

УТВЕРЖДАЮ

Руководитель практики от Университета

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

*(должность, ФИО)*

МП \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(Подпись)*

Дата « \_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 202\_ год

**ОТЧЕТ**

**ПО УЧЕБНОЙ ПРАКТИКЕ**

ПМ.05 Проектирование и разработка информационных систем

|  |  |
| --- | --- |
| Кривошеин Дмитрий Васильевич | |
| *(Ф.И.О. обучающегося)* | |
| 09.02.07 Информационные системы и программирование | |
| *(специальность)* | |
|  | |
| Учебная группа | ИСПк-204-52-00 |
|  |  |
| Место прохождения практики | ФГБОУ ВО «Вятский государственный университет», |
| Колледж ВятГУ | *(наименование организации, структурного подразделения организации)* |
|  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Итоговая оценка: |  | | | | |
| Руководитель  практики от университета |  |  |  |  |  |
|  | *(дата)* |  | *(подпись)* |  | *(Ф.И.О.)* |

Киров, 2024 г.



ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ НА ПРАКТИКУ

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Ф.И.О. обучающегося | | Кривошеин Дмитрий Васильевич | | | | | |
| Специальность | | 09.02.07 Информационные системы и программирование | | | | | |
| Учебная группа | | ИСПк-204-52-00 | | | | | |
| Вид практики | | учебная практика | | | | | |
| Тема индивидуального задания | | Разработка интерактивного учебника по английскому языку | | | | | |
| Сроки прохождения практики | | с | 15.01.2024 | по | 19.06.2024 | | |
| Место прохождения практики | | ФГБОУ ВО «Вятский государственный университет», Колледж ВятГУ | | | | | |
|  | | (наименование организации, структурного подразделения организации) | | | | | |
| № | Виды работ, выполняемых обучающимися во время практики | | | | | Объем работ (час) | Формируемые компетенции |
| 1 | Пройти инструктаж по ознакомлению с правилами внутреннего трудового распорядка, охраны труда, техники безопасности, противопожарной безопасности, санитарно-эпидемиологическими правилами и гигиеническими нормативами, а также вводный инструктаж и инструктаж на рабочем месте | | | | | 2 | ОК 08. |
| 2 | Постановка задачи | | | | | 8 | ОК 02, ОК 06, ОК 07, ОК 09. |
| 3 | Настройка рабочего окружения | | | | | 8 | ОК 01, ОК 09. |
| 4 | Анализ предметной области, постановка расширенного технического задания | | | | | 10 | ПК 5.1, ОК 03, ОК 04. |
| 5 | Проектирование | | | | | 35 | ПК 5.2, ПК 5.3, |
| 6 | Реализация программного кода | | | | | 35 | ПК 5.4, ПК 5.8. |
| 7 | Тестирование и отладка полученного кода | | | | | 22 | ПК 5.5, ОК 02, ОК 09. |
| 8 | Подготовка отчетной документации, включая детальное описание разработанных и использованных алгоритмов | | | | | 10 | ПК 5.6, ПК 5.7, ОК 09. |
| 9 | Подготовка к сдаче и сдача промежуточной аттестации | | | | | 2 | ОК 04, ОК 05. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Индивидуальное задание на практику разработано в соответствии с рабочей программой практики. | | | | | |
| Руководитель  практики от университета |  |  |  |  |  |
|  | (дата) |  | (подпись) |  | (Ф.И.О.) |
| С индивидуальным заданием ознакомлен(а) | |  | | | |
|  | | (дата, подпись обучающегося) | | | |

**ОТЗЫВ РУКОВОДИТЕЛЯ ПРАКТИКИ**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ф.И.О. обучающегося | Кривошеин Дмитрий Васильевич | | | |
| Специальность | 09.02.07 Информационные системы и программирование | | | |
| Учебная группа | ИСПк-204-52-00 | | | |
| Вид практики | учебная практика | | | |
| Сроки прохождения практики | с | 15.01.2024 | по | 19.06.2024 |
| Место прохождения практики | ФГБОУ ВО «Вятский государственный университет», Колледж ВятГУ | | | |
|  | (наименование организации, структурного подразделения организации) | | | |

ВИДЫ И КАЧЕСТВО ВЫПОЛНЕНИЯ РАБОТ

|  |  |  |  |
| --- | --- | --- | --- |
| Вид работ | Критерий выполнения работ | | |
| Выполнены полностью самостоятельно | Выполнены с незначительной помощью наставника | Выполнены с помощью наставника |
| Постановка задачи | V |  |  |
| Настройка рабочего окружения | V |  |  |
| Анализ предметной области, постановка расширенного технического задания | V |  |  |
| Проектирование | V |  |  |
| Реализация программного кода | V |  |  |
| Тестирование и отладка полученного кода | V |  |  |
| Подготовка отчетной документации, включая детальное описание разработанных и использованных алгоритмов | V |  |  |
| Подготовка к сдаче и сдача промежуточной аттестации | V |  |  |

Обучающийся ознакомлен с правилами внутреннего трудового распорядка, охраны труда, техники безопасности, противопожарной безопасности, санитарно-эпидемиологическими правилами и гигиеническими нормативами, а также прошел вводный инструктаж и инструктаж на рабочем месте.

Во время прохождения учебной практики обучающимся освоены следующие профессиональные и общие компетенции:

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование компетенции | Показатели оценки | Оценка | |
| Освоена | Не освоена |
| ПК 5.1. Собирать исходные данные для разработки проектной документации на информационную систему. | Способен выполнять анализ предметной области, выявляя существенные элементы, оказывающие влияние на проектируемую систему | V |  |
| ПК 5.2. Разрабатывать проектную документацию на разработку информационной системы в соответствии с требованиями заказчика. | Способен выполнять декомпозицию сущностей с целью получения наиболее полной картины о целесообразной структуре разработки | V |  |
| ПК 5.3. Разрабатывать подсистемы безопасной информационной системы в соответствии с техническим заданием. | Способен прогнозировать потенциально некорректные действия пользователя и предусматривать соответствующие реакции со стороны системы | V |  |
| ПК 5.4. Производить разработку модулей информационной системы в соответствии с техническим заданием. | Способен создавать программный код, отвечающий предъявляемым требованиям | V |  |
| ПК 5.5. Осуществлять тестирование информационной системы на этапе опытной эксплуатации с фиксацией выявленных ошибок кодирования в разрабатываемых модулях информационной системы. | Способен выполнять оценку корректности функционирования системы | V |  |
| ПК 5.6. Разрабатывать техническую документацию на эксплуатацию информационной системы. | Способен разрабатывать проектную, техническую и пользовательскую документации | V |  |
| ПК 5.7. Производить оценку информационной системы для выявления возможности ее модернизации. | Способен оценивать перспективы дальнейшего развития программной системы | V |  |
| ПК 5.8. Осуществлять интеграцию информационной системы с другими программными продуктами. | Способен повторно использовать готовые шаблонные решения при разработке программного продукта | V |  |
| ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам. | Способен анализировать специализированную информацию и находить оптимальные пути решения задач профессиональной деятельности | V |  |
| ОК 02. Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности. | Способен анализировать специализированную информацию и находить оптимальные пути решения задач профессиональной деятельности | V |  |
| ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по финансовой грамотности в различных жизненных ситуациях. | Способен находить пути улучшения имеющихся решений, позволяющих повысить их общий качественный уровень | V |  |
| ОК 04. Эффективно взаимодействовать и работать в коллективе и команде. | Способен конструктивно обмениваться информацией с коллегами, грамотно формулировать запросы в целях получения разъясняющей информации | V |  |
| ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста. | Способен конструктивно обмениваться информацией с коллегами, грамотно формулировать запросы в целях получения разъясняющей информации | V |  |
| ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения. | Способен конструктивно обмениваться информацией с коллегами, грамотно формулировать запросы в целях получения разъясняющей информации | V |  |
| ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях. | Способен эффективно использовать компьютерное время, а также материальные ресурсы, необходимые для решения поставленных задач | V |  |
| ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности. | Способен соблюдать требования внутреннего трудового распорядка организации, охраны труда и техники безопасности в целях сохранения собственного здоровья | V |  |
| ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках. | Способен использовать в своей работе специализированную документацию | V |  |

