ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ ГОРОДА МОСКВЫ

Государственное Бюджетное Общеобразовательное Учреждение города Москвы «Школа №1502 «Энергия»

**ПРОЕКТ**

по информатике на тему

**«Разработка приложения для изучения технического английского языка»**

Выполнил: ученик 10 класса Т  
Коныгина М.А

Научный руководитель: учитель информатики  
Кузьмина Е.Ф.

Москва, 2024

ОГЛАВЛЕНИЕ

[ВВЕДЕНИЕ 3](#_Toc156160122)

[1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ 4](#_Toc156160123)

[1.1 Технический английский 4](#_Toc156160124)

[1.2 Языки программирования 7](#_Toc156160125)

[1.3 Библиотеки Python 9](#_Toc156160126)

[2. ТЕХНОЛОГИЧЕСКИЙ РАЗДЕЛ 11](#_Toc156160127)

[2.1 Методы и библиотеки 11](#_Toc156160128)

[2.3 Руководство для пользователя 13](#_Toc156160129)

[2.3 Формат тестирования 15](#_Toc156160130)

[2.4 Теория 17](#_Toc156160131)

[Список литературы: 20](#_Toc156160132)

# ВВЕДЕНИЕ

В современном мире начать собственное дело без технологий IT практически невозможно. Иногда, это простые программы которые упрощают и систематизируют рабочий процесс, а иногда сложные ПО, для которых нужна поддержка специально обученного человека. Даже если ваша деятельность не связана с технологическими решениями, вы все равно будете зависеть от IT. Английский язык в профессии программиста стоит на втором месте после специализированных в сфере знаний. Многие начинающие утверждают, что английский язык для программистов совершенно не важен и что главное, добыть квалификацию и найти заинтересованных в своей работе клиентов. На самом деле это заблуждение, ведь все технологические новинки и инновационные ПО сначала появляются в США и Европе, а значит, на английском языке.

Программист со знанием английского языка имеет явные преимущества над теми, кто его не знает, ведь у него есть возможность с первых моментов ориентироваться в нововведениях и современных новинках.

С развитием компьютерных технологий обучение языкам стало гораздо удобнее и интерактивнее. Интернет позволил появиться такой форме образования, как дистанционное обучение. Среди электронных форм обучения одно из ведущих мест в последние годы занимают компьютерные тренажеры. Поэтому разработка тренажера для изучения технического английского языка **актуальна** в наши дни.

**Цель**: разработка обучающего приложения для изучения технического английского языка на языке Python.

Для достижения цели необходимо решить следующие **задачи**:

* изучить базовые темы в английском языке;
* отобрать и систематизировать материал;
* изучить возможности языка Python, библиотеки PyQT5 и QT Designer;
* продумать функционал и интерфейс для приложения;
* разработать программный код;
* произвести отладку и корректировку проекта.

# 1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

1.1 Технический английский

Технический английский язык — это специализированная разновидность английского языка, используемая в области науки, техники и инженерии. Он содержит специфическую терминологию и грамматические конструкции, которые помогают точно и четко выражать технические и научные идеи. Сложность понимания технического стиля заключается в применении инверсии — обратного порядка слов, а также использования сложных аббревиатур и длинных цифр. Статьи на узкопрофильные темы кажутся непонятными простому человеку, даже если они написаны на русском языке. Поэтому для изучения таких терминов нужно овладеть базовым английским, постепенно усложняя программу. Сложность специализированного языка заключается даже не в терминах, а в умении работать с разными значениями английских слов. Неправильный перевод может исковеркать весь смысл текста, поэтому нужно знать и сам технический предмет, чтобы понять контекст.

Когда дело доходит до программирования, английский язык предпочтительнее, потому что он разрушает коммуникационные барьеры, которые могут существовать между разными программистами. Английский является наиболее объединяющим языком. Также большинство англоязычных стран были «домами» информатики. Вначале информатика преподавалась в США, Великобритании и Канаде. Другие страны начали сосредотачиваться на этом предмете годы спустя. Каждому IT-специалисту важно владеть английским языком на хорошем уровне. Для этого есть много причин:

1. Большинство языков программирования основано на ключевых словах английского происхождения. Без знания ключевых слов понимать и писать коды для вас будет очень долгими и сложным процессом.
2. Практически все справочные материалы и технические задания пишутся на английском.

