

5. Так же каждую итерацию создаем и открываем файл для добавления из файлов, которые были открыты в предыдущем пункте (см. 5.).
6. Создаем итерирующую переменную.
7. Создаем строку `result` (типа `string`).
8. Заходим в цикл `while`.
9. Пока не найдем подходящий нам уровень, который мы высчитывали в первом алгоритме (см. приложении В) и записываем его в строку `result`.
10. Итерируем переменную `count`
11. Вывод результата на экран.
12. Конец.

Блок-схема алгоритма представлена в приложении Г.

Последний очень простой алгоритм – алгоритм записи данных, которые ввел пользователь в файл. Этот метод нужен для хранения информации о человеке. Ниже приведен пошаговый разбор данного метода (алгоритма).

1. Создаем строку, которую инициализируем информацией из анкеты, которую заполнил пользователь.
2. Создаем файл.
3. Открываем файл:
  - 3.1. Если файл не открылся, то выводим информацию об ошибке - `"QMessageBox::information"` и выходим.
  - 3.2. Иначе, записываем данные пользователя в файл, который открылся.
4. Закрытие файла.

Блок-схема алгоритма представлена в приложении Д.

## 5 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

Данное приложение может работать на любом персональном компьютере и на любой операционной системе.

Разработанное приложение просто в использовании. Ниже будут даны описания последовательностей действий, необходимых для корректного взаимодействия с приложением.

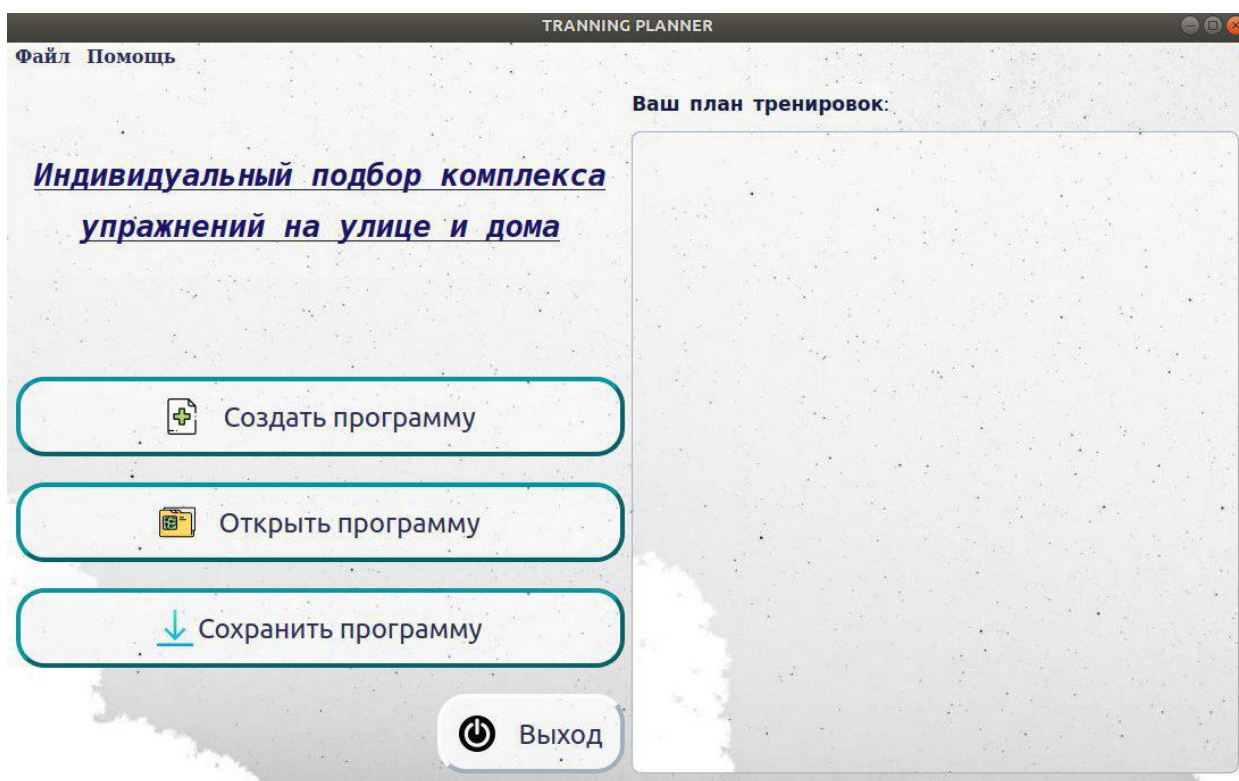


Рисунок 1. Главный экран

На рисунке 1 показан вид приложения после открытия. Для того, чтобы создать программу тренировок нужно нажать на соответствующую кнопку, на рисунке 2 показана эта кнопка стрелкой.



Программа работает только с файлами формата .txt, доступ к другим форматам запрещен.