**Краткая характеристика работы обучающегося**

|  |
| --- |
| Программа практики выполнена в полном объеме. Все виды работ выполнялись в срок, |
| без существенных замечаний. В достаточной степени была проявлена самостоятельность |
| и умение грамотно пользоваться источниками информации, находящимися в свободном |
| доступе. |

|  |  |  |
| --- | --- | --- |
|  |  | Руководитель практики  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Подпись ФИО  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (должность)  Дата «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 202\_\_ год |

**СОДЕРЖАНИЕ**

[**ВВЕДЕНИЕ** 3](#_Toc170210883)

[**1. СВЕДЕНИЯ О РАБОТЕ, ВЫПОЛНЕННОЙ В ПЕРИОД ПРОХОЖДЕНИЯ УЧЕБНОЙ ПРАКТИКИ** 4](#_Toc170210884)

[**2. НАСТРОЙКА РАБОЧЕГО ОКРУЖЕНИЯ** 5](#_Toc170210885)

[**3. ФОРМУЛИРОВКА ИНДИВИДУАЛЬНОГО ЗАДАНИЯ** 6](#_Toc170210886)

[**4. ОПИСАНИЕ ВЫПОЛНЕНИЯ ИНДИВИДУАЛЬНОГО ЗАДАНИЯ** 7](#_Toc170210887)

[**4.1 Анализ предметной области и обзор аналогов** 7](#_Toc170210888)

[**4.2 Разработка структуры приложения и алгоритмов функционирования** 9](#_Toc170210889)

[**4.3 Реализация программы** 12](#_Toc170210890)

[**4.4 Тестирование программного обеспечения** 22](#_Toc170210891)

[**ЗАКЛЮЧЕНИЕ** 24](#_Toc170210892)

[**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ** 25](#_Toc170210893)

[**ПРИЛОЖЕНИЕ А** 26](#_Toc170210894)

[**ПРИЛОЖЕНИЕ Б** 36](#_Toc170210895)

[**ПРИЛОЖЕНИЕ В** 37](#_Toc170210896)

[**ПРИЛОЖЕНИЕ Г** 38](#_Toc170210897)

# **ВВЕДЕНИЕ**

Учебная практика ПМ.05 проходила на базе Колледжа ФГБОУ ВО «Вятский государственный университет» в период с 15.01.2024 г. по 19.06.2024 г. по четвергам.

Цель практики: сформировать у обучающихся представление о процессе разработки прикладного программного обеспечения.

Задачи практики:

– дать представление о различных методах проектирования прикладного программного обеспечения;

– сформировать навыки написания документации, требуемой при разработке прикладного программного обеспечения;

– дать представление о различных инструментах, используемых при разработке прикладного программного обеспечения.

# **1. СВЕДЕНИЯ О РАБОТЕ, ВЫПОЛНЕННОЙ В ПЕРИОД ПРОХОЖДЕНИЯ УЧЕБНОЙ ПРАКТИКИ**

В период с 15.01.2024 по 19.06.2024 при прохождении учебной практики ПМ.05 на базе Колледжа ФГОБУ ВО «Вятский государственный университет» был выполнен следующий перечень работ, представленный в таблице 1.

Таблица 1 – Сведения о работе, выполненной в период практики

| Дата | Краткое содержание выполненных работ |
| --- | --- |
| 15.01.2024 | Ознакомление с правилами внутреннего трудового распорядка, охраны труда, техники безопасности, противопожарной безопасности, санитарно-эпидемиологическими правилами и гигиеническими нормативами, а также вводный инструктаж и инструктаж на рабочем месте |
| 16.01.2024-29.01.2024 | Постановка задачи |
| 30.01.2024-12.02.2024 | Настройка рабочего окружения |
| 13.02.2024-26.02.2024 | Анализ предметной области, постановка расширенного технического задания |
| 27.02.2024-01.04.2024 | Проектирование |
| 02.04.2024-06.05.2024 | Реализация программного кода |
| 07.05.2024-27.05.2024 | Тестирование и отладка полученного кода |
| 28.05.2024-13.06.2024 | Подготовка отчетной документации, включая детальное описание разработанных и использованных алгоритмов |
| 17.06.2024-19.06.2024 | Подготовка к сдаче и сдача промежуточной аттестации |

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

(дата) (подпись)

# **2. НАСТРОЙКА РАБОЧЕГО ОКРУЖЕНИЯ**

Python — это язык программирования, который широко используется в интернет-приложениях, разработке программного обеспечения, науке о данных и машинном обучении. Пайтон быстро набрал популярность за счет простоты и понятного синтаксиса. Язык особо полюбился начинающим программистам, но и опытные разработчики нашли в нем немало плюсов. Изначально Пайтон строился с упором на читаемость кода, что сделало его понятным и поддерживаемым даже в масштабных проектах.  Синтаксис ядра языка минималистичен, за счёт чего на практике редко возникает необходимость обращаться к документации. IDE которая была использована при создании проекта является PyCharm. PyCharm – это кроссплатформенная интегрированная среда разработки для языка программирования Python. Представляет собой комплекс средств для написания кода и визуальный отладчик.