Документации для различных языков программирования всегда пишутся на английском, хоть мы и можем найти переводы в интернете, но для лучшего понимаю оригинал всегда лучше.

1. Программистам нужно быть в курсе всех новостей и обновлений, а новейшие книги и статьи в IT-сфере публикуются на английском.
2. В своей работе IT-специалисты периодически сталкиваются со сложными задачами и некоторыми проблемами, решение которых можно найти в англоязычной интернет-среде.
3. Понимание сленга, используемого программистами.

Часто слушая речь работников IT сферы, человеку, не ознакомленному со сленгом, обычно образованным от английских слов, может быть проблематично понять, о чем говорят его коллеги.

Пример:

Fork (вилка) - новое приложение, разработанное на основе существующего. Когда приложение "разветвляется", оно создает новую отдельную программу, а не новую ветвь разработки.

Как можно понять из приведенных выше примеров, уверенное владение техническим английским не только позволяет специалисту существенно вырасти в профессиональном плане, но и дает ему неоспоримое преимущество на рынке труда.

Типы терминов, используемых программистами:

1. Ключевые слова – в языках программирования слово, имеющее специальное значение.

Примеры:

String - тип данных, используемый для представления текста.

Class - это шаблонное определение методов и переменных в объекте определенного типа.

While - используется для повторения определенного блока кода неизвестное количество раз, пока не будет выполнено условие.

1. Термины, используемые в документации для описания того, как язык/ библиотека работают. Такие термины одинаковы для всех языков программирования.

Примеры:

Function definition - определение функции.

Initializing the module - инициализация модуля.

1. Термины, используемые при разработке.

Примеры:

Backup -копия одного или нескольких файлов, созданных в качестве альтернативных в случае потери исходных данных или невозможности их использования.

Debugging - поиск и исправление ошибок в программе.

Encoding - процесс преобразования данных из одной формы в другую.

## 1.2 Языки программирования

Для разработки приложений используются разные языки программирования. Давайте рассмотрим некоторые из них.

C++ - считается отличным выбором для разработки мобильных приложений. Это универсальный объектно-ориентированный язык с общими и низкоуровневыми функциями управления памятью.

Плюсы:

* Быстрый и стабильный, имеет обширный набор готовых к использованию встроенных библиотек и компиляторов;
* если вы знаете C ++, вы можете быстро выучить JAVA, C, C #;
* можно получить полный контроль, и приложение может работать на любой платформе и использовать преимущества любого оборудования.

Минусы:

* Сложный синтаксис, небольшая стандартная библиотека;
* Самый сложный язык программирования для освоения;
* Объектная ориентация весьма важна по сравнению с другими языками.

JavaScript - лучше всего подходит для кросс-платформенной разработки приложений, мобильных приложений, используемых для различных платформ, и просмотра [веб](https://notissimus.com/luchshie-frejmvorki-dlya-veb-razrabotki-kotorye-stoit-vybrat-v-2023-godu/)-страниц.

Плюсы:

* Универсальный и гибкий язык, который может использоваться многими способами благодаря Node.js;
* Более быстрый просмотр на стороне клиента и не требует компиляции;
* Простой в работе и простой в управлении.

Минусы:

* Уязвим и в некоторых случаях может быть использован в злонамеренных целях;
* Сценарии на стороне сервера всегда обеспечивают одинаковый результат, а на стороне клиента это немного непредсказуемо;
* В некоторых случаях вы можете столкнуться с некоторыми проблемами с поддержкой браузера.

Python - широко распространенный язык программирования используется для разработки веб-приложений, настольных приложений, медиа-инструментов, машинного обучения, сетевых серверов и многого другого. Примерами Python являются YouTube, Instagram и Pinterest.