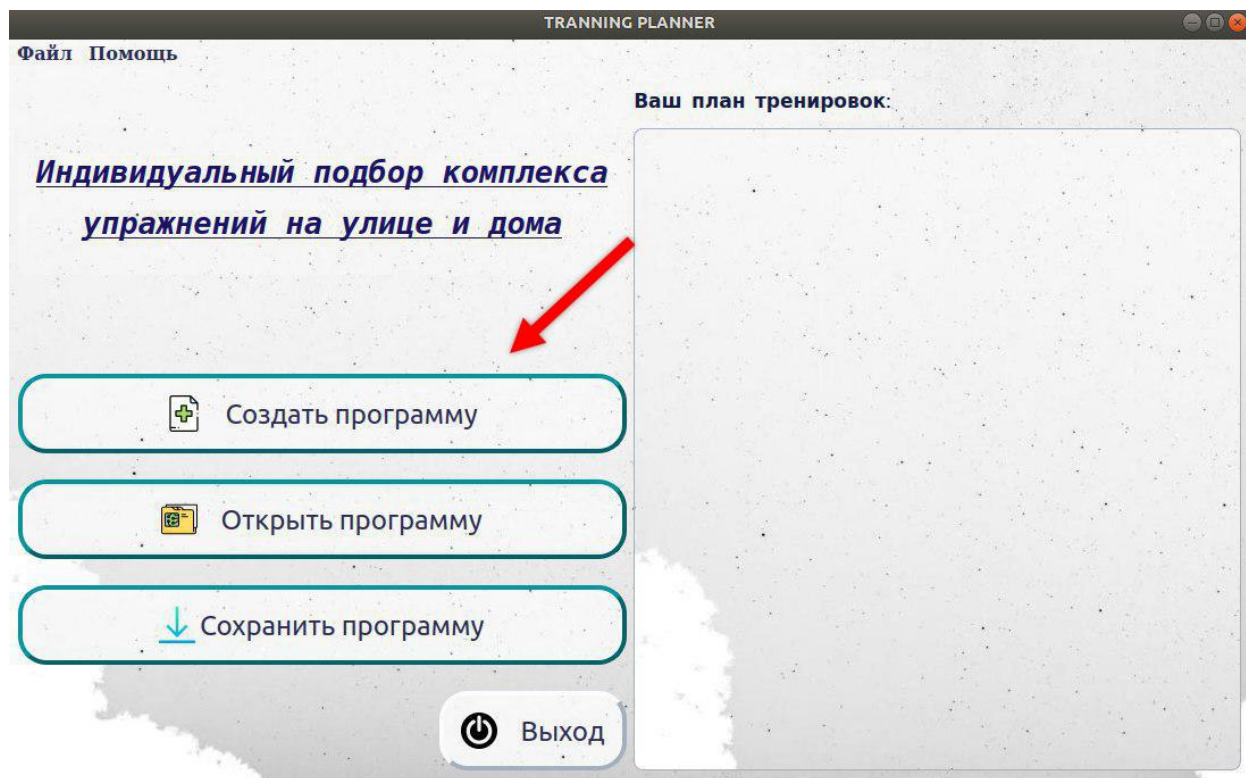


Рисунок 2. Кнопка создания программы

После того, как кнопка «Создать программу» была нажата, на экране появляется новое окно (анкета заполнения), которое необходимо заполнить. В этом окне есть две кнопки:

1. Кнопка «Готово» - кнопка работает, при условии, что все поля анкеты заполнены корректно, и имя (nickname) доступно, т.е. его еще никто не вводил, то нажатие на кнопку возможно, следовательно, программа будет создана, если же хоть одно поле анкеты не заполнено нажатие на кнопку невозможно.
2. Кнопка «Выйти» - кнопка работает, при любых условиях, т.е. эту кнопку можно нажать всегда, но при этом программа не будет создана.

На рисунке 3 показана примерная работа этого окна.

Application form

### Заполните анкету:

Введите свой nickname(имя):

Пол  Возраст  Вес (кг)  Рост (см)

Количество тренировок в неделю

Уровень тренировки



**Заполнение анкеты.**  
**Если не заполнены все**  
**поля анкеты, то нажать на**  
**кнопку "Готово" нельзя!**

Application form

### Заполните анкету:

Введите свой nickname(имя):

МУЖ.  10  30  Рост (см)

Количество тренировок в неделю

Уровень тренировки

Рисунок 3. Анкета заполнения

После успешного заполнения анкеты (см. рисунок 4), на главном экране справа будет записан, подробный план тренировок на недели, в зависимости от того, как была заполнена анкета (см. рисунок 5 и рисунок 5.1). На рисунке 5 видно, что в поле «Ваш план тренировок» записана информация, которую ввел пользователь и пошаговый план тренировок с упражнениями.

Application form

## Заполните анкету:

Введите свой nickname(имя): Vlad

МУЖ. 18 74 182

3

Сложно

Выйти Готово

Рисунок 4. Корректно заполненная анкета

TRAINING PLANNER

Файл Помощь

## Индивидуальный подбор комплекса упражнений на улице и дома

Создать программу

Открыть программу

Сохранить программу

Выход

### Ваш план тренировок:

Имя: Vlad  
Возраст: 18  
Пол: МУЖ.  
Рост: 182  
Вес: 74  
Уровень тренировок: Сложно  
Количество тренировок: 3

#### ВАШ ПЛАН ТРЕНИРОВОК

##### ПЕРВАЯ ТРЕНИРОВОКА:

\*Пробежка .....<10-30 мин> + Разминка .....<5-15 мин>  
\*Алмазные отжимания .....<2-3х10-15>  
\*Приседания с выпрыгиваниям .....<2-3х10-20>  
\*Высокие подтягивания до груди .....<2-3х5-12>  
\*Упражнение на пресс:Велосипед .....<2-3х8-20>

##### ВТОРАЯ ТРЕНИРОВОКА:

\*Пробежка .....<10-30 мин> + Разминка .....<5-15 мин>  
\*Отжимания-V .....<2-3х10-15>  
\*Упражнение-Стульчик у стены .....<1-5 мин>  
\*Подъём с переворотом на турнике .....<2-3х7-15>  
\*Поднятие согнутых ног в висе, к груди .....<2-3х10-20>

##### ТРЕТЬЯ ТРЕНИРОВОКА:

\*Пробежка .....<10-30 мин> + Разминка .....<5-15 мин>  
\*Алмазные отжимания .....<2-3х10-15>  
\*Приседания с выпрыгиваниям .....<2-3х10-20>  
\*Высокие подтягивания до груди .....<2-3х5-12>  
\*Упражнение на пресс:Велосипед .....<2-3х8-20>