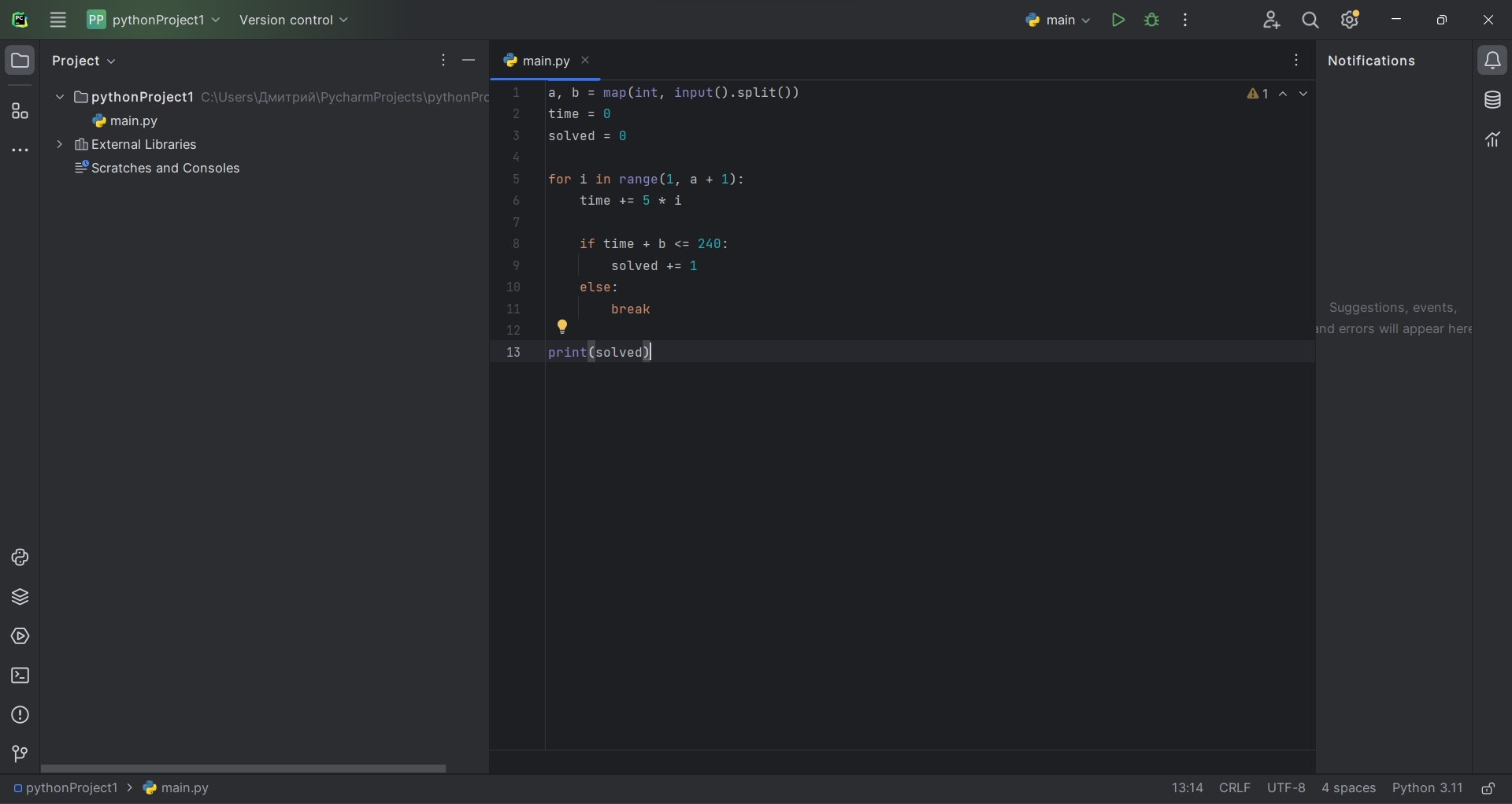


Рисунок 1 – PyCharm.

# **3. ФОРМУЛИРОВКА ИНДИВИДУАЛЬНОГО ЗАДАНИЯ**

В соответствии с индивидуальным заданием на учебную практику необходимо:

Создание Интерактивного учебника по английскому языку. Пользователь может ознакомится с выбранной темой по определенному учебнику, так же каждая тема содержит тестирование для практического применения полученной информации. Тест состоит из двух блоков, в первом необходимо ввести правильный ответ в соответствующее поле, во второй части нам представлен вопрос с четырьмя кнопками выбора варианта ответа. По завершению теста мы может ознакомится с количеством набранных баллов, а также с процентом правильных ответов.

# **4. ОПИСАНИЕ ВЫПОЛНЕНИЯ ИНДИВИДУАЛЬНОГО ЗАДАНИЯ**

## **4.1 Анализ предметной области и обзор аналогов**

Интерактивный учебник представляет собой учебный материал по английскому языку, который содержит различные темы по данной дисциплине, а также тестирование по выбранной теме. Программа может использоваться в учебных заведениях и при обучении на дому. Учебник может быть полезен преподавателям, ведущим уроки английского для учеников различных возрастов. Так же для самих учеников для подготовки к сдаче экзамена.

Существуют следующие аналоги приложения, это сайты: begin‑english.ru [1], www.englishgu.ru/online-uchebnik-po-angliyskomu/ и https://learnenglish.britishcouncil.org. Внешний вид сайтов приведен на рисунках 2, 3 и 4.

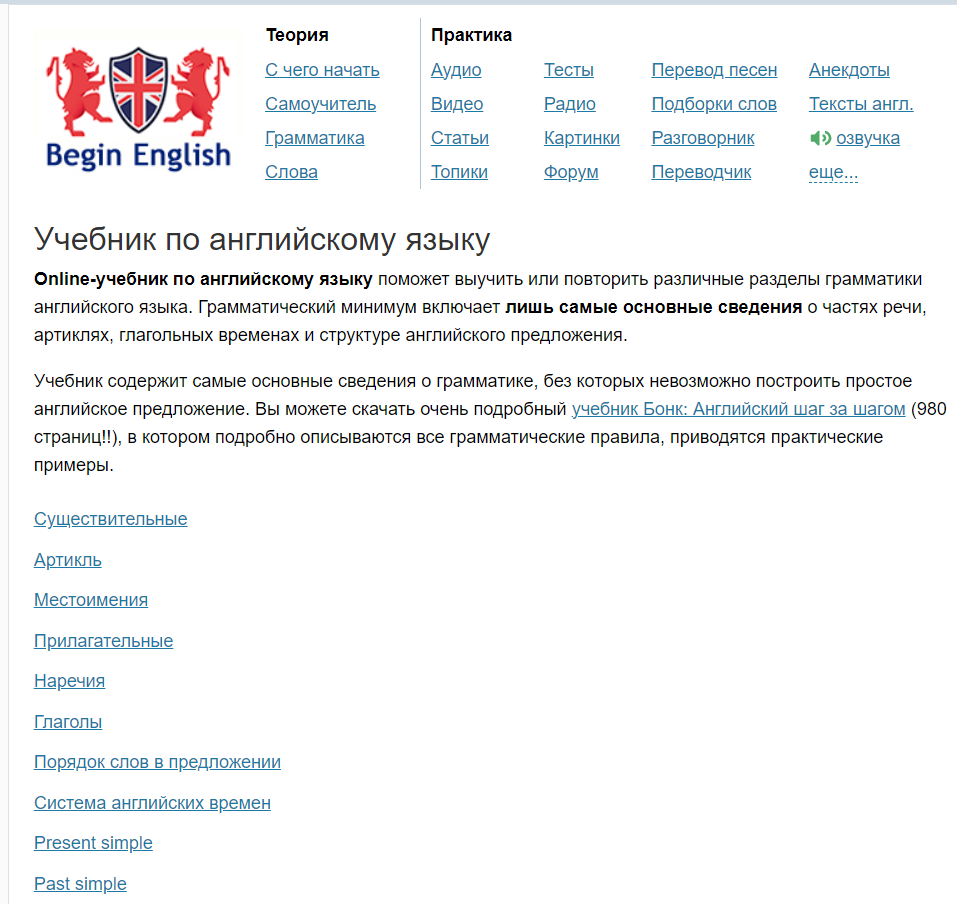


Рисунок 2 – Сайт http://begin-english.ru/study/.