Плюсы:

* Быстрый, простой в использовании, быстрый в развертывании, простой в освоении и отлично читаемый;
* Язык программирования высокого уровня, используемый для анализа, вычисления научных и числовых данных, а также для разработки программного обеспечения;
* Интерактивный язык, используемый для создания масштабируемых приложений и поддержки приложений с графическим интерфейсом.

Минусы:

* Медленное время выполнения;
* Не лучший выбор для задач, требующих большого объема памяти;
* Не так уж хорошо с современной разработкой мобильных приложений.

Я буду использовать Python так он легок и быстр в освоении, а также имеет нужные мне библиотеки.

## 1.3 Библиотеки Python

Библиотеки — это готовые модули кода, которые используют разработчики, чтобы не писать один и тот же код несколько раз. Python имеет большое количество библиотек.

Стандартные библиотеки Python — это модули кода, которые доступны разработчикам без дополнительной установки

Например, библиотека math в Python дает возможность выполнять самые разные математические функции. Библиотека os Python необходима для работы с файловой системой компьютера и операционной системой. Другие стандартные библиотеки — random и datetime. Первая дает инструменты для работы со случайными числами, а вторая незаменима для программ, которые оперируют временем и датами.

Стандартные библиотеки Python не требуют отдельного подключения. Чтобы использовать их функционал, достаточно просто в начале программы написать import и название библиотеки.

Рассмотрим другие библиотеки Python.

Pandas — это библиотека обработки и анализа данных, которая предоставляет широкий спектр инструментов для манипуляции данными, включая чтение и запись данных из различных источников, манипулирование индексами и колонками, объединение и разделение данных и многое другое. Numpy — это фундаментальная библиотека для работы с многомерными массивами (или “векторами”) в Python. С помощью Numpy вы можете выполнять различные математические операции над этими массивами, включая сложение, вычитание, умножение, а также более сложные операции, такие как линейная регрессия или PCA (метод главных компонент). [Python Imaging Library](https://pillow.readthedocs.io/en/stable/), или PIL – это бесплатная библиотека Python, которая позволяет интерпретатору Python обрабатывать изображения. Иными словами, PIL дает возможность обрабатывать, открывать и сохранять различные форматы изображений в Python. [Pygame](https://www.pygame.org/) – это бесплатная библиотека Python с открытым исходным кодом. Она предназначена для разработки мультимедийных приложений на Python, в частности, двумерных игр. И как результат, она популярна среди как обычных, так и профессиональных разработчиков игр на Python.

# 2. ТЕХНОЛОГИЧЕСКИЙ РАЗДЕЛ

## 2.1 Методы и библиотеки

Используемые библиотеки:

Для разработки данного приложения будет использоваться библиотеку PyQt в Python. PyQt — это библиотека Python для создания приложений с графическим интерфейсом с помощью инструментария Qt. Кроме того, позволяет работать с базами данных и другими внешними источниками данных. PyQt является открытым исходным кодом и распространяется под лицензией GPL. Для создания интерфейса в приложении используется Qt Designer. Он позволяет создавать графические интерфейсы пользователя при помощи ряда инструментов. Существует панель инструментов «Панель виджетов», в которой доступны для использования элементы интерфейса — виджеты, такие как, например, «выпадающий список» ComboBox, «поле ввода» LineEdit, «кнопка» PushButton и многие другие. Каждый виджет имеет свой набор свойств, определяемый соответствующим ему классом библиотеки Qt. Свойства виджета могут быть изменены при помощи «Редактора свойств». Для каждого класса свойств виджета существует свой специализированный редактор.

Для осуществления регистрации используется база данных созданная в SQLite. Это однофайловая СУБД, которая не имеет сервера и позволяет хранить всю базу локально на одном устройстве. СУБД - это система управления базами данных, которая позволяет хранить, изменять и извлекать данные. SQLite имеет ряд преимуществ, которые делают ее привлекательной для разработчиков. Во-первых, она очень компактна, что позволяет использовать ее на устройствах с ограниченными ресурсами. Во-вторых, она быстро работает, так как не требует дополнительных процессов или серверов. В-третьих, SQLite очень проста в использовании, что делает ее идеальной для начинающих разработчиков. Для работы с базами данных в Python используется встроенная библиотека sqlite3. Для установки соединения нужно указать название базы данных, к которой требуется подключиться.