Рисунок 5. Готовая программа тренировок



**Ваш план тренировок:**

Имя: Vlad  
 Возраст: 18  
 Пол: МУЖ.  
 Рост: 182  
 Вес: 74  
 Уровень тренировок: Сложно  
 Количество тренировок: 3

**ВАШ ПЛАН ТРЕНИРОВОК**

**ПЕРВАЯ ТРЕНИРОВОКА:**

\*Пробежка .....<10-30 мин> + Разминка .....<5-15 мин>  
 \*Алмазные отжимания .....<2-3x10-15>  
 \*Приседания с выпрыгиваниям .....<2-3x10-20>  
 \*Высокие подтягивания до груди .....<2-3x5-12>  
 \*Упражнение на пресс:Велосипед .....<2-3x8-20>

**ВТОРАЯ ТРЕНИРОВОКА:**

\*Пробежка .....<10-30 мин> + Разминка .....<5-15 мин>  
 \*Отжимания-V .....<2-3x10-15>  
 \*Упражнение-Стульчик у стены .....<1-5 мин>  
 \*Подъём с переворотом на турнике .....<2-3x7-15>  
 \*Поднятие согнутых ног в висе, к груди .....<2-3x10-20>

**ТРЕТЬЯ ТРЕНИРОВОКА:**

\*Пробежка .....<10-30 мин> + Разминка .....<5-15 мин>  
 \*Алмазные отжимания .....<2-3x10-15>  
 \*Приседания с выпрыгиваниям .....<2-3x10-20>  
 \*Высокие подтягивания до груди .....<2-3x5-12>  
 \*Упражнение на пресс:Велосипед .....<2-3x8-20>

Рисунок 5.1. Готовая программа тренировок

Также, на главном экране присутствуют еще три кнопки – «Сохранение программы», «Открытие программы» и «Выход». Точно такие же возможности можно найти в верхнем левом углу – «Файл» (см. рисунок 6.)

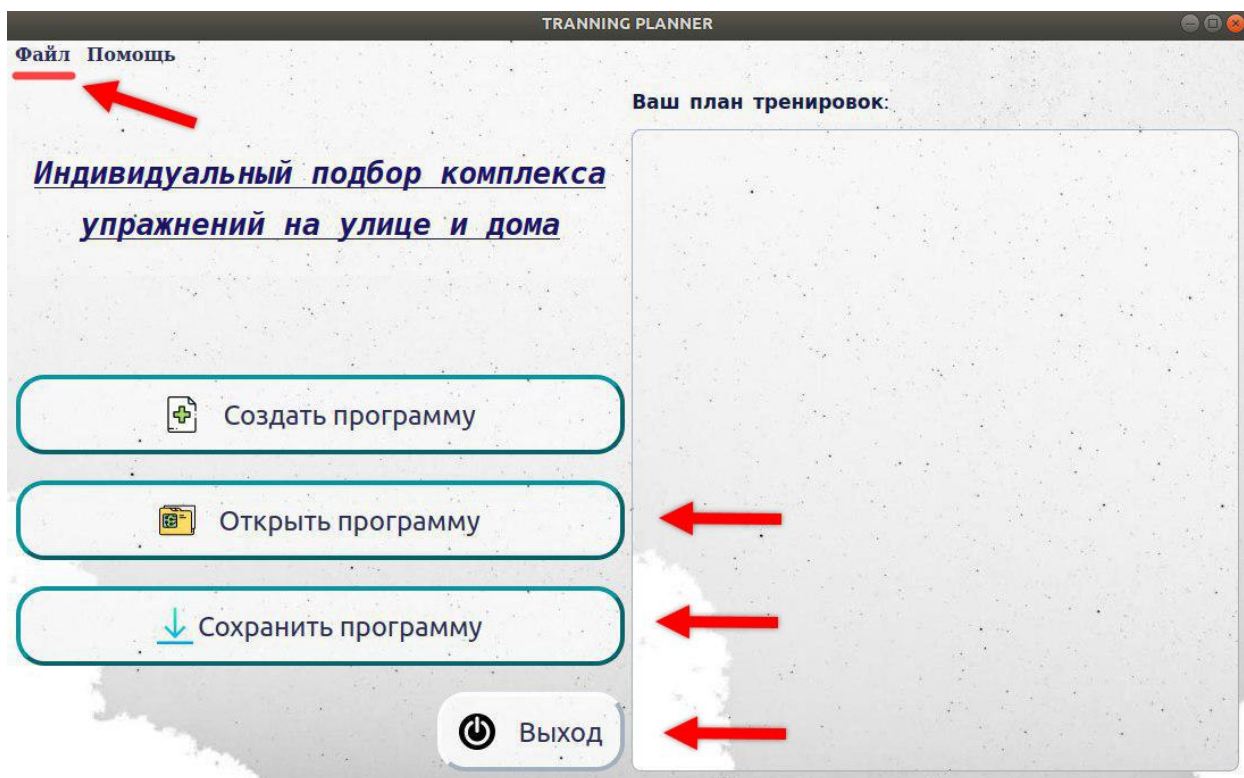


Рисунок 6.

В меню «Файл» можно найти все те же кнопки, также там есть подсказка комбинаций клавиш, с помощью которой можно наживать на все эти кнопки с помощью сочетаний определённых клавиш на клавиатуре (см. рисунок 7.1).