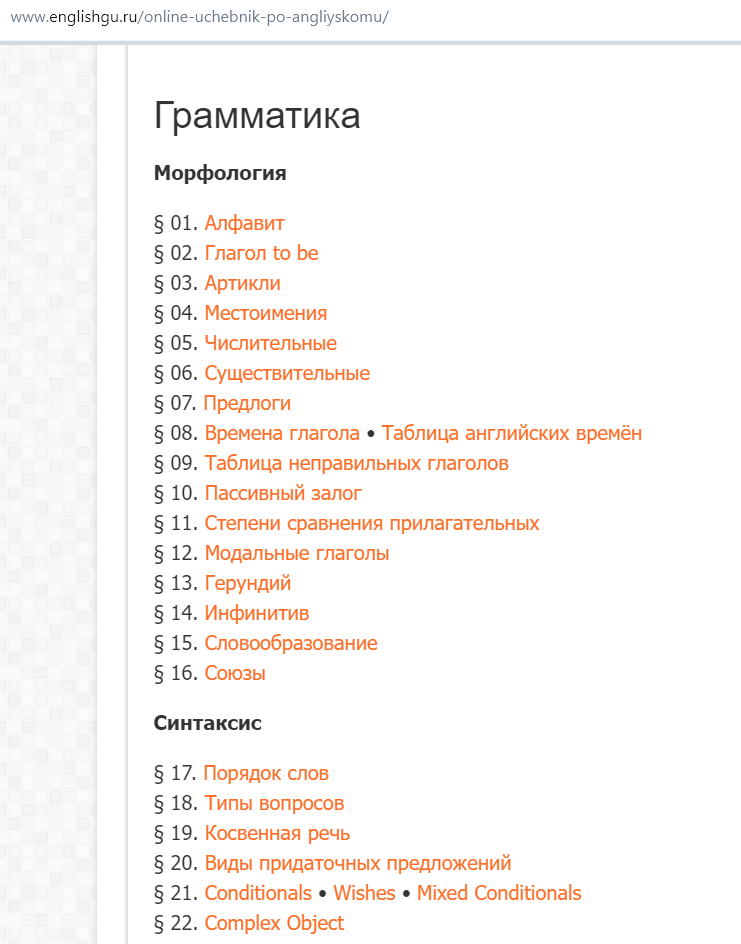


Рисунок 3 – Сайт <www.englishgu.ru/online-uchebnik-po-angliyskomu/>.

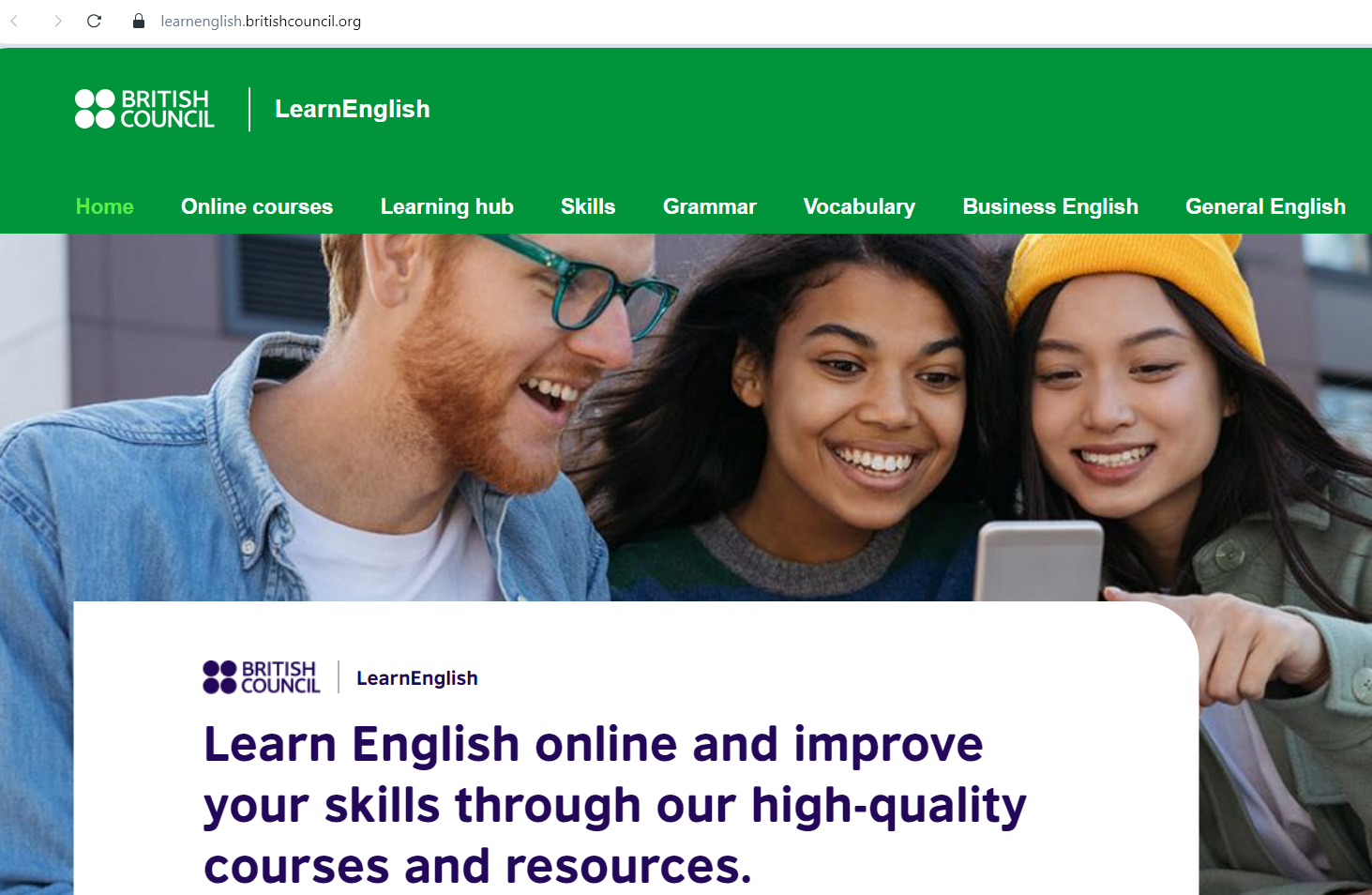


Рисунок 4 – Сайт <https://learnenglish.britishcouncil.org>.

Функциональными возможностями сайта <http://begin-english.ru/study/> является: просмотре тем с основными темами о грамматике, наличие различных видов практики по типу аудио, видео, статей. Наличие простых тестов для определения уровня знаний пользователя.

Функциональными возможностями сайта <https://www.englishgu.ru/online-uchebnik-po-angliyskomu/> является: наличие различных тем для ознакомления, разные виды тестирований и наличие видео уроков.

Функциональными возможностями сайта <https://learnenglish.britishcouncil.org> является: просмотр онлайн курсов, выбор тем с уровнем сложности в зависимости от знаний языка, тестирования чтения, письма, прослушивания и речи.

Их общими преимуществами являются доступность на любых устройствах, а также большой выбор различных тем. Общими недостатками являются невозможность прочтения учебника без наличия интернета, скорость работы учебников зависит от скорости интернета.

В результате анализа аналогов стала понятна необходимость разработки приложения не требующего выхода в интернет, не требующего регистрации и независящего от скорости интернета.

Функциональными возможностями нашего приложения является: просмотр содержимого по выбранной теме, наличие тестирования по теме, выбранной в главном меню.

## **4.2 Разработка структуры приложения и алгоритмов функционирования**

Разработанное мной приложение содержит следующие основные функциональные компоненты:

1. Компоненты загрузки и отображения темы учебника – при нажатии на кнопку выбранной темы в главном меню отображается необходима тема.
2. Компонент загрузки вопросов и вариантов ответов для первого блока тестирования – считываются вопросы и правильный ответ из текстового файла, в который данные записываются в следующем формате «текст вопроса; ответ на вопрос».
3. Компонент загрузки вопросов и вариантов ответов для второго блока тестирования – считываются вопросы и варианты ответов из текстового файла, в который данные записываются в следующем формате «текст вопроса; 1 ответ; 2 ответ; 3 ответ; 4 ответ».
4. Компонент проверки правильности ответа – после выбора пользователем варианта ответа производится проверка. Если пользователь ответил правильно, то счетчик правильных ответов обновляется и прибавляется один бал, если же ответ был неверным, то бал не засчитывается.