Термины и их значения записываются в csv файлы. Python имеет встроенную библиотеку csv, которая предоставляет функции для чтения и записи данных в формате CSV.

Используемые методы:

* MainWindow() – открывает главный экран для входа или регистрации;
* RegWindow() – открывает окно регистрации;
* SignWindow() – открывает окно для входа в аккаунт;
* Testing() – запускает пустой экран с меню, в котором будут показываться все виджеты;
* MainPage() – виджет главной страницы приложения;
* DevelopTestWindow() – виджет теста с словами для разработки;
* KeywordsTestWindow() – виджет теста с ключевыми словами;
* ComputerTestWindow() – виджет теста с частями компьютера;
* TermsTestWindow() – виджет теста с усложненными терминами;
* KeywordsWindow() – виджет теории с ключевыми словами;
* DevelopWindow() – виджет теории с словами для разработки;
* TermsWindow() – виджет теории с усложненными терминами;
* ComputerWindow() – виджет теории с частями компьютера.
* DictionaryWindow() – виджет со словарем
* ChooseWord() – окно с добавлением нового слова в словарь

## 2.3 Руководство для пользователя

Для открытия проекта требуется открыть файл «TechEng.exe».



Рисунок 1. Иконка приложения

*Для корректной работы приложения требуется:*

* операционная система «Windows 7/8/10», Linux или macOS;
* место на диске 100Мб;
* 64-разрядный процессор и операционная система;
* подключение интернета для скачивания;
* приложение открытия exe файлов для Linux и macOS (например, Wine)

2.2 Структура

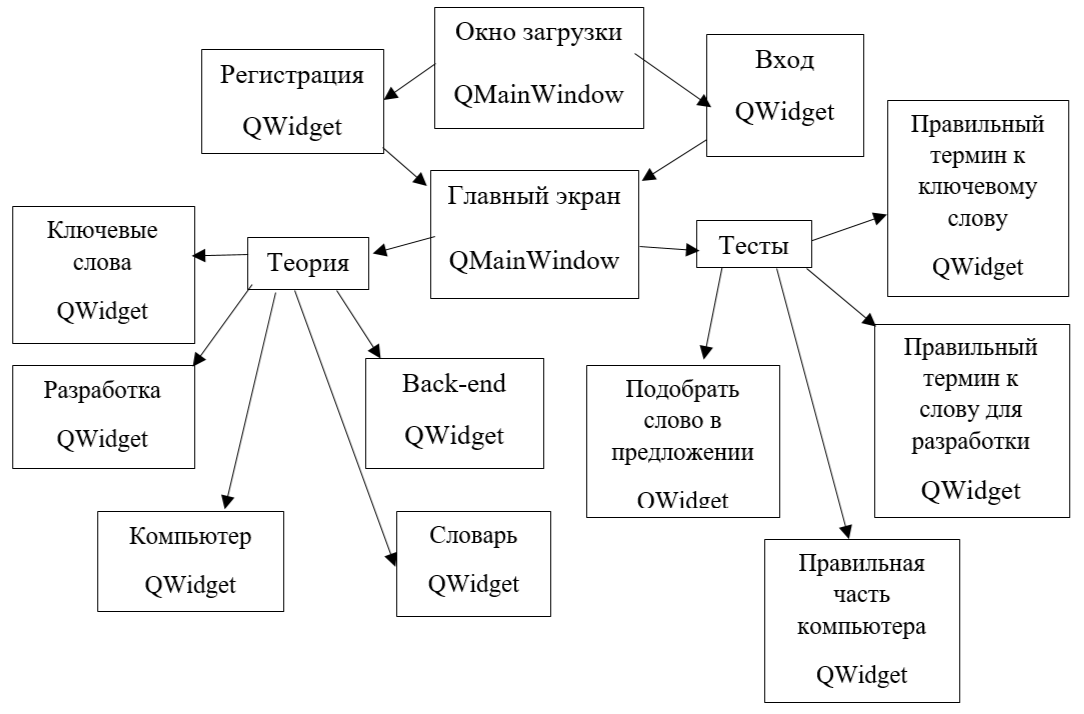


Рисунок 2. Структура приложения

При загрузке приложения появляется окно в котором пользователь должен выбрать “Регистрацию” или “Вход”. Для регистрации требуется логин из более чем 3 символов и пароль (рис. 3).