Рисунок 7.1. «Файл»

Так же в меню есть пункт «Помощь» он содержит нужную для пользователя информацию (см. рисунок 7.2)

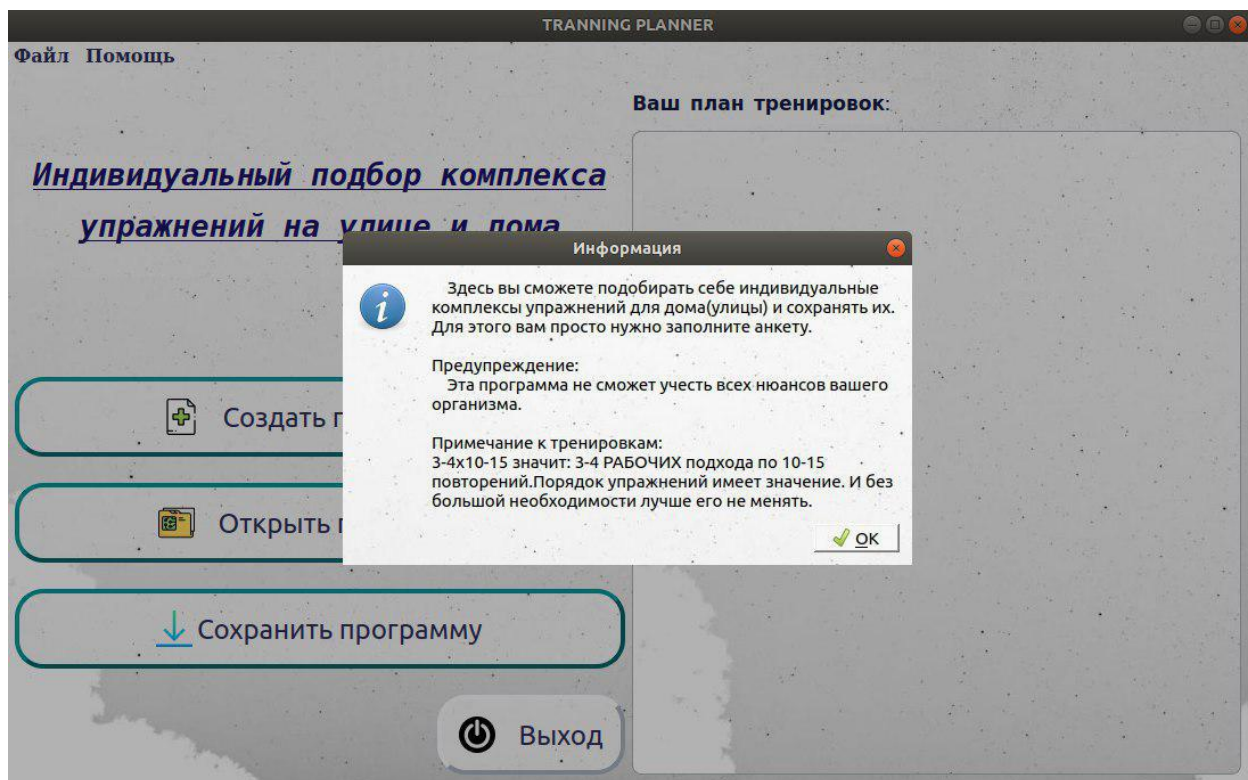


Рисунок 7.1. «Помощь»

При нажатии на кнопку «Выход» или выбор в *файл* → *Выход* , или комбинацией клавиш Ctrl+Q главное окно просто закрывается, и работа программы завершается.

При нажатии на кнопку «Сохранить программу» или выбор в *файл* → *Сохранить как...* или комбинацией клавиш Ctrl+Shift+S на экране появляется окно в котором видны директории на вашем компьютере, вы можете выбрать любое место, куда бы вы хотели сохранить ваш файл (см. рисунок 8).