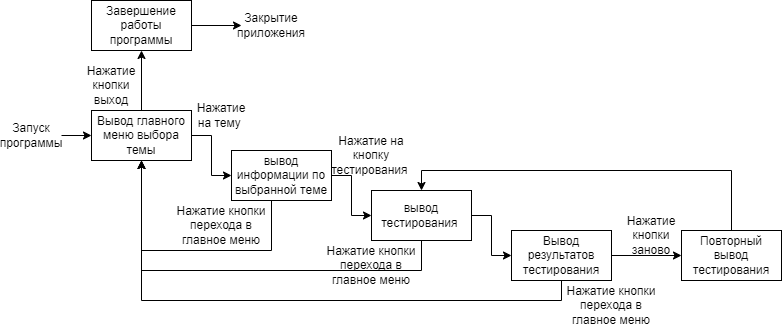


Рисунок 5 – Схема взаимодействия компонентов.

Описание алгоритмов:

1. Компонент загрузки вопросов и вариантов ответа из файла создан для того, чтобы можно было хранить вопросы и варианты ответов к ним не в самом коде, а в другом файле. Для начала создаем список questions который будет хранить все вопросы в нем. После этого подгружаем текстовый файл, который и считывает каждую строку. Вопрос и ответ хранятся в раздельных строках в формате «Тема Вопрос 1 ответ 2 ответ 3 ответ 4 ответ правильный ответ». Записываем в переменные тему, вопросы и ответы.

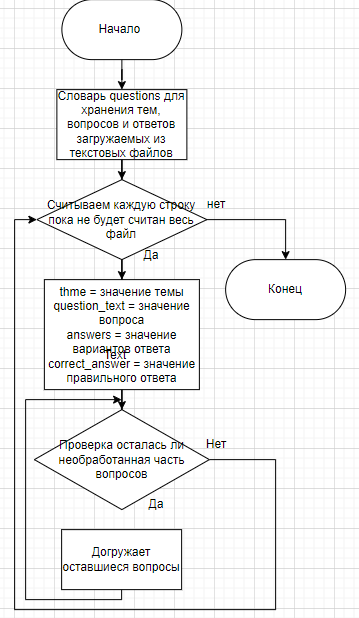


Рисунок 6 – Схема алгоритма компонента 1.

1. Компонент вывода вопросов и вариантов ответа создан для того, чтобы на кнопках и нужных полях выводился необходимый текст по определенному вопрос. Создается список всех вариантов ответ Answer. Происходит проверка наличия вопросов, если они заканчиваются, то тестирование завершается. Извлекаем элементы из словаря questions, затем сохраняем этот элемент в переменную questions, затем сохраняем этот элемент в переменную question\_text для использования. Также присутствует 4 варианта ответа для вопроса, если пользователь выбирает ответ, мы переходим к следующему вопросу. После этого мы выводим весь текст на необходимы нам элементы. То есть поля и кнопки для вопросов.

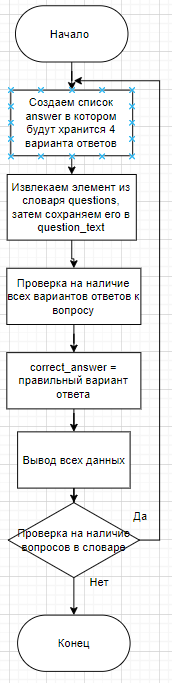


Рисунок 7 – Схема алгоритма компонента 2.

## **4.3 Реализация программы**

**4.3.1. Программная реализация**

Для реализации программы был выбран язык программирования Python

Для реализации программы был выбран язык программирования Python в среде разработки PyCharm.

Пользовательский интерфейс был разработан с использованием библиотеки Tkinter для графического отображения данных. Для работы с pdf файлами была использована библиотека Fitz, библиотека PIL для работы с изображениями, для работы с файлами и нахождения их расположения – библиотека OS, а для компоновки кода в исполняемый файл PyInstaller.

Данные, необходимые для работы программы, хранятся в текстовых файлах (расширение .txt и .pdf), что обеспечивает удобство и простоту доступа к ним.

Класс «MainManu» – главное меню учебника. Здесь используются простые функции: «main\_menu» с помощью которого производится отрисовка основного окна и всех кнопок, функция «start\_theme» отвечает за переход к другому классу и открытию окна с темами, «start\_dictionary» выполняет переход к другому классу, а также открытие окна со словарем.

Класс «Dictionary» – новое окно со словарём, где имеются несколько функций: «render\_page» – которая выполняет функцию создания и отрисовки файла (формата .pdf) c имеющимся словарем, функции «next\_page» и «prev\_page» отвечающие за перемещение пользователя по страницам файла и функция «glav\_menu» которая позволяет переместиться обратно в главное меню.

Класс «TeachBook» – третье окно с учебником. В этом классе находятся функции, которые уже использовались в предыдущем классе, а именно «render\_page», «glav\_menu», «next\_page» и «prev\_page» которые выполняют те же функции что и в прошлом классе. Так же добавились новая функция «start\_test» – позволяет перейти к тестированию по выбранной теме.

Класс «Tester» – отвечает за создание одного из тестирований. Этот класс включает в себя следующие функции, а именно «load\_file» – отвечает за загрузку файлов с вопросами по текущей теме, «load\_question» – загружает вопросы из указанного файла, «setup\_ui»  –настраивает графическй интерфейс пользователя, «show\_question» – отображает текущий вопрос и варианты ответов, «show\_results» – отображают результаты теста, «check\_answer» – проверяет правильность ответа и обновляет счет и последняя функция в данном классе «prev\_question» – позволяет возвращаться к предыдущему вопросу

Класс «TestApp» – содержит некоторые функции, которые использовались в предыдущем классе, но имеют некоторые отличия в структуре. Функция «load\_questions\_from\_file» – загружает вопросы и ответы из указанного файла, «display\_questions» – отображает вопросы текущего блока, «back\_to\_theme» – возвращает пользователя к выбранной теме, «next\_block» – отвечает за переход к следующему блоку вопросов, «prev\_block» – позволяет пользователя вернуться к предыдущему блоку вопросов, «save\_current\_answers» – сохраняет ответы пользователя для текущего блока и наконец «calculate\_block\_score» – вычисляет баллы для указанного блока вопросов.

Описание переменных класса «MainMenu»

theme\_Button – переменная для хранения кнопки темы.

canvas – канвас (область рисования) для основного окна приложения.

Описание переменных класса «Dictionary»

pdf\_file – путь к PDF-файлу словаря.

page\_num – номер текущей страницы, отображаемой в словаре, начальное значение - 0.

img\_tk – переменная для хранения изображения текущей страницы.

canvas – канвас для отображения содержимого словаря.

button\_glav – кнопка для возврата в главное меню.

Описание переменных класса «TeachBook»

theme – тема учебника.

pdf\_file – путь к PDF-файлу учебника.

page\_num – номер текущей страницы, отображаемой в учебнике, начальное значение - 0.

img\_tk – переменная для хранения изображения текущей страницы.

root – корневой элемент интерфейса (основное окно).

canvas – канвас для отображения содержимого учебника.

button – кнопка для запуска тестирования.

button\_glav – кнопка для возврата в главное меню.

Описание переменных класса «Tester»

root – корневой элемент интерфейса (основное окно).

theme – тема теста.

questions – список вопросов для теста.

current\_question – индекс текущего вопроса, начальное значение - 0.

score – счет правильных ответов, начальное значение - 0.

false – счет неправильных ответов, начальное значение - 0.

first\_test\_score – начальный счет теста (если есть), передаваемое значение.

first\_test\_questions – общее количество вопросов в начальном тесте (если есть), передаваемое значение.

history – история ответов пользователя.

file – файл с вопросами для теста.

canvas – канвас для отображения текущего вопроса.

buttons\_frame – фрейм для размещения кнопок с ответами.

assignment\_label – метка для отображения темы вопроса.

question\_label – метка для отображения текста вопроса.

back\_button – кнопка для возврата к предыдущему вопросу.

buttons – список кнопок для выбора ответа.