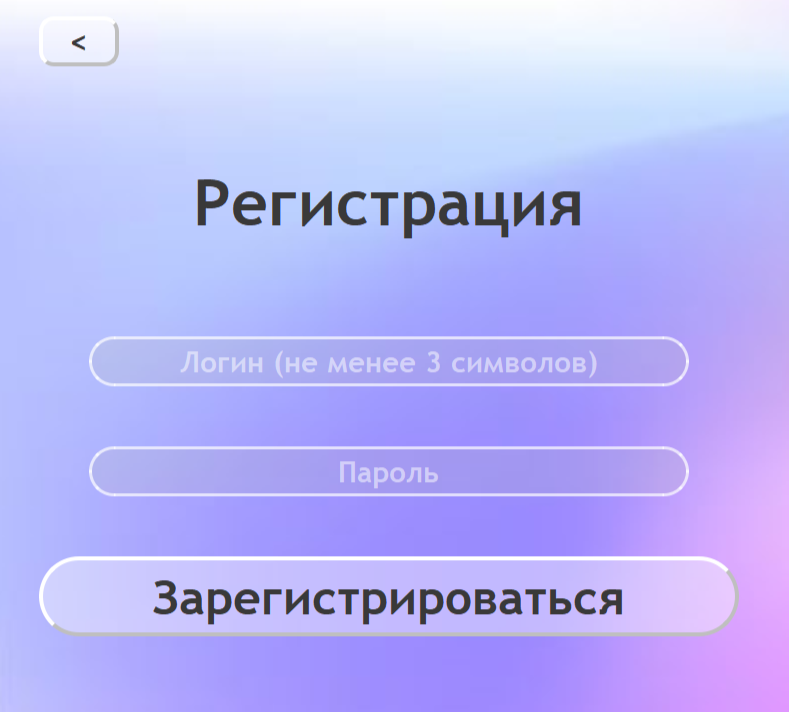


Рисунок 3. Регистрация

После входа пользователь видит главный экран предложения. На нем можно увидеть ваш логин, количество баллов за тестирование и теорию, слово дня (рис. 4).

В меню в верхней части экрана находятся вкладки, где можно выбрать тесты и теорию для изучения, а также вернутся на главный экран.

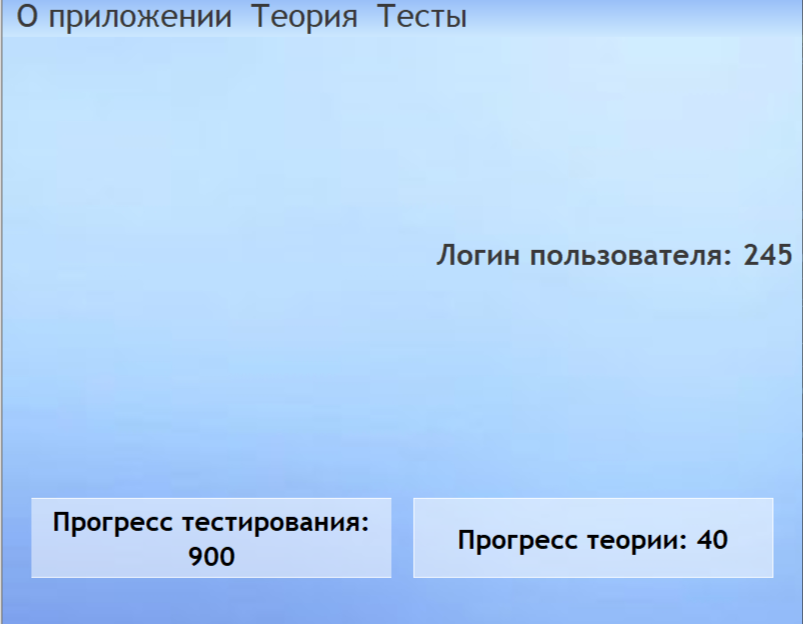


Рисунок 4. Главный экран

## 2.3 Формат тестирования

Для начала теста нужно нажать на кнопку “Далее”. Для выбора ответа нужно нажать кнопку номера ответа. Она загорится красным или зеленым, соответствии с правильностью.