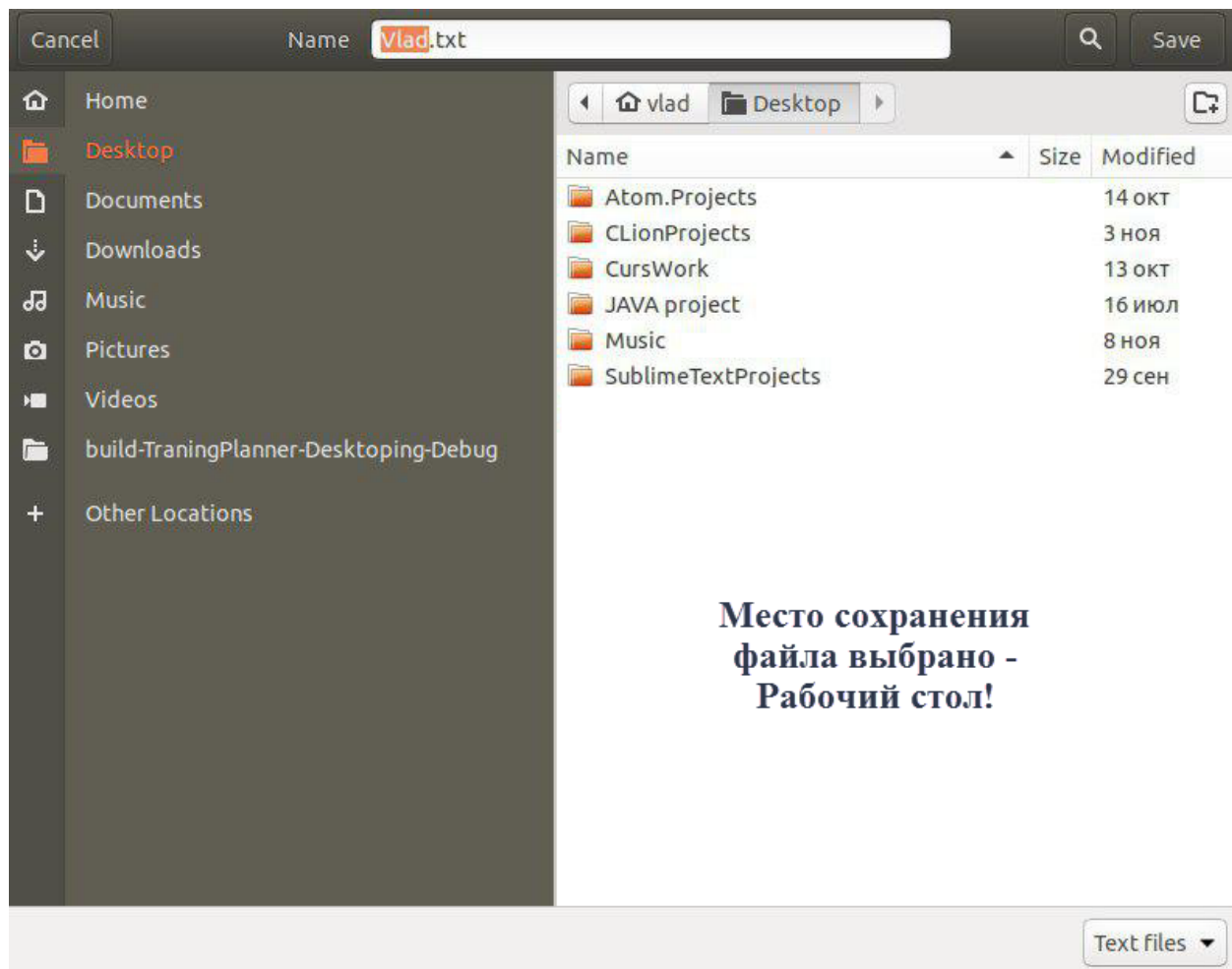


Рисунок 8. Сохранение программы

При нажатии на кнопку «Открыть программу» или выбор в *файл → Открыть* , или комбинацией клавиш Ctrl+O на экране появляется окно в котором видны директории на вашем компьютере, вы можете открыть любой текстовый файл (см. рисунок 9).



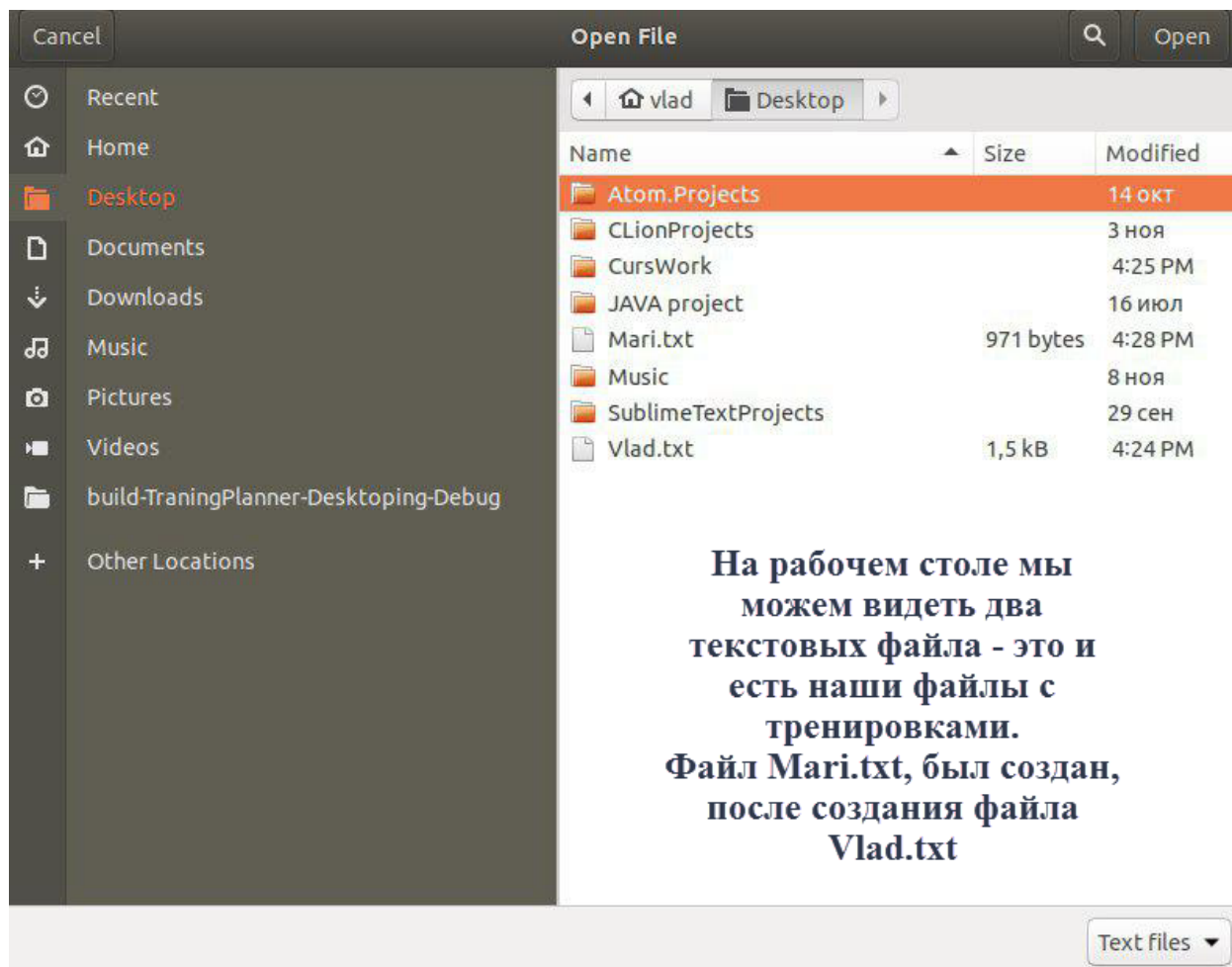


Рисунок 9. Открытие программы

После открытие мы можем видеть тот же результат, что и после создания, на главном экране появляется наш план тренировок с упражнениями. Так как мы уже видели программу тренировок Vlad.txt, давайте откроем программу Mari.txt (см. рисунок 9.1).