Описание переменных класса «TestsApp»

root – корневой элемент интерфейса (основное окно).

theme – тема теста.

current\_block – индекс текущего блока вопросов, начальное значение - 0.

questions\_blocks – список блоков вопросов.

answers\_blocks – список блоков правильных ответов.

user\_answers – список ответов пользователя.

scores – список счетов для каждого блока.

block\_titles – заголовки блоков вопросов.

total\_score – общий счет пользователя по всем блокам, начальное значение - 0.

total\_questions – общее количество вопросов во всех блоках.

canvas – канвас для отображения текущего блока вопросов.

frame – фрейм для размещения вопросов и ответов текущего блока.

next\_button – кнопка для перехода к следующему блоку вопросов.

prev\_button – кнопка для возврата к предыдущему блоку вопросов.

question\_labels – список меток для отображения текста вопросов текущего блока.

answer\_entries – список полей ввода для ответов пользователя на текущий блок вопросов.

**4.3.2. Реализация интерфейса**

Экранная форма главного меню представлена на рисунке 8 и предоставляет выбор из 3 кнопок: «Unit’s», «Dictionary», «Exit». При нажатии на кнопку: «Unit» открывается экранная форма отображения темы; «Dictionary» выводит словарь; «Exit» по нажатию этой кнопки осуществляется выход из приложения. Также наверху экрана находится надпись «Topics for review».

****

Рисунок 8 – Экранная форма главного меню.

Экранная форма отображения темы представлена на рисунке 9 и предоставляет поле вывода pdf файла с темами, а также 2 кнопки: «Main menu» и «Testing». При нажатии на кнопку: «Main menu» осуществляется переход к окну с выбором темы; «Testing» запускает тестирование.

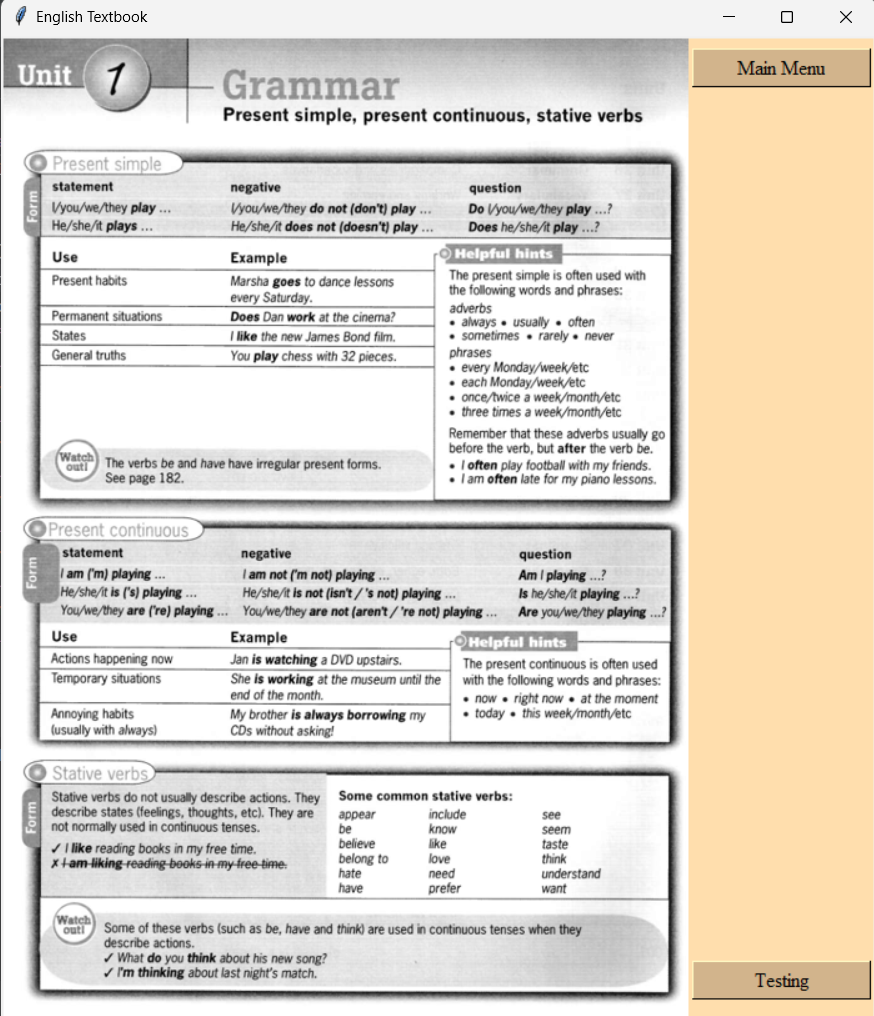


Рисунок 9 – Экранная форма окна вывода тем.

Экранная форма первой части тестирования представлена на рисунке 10 и предоставляет поле вывода задания, поля ввода ответов, а также 2 кнопки: «The nest block» «Back». При нажатии на кнопку: «The next block» осуществляется переход к следующему блоку вопрос; «Back» позволяет пользователю вернуться к предыдущему вопросу.

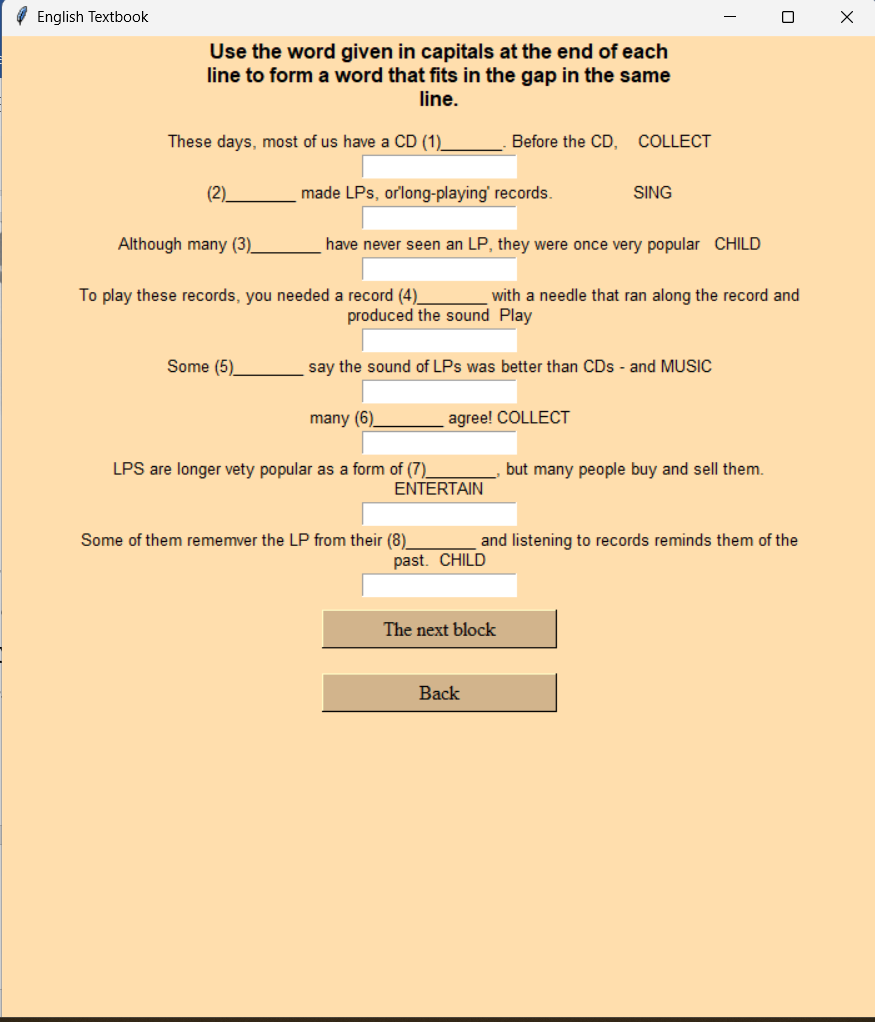


Рисунок 10 – Экранная форма тестирования.