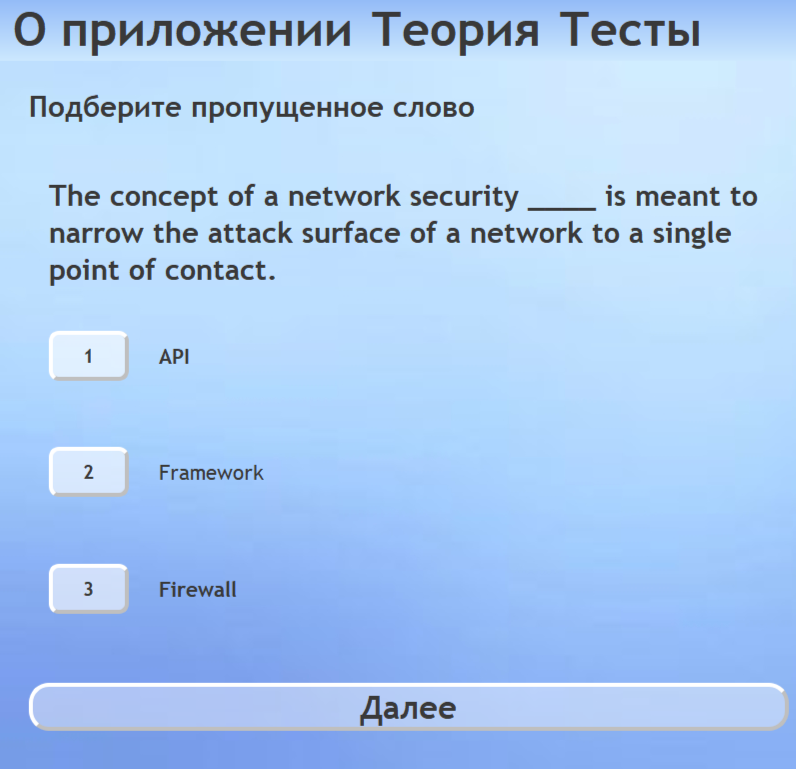


Рисунок 5. Тестирование

После прохождения теста высвечивается окно с количеством баллов. Они записываются в базу данных. (рис. 6)

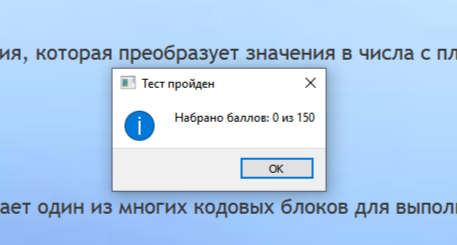


Рисунок 6. Окно окончания тестирования

При прохождении тестирования с элементами компьютера, пользователю нужно вести правильное название элемента на английском в окно с номером этого элемента. (рис. 7)



Рисунок 7. Тестирование элементов компьютера

## 2.4 Теория

При прочтении теории пользователь пролистывает список терминов или просматривает картинки. После ознакомления нужно нажать на кнопку “Прочитано”. Эта часть теории запишется в базу данных пользователя и повторное нажатие кнопки не добавит баллов.