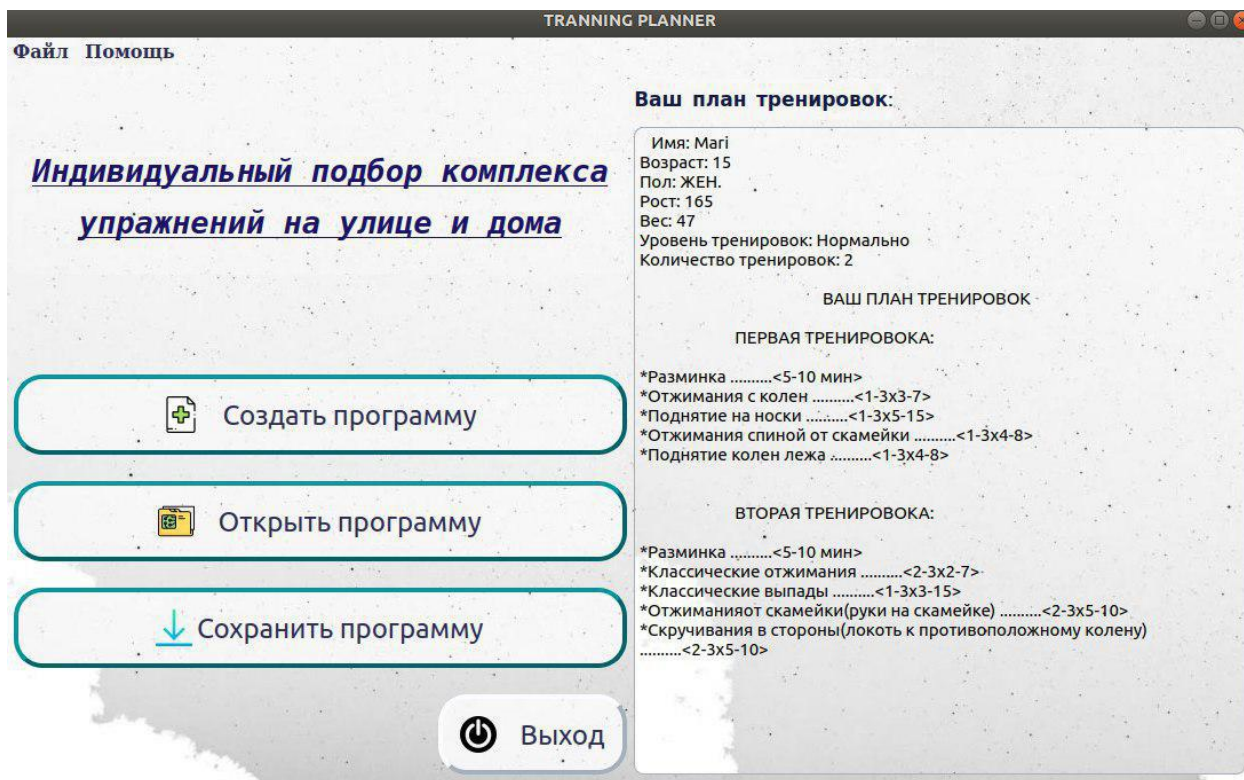


Рисунок 9.1. Открытие программы (продолжение)

Из рисунка 9.1 видно, что анкета была составлена по-другому, следовательно, и программа тренировок будет выглядеть иначе.

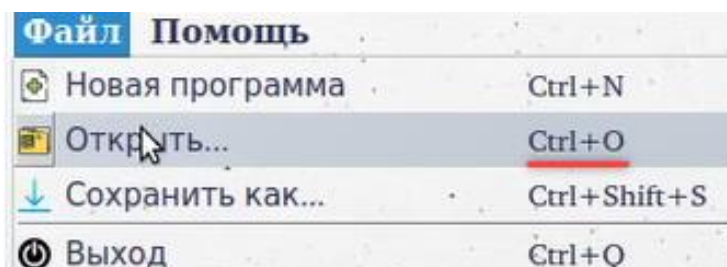
## 6 ТЕСТИРОВАНИЕ

Проведем тестирование некоторых элементов пользовательского интерфейса.

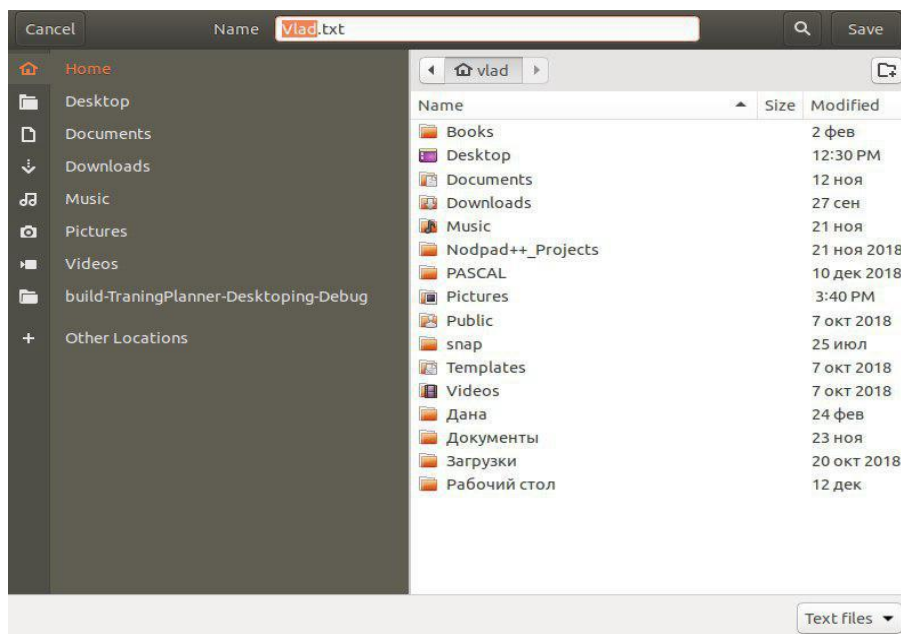
Вид элемента	Действие	Ожидаемый результат	Успешность тестирования
1	2	3	4
<i>файл → Открыть</i>	Ctrl+O	Открытие окна для сохранения	Да
Кнопка «Создать тренировку»	Нажатие на кнопку	Открытие окна для заполнения анкеты	Да
Кнопка «Выход» или <i>файл → Выход</i>	Нажатие на кнопку или Ctrl+O	Выход из программы	Да