Экранная форма второй части тестирования представлена на рисунке 11 и предоставляет поле вывода задания, четыре кнопки для выбора варианта ответа и кнопка: «Back». При нажатии на кнопку выбора варианта ответа осуществляется переход к следующему вопросу; «Back» позволяет пользователю вернуться к предыдущему вопросу.

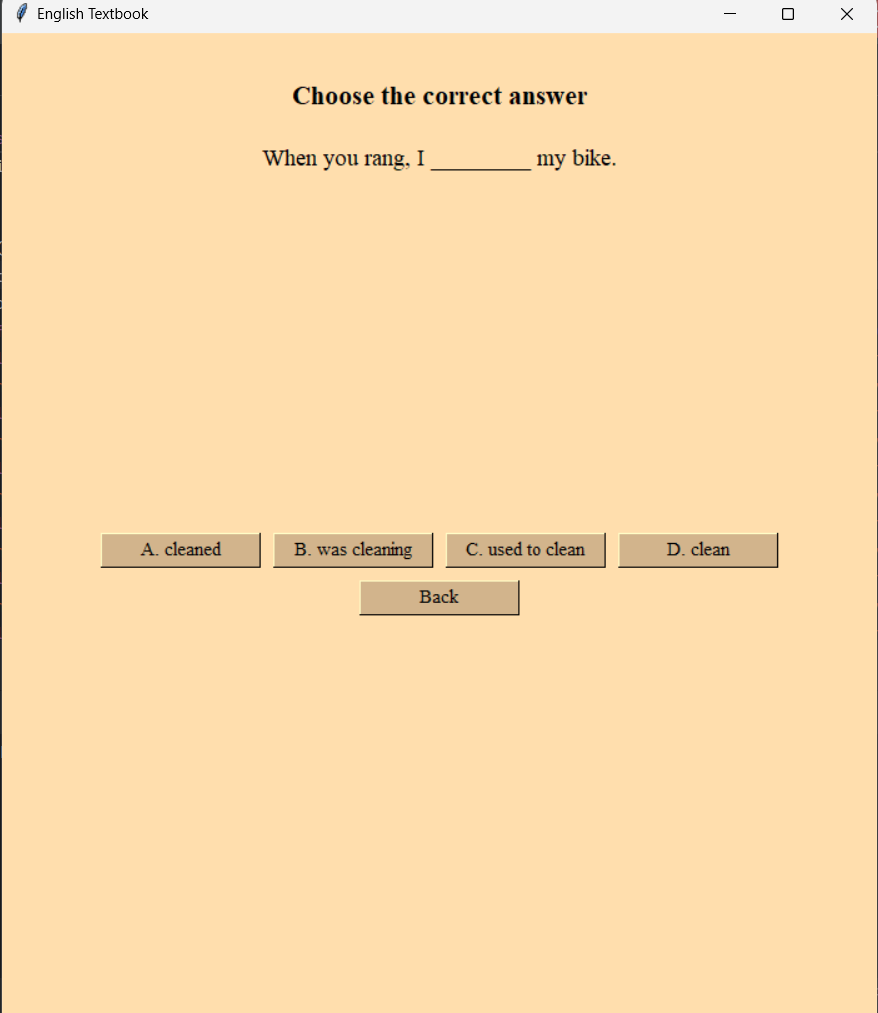


Рисунок 11 – Экранная форма тестирования.

Экранная форма вывода результатов тестирования представлена на рисунке 12 и предоставляет поле вывода результатов и две кнопки: «Go to the main menu» и «Start the test again». При нажатии на кнопку «Go to the main menu» осуществляется переход к окну с выбором тем; «Start the test again» позволяет пользователю пройти тестирование повторно.

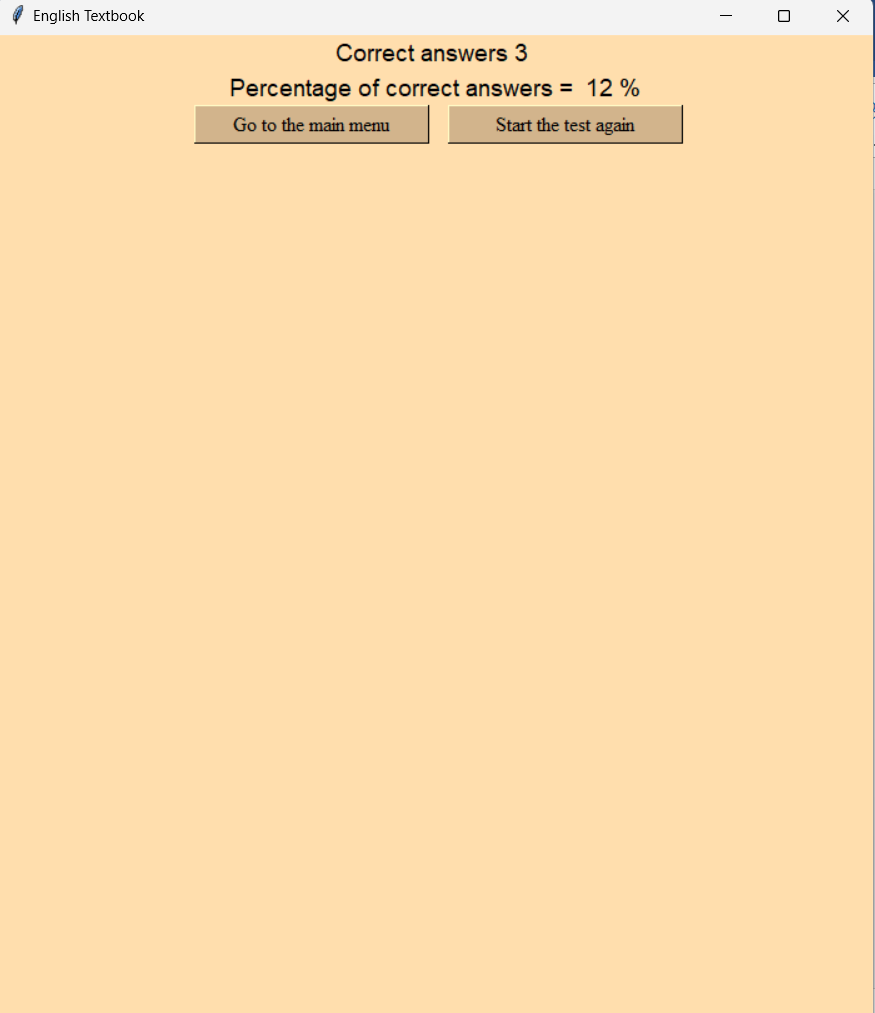


Рисунок 12 – Экранная форма вывода результатов тестирования.