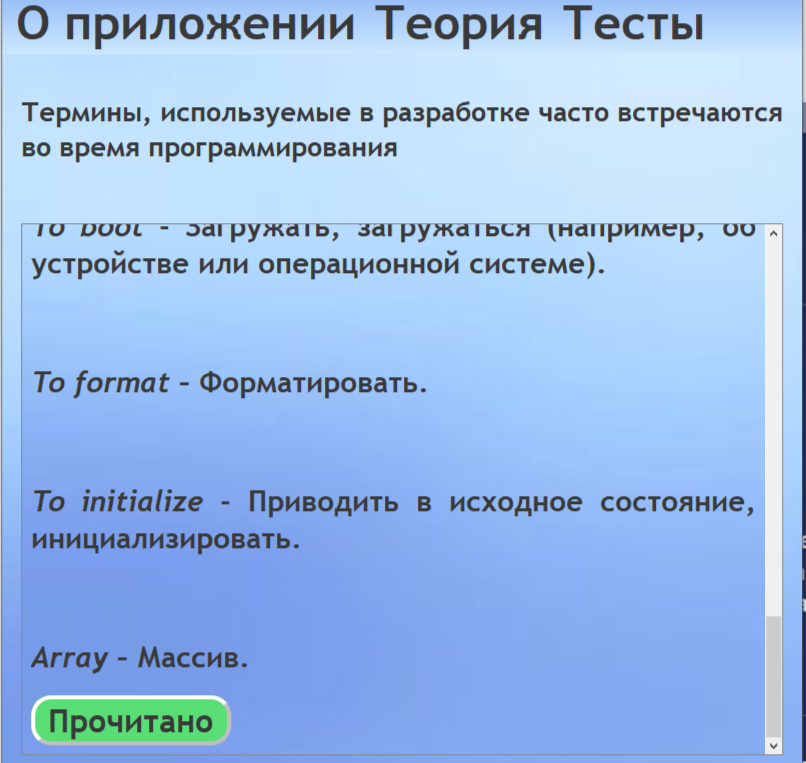


Рисунок 8. Теория

Также приложение имеет словарь всех изученных слов (рис. 9). При нажатии на слово, слева появляется его значение.

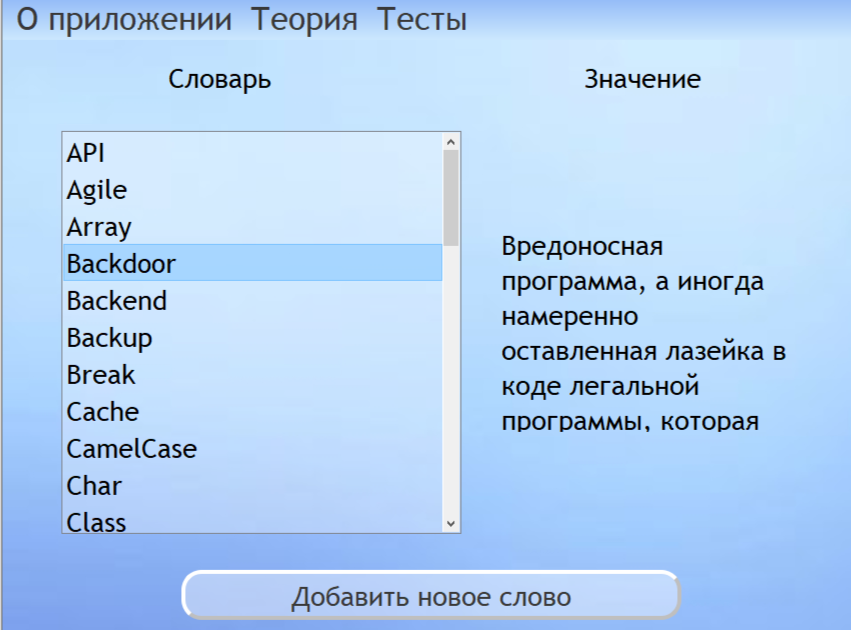


Рисунок 9. Словарь

Добавлять новые слова в словарь можно по кнопке “Добавить новое слово”. Для этого нужно записать слово и его значение, затем выбрать тип слова и нажать кнопку “Добавить слово” (рис. 10)