Тестирование элемента - *файл* → *Открыть* :

До

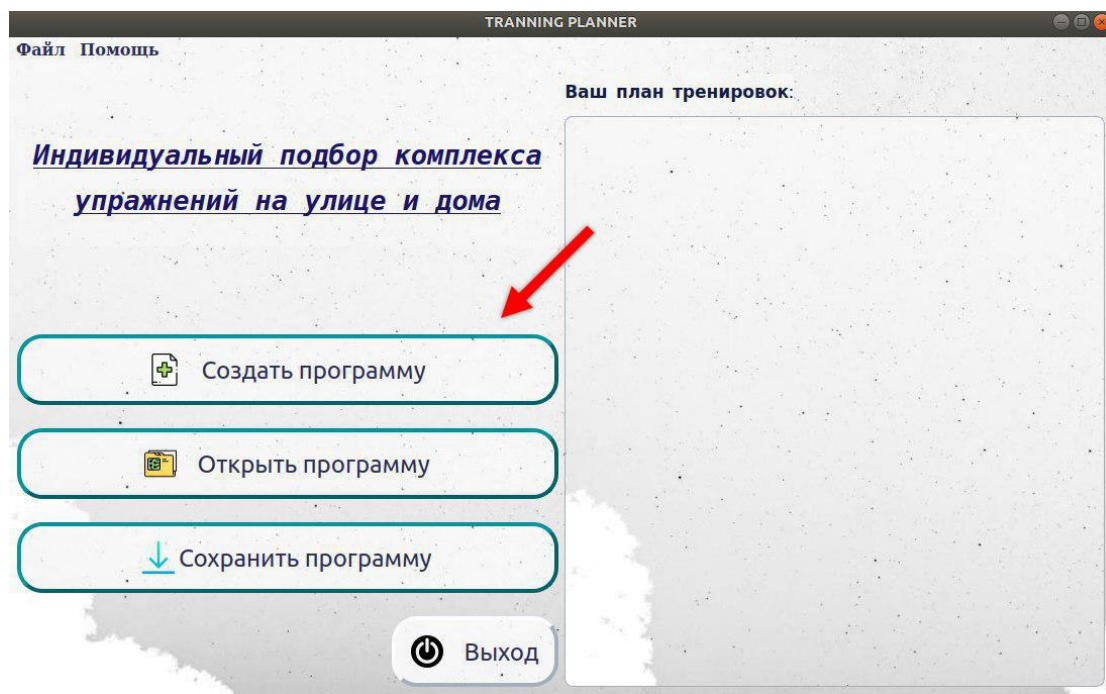


После



Тестирование элемента - «Создать тренировку»:

До

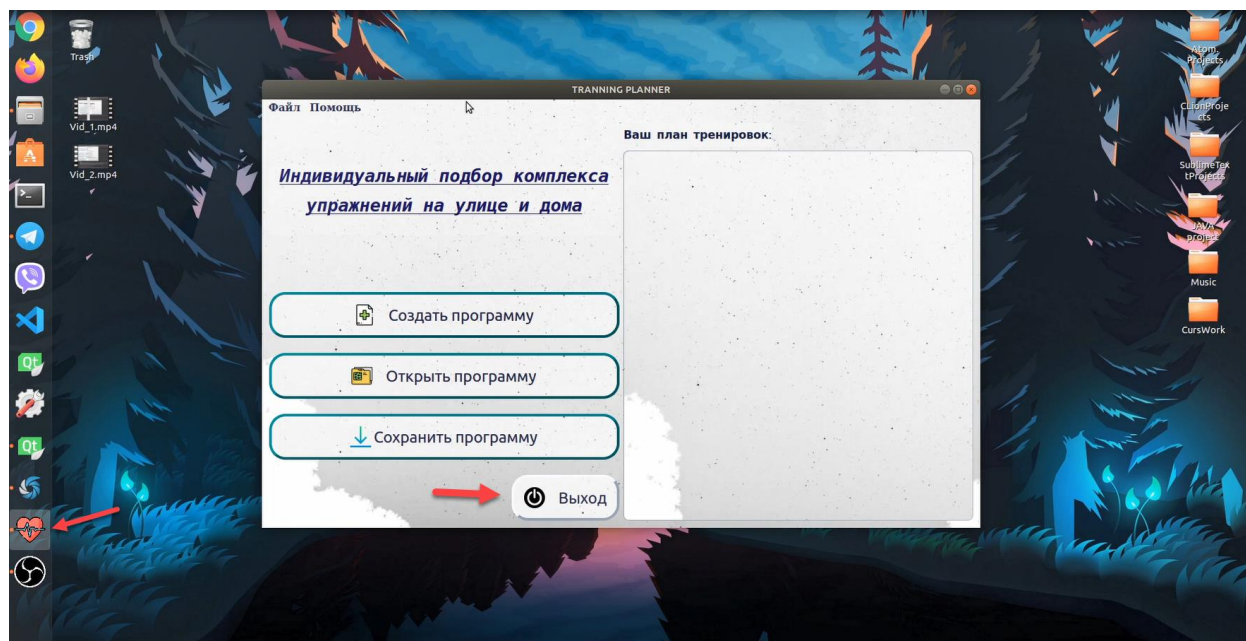


После

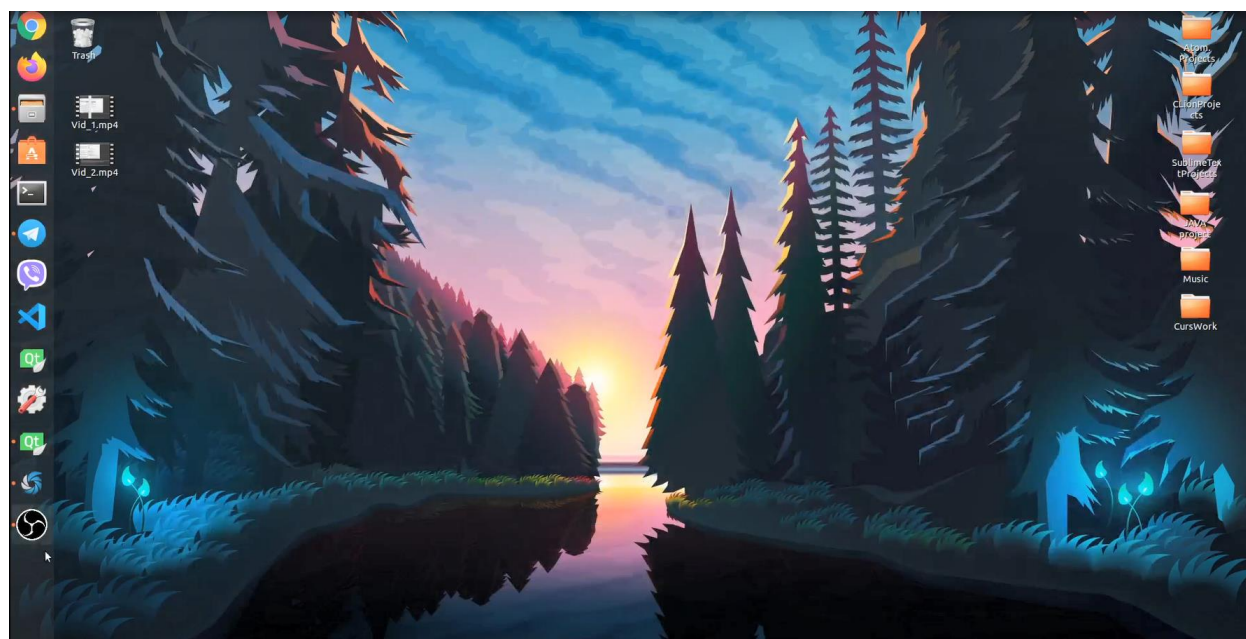
The screenshot shows the 'Application form' dialog box. The title bar reads 'Application form'. The main heading is 'Заполните анкету:'. Below the heading, there are several input fields: 'Введите свой nickname(имя):' followed by a text input field, and four dropdown menus for 'Пол', 'Возраст', 'Вес (кг)', and 'Рост (см)'. Below these are two more dropdown menus for 'Количество тренировок в неделю' and 'Уровень тренировки'. At the bottom, there are two buttons: 'Выйти' and 'Готово'.

Тестирование элемента - «Выход» или *файл* → *Выход* :

До



После



Вывод:

по результатам тестирования видно, что программа выполняет работу, которую хочет сделать пользователь, до ожидаемого результата. Бесплезные или неработоспособные элементы отсутствуют, следовательно, приложение соответствует исходным требованиям.



## ЗАКЛЮЧЕНИЕ

Во время выполнения курсовой работы были выполнены все поставленные задачи:

- 1) создание простого и понятного в использовании приложения, которое будет быстро и более-менее хорошо составлять план тренировок, а также совершать не сложные операции над ними;
- 2) использование объектно-ориентированного программирования;
- 3) рассмотрение и знакомство с кроссплатформенным фреймворком Qt.

В ходе работы было создано кроссплатформенное приложение, были проведены все необходимые тесты для проверки работоспособности программы, а также продемонстрировано применение программы. Конечно, этот «Планировщик тренировок» не самый лучший и не учитывает все возможные качества и уровень человека, но не существует калькулятора, который смог бы учесть и посчитать все возможные характеристики, уровень и качества человека. Несмотря на это, главная задача, курсового проекта была выполнена – создание простого «Планировщика тренировок»

## СПИСОК ЛИТЕРАТУРЫ

- [1] Документация <https://doc.qt.io/>.
- [2] Макс Шлее “Qt 5.3 Профессиональное программирование на C++”.
- [3] Герберт Шилдт “Базовый курс C++”.
- [3] Жасмин Бланшет, Марк Саммерфилд. Qt 4: программирование GUI на C++.



ПРИЛОЖЕНИЕ А  
(Обязательное)

Структурная схема программы

ПРИЛОЖЕНИЕ Б  
(Обязательное)

Диаграмма классов

ПРИЛОЖЕНИЕ В  
(Обязательное)

Блок-схема алгоритм 1

ПРИЛОЖЕНИЕ Г  
(Обязательное)

Блок-схема алгоритма 2

ПРИЛОЖЕНИЕ Д  
(Обязательное)

Блок-схема алгоритма 3