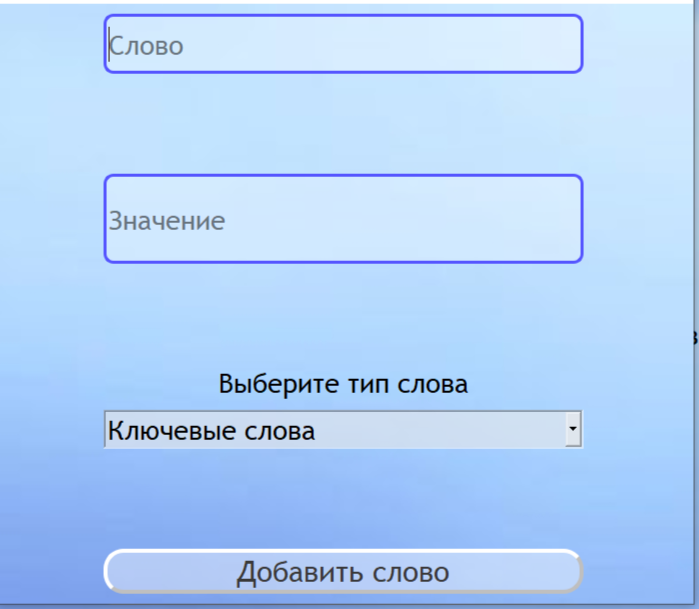


Рисунок 10. Добавить слово в словарь

Заключение

В результате проделанной работы было создано приложение, которое помогает программистам изучить и понять разные технические термины. Были решены все поставленные задачи. Было изучено большое количество технических терминов и информации о библиотеках языка Python. Был создан интерфейс и система тестирования пройденной информации. Проект может быть использован в образовательных учреждениях, на домашних компьютерах учеников или программистов.

В процессе работы было решено большое количество задач: ошибки в коде, неправильное отображение изображений, ошибки в использовании баз данных и открытии окон, отображение приложения на разных ОС. В проекте были использованы решения, которые используются при разработку приложений с использованием Qt.

В будущем проект может быть доработан: путём добавления новых заданий и теории, рейтинга пользователей и усложненной регистрации. Проект, в виду все более частого использования программирования становится всё более актуальным.

# Список литературы:

1. English for Information Technology / Olejniczak Maja, 2011
2. English for Telecoms and Information Technology IT [Oxford - Express Series] / Tom Ricca-McCarthy, Michael Duckworth, 2007
3. Python – это просто / Нилаб Нисчал, 2022
4. Python 3 и PyQt 5. Разработка приложений / Дронов Владимир Александрович, Прохоренок Николай Анатольевич, 2013
5. Изучаем Python. Программирование игр, визуализация данных, веб-приложения. / Эрик Мэтиз, 2017
6. Английский для IT. Как учиться программисту + 84 ресурса [Электронный ресурс]. Режим доступа URL: <https://englex.ru/english-for-it-specialists/>
7. 16 лучших языков программирования для разработки мобильных приложений 2023 [Электронный ресурс]. Режим доступа URL: <https://notissimus.com/16-luchshih-yazykov-programmirovaniya-dlya-razrabotki-mobilnyh-prilozhenij-2023/#Python>
8. PyQt6 - полное руководство для новичков [Электронный ресурс]. Режим доступа URL: <https://habr.com/ru/companies/skillfactory/articles/599599/>
9. Python and PyQt: Creating Menus, Toolbars, and Status Bars [Электронный ресурс]. Режим доступа URL: <https://realpython.com/python-menus-toolbars/#connecting-signals-and-slots-in-menus-and-toolbars>
10. SQLite [Электронный ресурс]. Режим доступа URL: <https://blog.skillfactory.ru/glossary/sqlite/>
11. TechTerms. The Computer Dictionary [Электронный ресурс]. Режим доступа URL: <https://techterms.com/browse/>
12. Английский язык для программиста: how to [Электронный ресурс]. Режим доступа URL: <https://blog.skillfactory.ru/anglijskie-slova-i-termini-it/>
13. Как работать с модулем sqlite3 в Python [Электронный ресурс]. Режим доступа URL: <https://sky.pro/media/kak-rabotat-s-modulem-sqlite3-v-python/>
14. Яндекс Лицей [Электронный ресурс]. Режим доступа URL: https://lyceum.yandex.ru/