## **4.4 Тестирование программного обеспечения**

Таблица 1 – Критерии приёмки

|  |  |  |  |
| --- | --- | --- | --- |
| № п.п. | Действия пользователя | Критерии приемки | Отметка о прохождении |
| 1 | Проверка наличия элементов взаимодействия: кнопка открытия тем, кнопка открытия словаря, кнопка выхода. Поля отображения тем, а также поля для ввода и вывода информации. | В интерфейсе присутствуют все элементы включая кнопки, поля отображения информации или файлов и поля для ввода. | Успешно |
| 2 | По левой стороне окна кнопки запуска тем:   1. Нажать на кнопку «Unit». | Система отобразит весь текст и изображения корректно и не выходят за рамки. | Успешно |
| 3 | Нажатие кнопки «Exit» | После нажатия на кнопку программа завершается и происходит её закрытие | Успешно |
| 4 | В окне с темами и словарем:   1. Нажатие стрелочки вправо и влево на клавиатуре. | Система отрисует следующую или предыдущую страницу темы или словаря. | Успешно |
| 5 | В правом верхнем углу окна перехода в главное меню:  Нажатие на кнопку «Main Menu». | Система вернет пользователя в главное меню с выбором темы. | Успешно |

Продолжение таблицы 1 – Критерии приёмки

|  |  |  |  |
| --- | --- | --- | --- |
| 6 | В правом нижнем углу окна переход в тестирование:   1. Нажать на кнопку «Testing». | Система отобразит вопросы тестирования с полями для ввода ответа. | Успешно |
| 7 | 1. Нажать кнопку «Back». | Система вернет пользователя к предыдущему вопросу, а в случае нахождения на первой странице тестирования, вернет пользователя к выбранной ранее теме. | Успешно |
| 8 | Закончить тестирование. | После окончания тестирования система подсчитает количество правильных ответов и отобразит их в баллах, а также в процентах. | Успешно |
| 9 | Ниже поля вывода результатов тестирования:   1. Нажать кнопку «Start the test again». | Система отчистит предыдущий результат по тестированию и запустит этот же тест заново. | Успешно |
| 10 | Ниже поля вывода результатов тестирования:   1. Нажать кнопку «Go to the main menu». | Система вернет пользователя в меню с выбором темы. | Успешно |

# **ЗАКЛЮЧЕНИЕ**

В ходе выполнения учебной практики, в соответствии с индивидуальным заданием, была разработан интерактивный учебник по английскому языку и закреплены навыки использования языка программирования Python и библиотеки Tkinter. Были проведены анализ предметной области, разработка программного кода, тестирование, а также написана отчетная документация.

Таким образом, программа учебной практики ПМ.05 в период с 15.01.2024 по 19.06.2024 была выполнена мной в полном объеме.

При выполнении работ, предусмотренных заданием на практику, в организации ФГБОУ ВО «Вятский государственный университет» мной, Кривошеиным Дмитрием Васильевичем, соблюдались правила внутреннего трудового распорядка, охраны труда, техники безопасности, противопожарной безопасности, санитарно-эпидемиологические правила и гигиенические нормативы. Пройден вводный инструктаж и инструктаж на рабочем месте. Замечаний не имел.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Аналог интерактивного учебника по английскому языку – URL: http://begin-english.ru/study/
2. Аналог интерактивного учебника по английскому языку – URL: <https://learnenglish.britishcouncil.org>
3. Основные команды Python – полное описание с примерами – URL: <https://eternalhost.net/blog/razrabotka/komandy-python>
4. Язык программирования Python. Основы синтаксиса [Электронный ресурс]: Википедия. Свободная энциклопедия. – URL: <https://www.python-lang.org/ru/> (Дата обращения: 06.04.2024)
5. Язык программирования Python. Циклы и переменные [Электронный ресурс]: Википедия. Свободная энциклопедия. URL: <https://docs.python.org/3/tutorial/controlflow.html> (Дата обращения: 01.05.2024)
6. Руководство по tkinter [Электронный ресурс] – URL: <https://metanit.com/python/tkinter/>

# **ПРИЛОЖЕНИЕ А**

**ИСХОДНЫЙ КОД**

from tkinter import Tk, Label, Button, messagebox  
import tkinter as tk  
import fitz  
from PIL import Image, ImageTk  
from os import path  
  
window = Tk()  
abs\_path\_files = f'{str(path.dirname(path.abspath(\_\_file\_\_)))}{path.sep}files{path.sep}'  
  
  
class MainMenu:  
 def \_\_init\_\_(self):  
 self.theme\_Button = None  
 self.canvas = tk.Canvas(window, bg='#FFDEAD')  
 self.canvas.pack(fill=tk.BOTH, expand=True)  
 self.main\_menu()  
 window.mainloop()  
  
 def main\_menu(self):  
 window.title("English textbook")  
 window.geometry("700x1000+500+0")  
 Label(self.canvas, text="Topics for review", font=("Arial Bold", 20), background='#FFDEAD').place(x=20, y=10)  
 Button(self.canvas, text="Unit 1", command=lambda: self.start\_theme("1"), bd=1, bg='#D2B48C', width=20,  
 font=("Times New Roman", 12), overrelief="ridge", activebackground="#F5DEB3").place(x=20, y=60)  
 Button(self.canvas, text="Unit 2", command=lambda: self.start\_theme("2"), bd=1, bg='#D2B48C', width=20,  
 font=("Times New Roman", 12), overrelief="ridge", activebackground="#F5DEB3").place(x=20, y=100)  
 Button(self.canvas, text="Unit 3", command=lambda: self.start\_theme("3"), bd=1, bg='#D2B48C', width=20,  
 font=("Times New Roman", 12), overrelief="ridge", activebackground="#F5DEB3").place(x=20, y=140)  
 Button(self.canvas, text="Unit 4", command=lambda: self.start\_theme("4"), bd=1, bg='#D2B48C', width=20,  
 font=("Times New Roman", 12), overrelief="ridge", activebackground="#F5DEB3").place(x=20, y=180)  
 Button(self.canvas, text="Unit 5", command=lambda: self.start\_theme("5"), bd=1, bg='#D2B48C', width=20,  
 font=("Times New Roman", 12), overrelief="ridge", activebackground="#F5DEB3").place(x=20, y=220)  
 Button(self.canvas, text="Unit 6", command=lambda: self.start\_theme("6"), bd=1, bg='#D2B48C', width=20,  
 font=("Times New Roman", 12), overrelief="ridge", activebackground="#F5DEB3").place(x=20, y=260)  
 Button(self.canvas, text="Unit 7", command=lambda: self.start\_theme("7"), bd=1, bg='#D2B48C', width=20,  
 font=("Times New Roman", 12), overrelief="ridge", activebackground="#F5DEB3").place(x=20, y=300)  
 Button(self.canvas, text="Unit 8", command=lambda: self.start\_theme("8"), bd=1, bg='#D2B48C', width=20,  
 font=("Times New Roman", 12), overrelief="ridge", activebackground="#F5DEB3").place(x=20, y=340)  
 Button(self.canvas, text="Unit 9", command=lambda: self.start\_theme("9"), bd=1, bg='#D2B48C', width=20,  
 font=("Times New Roman", 12), overrelief="ridge", activebackground="#F5DEB3").place(x=20, y=380)  
 Button(self.canvas, text="Unit 10", command=lambda: self.start\_theme("10"), bd=1, bg='#D2B48C', width=20,  
 font=("Times New Roman", 12), overrelief="ridge", activebackground="#F5DEB3").place(x=20, y=420)  
 Button(self.canvas, text="Unit 11", command=lambda: self.start\_theme("11"), bd=1, bg='#D2B48C', width=20,  
 font=("Times New Roman", 12), overrelief="ridge", activebackground="#F5DEB3").place(x=20, y=460)  
 Button(self.canvas, text="Unit 12", command=lambda: self.start\_theme("12"), bd=1, bg='#D2B48C', width=20,  
 font=("Times New Roman", 12), overrelief="ridge", activebackground="#F5DEB3").place(x=20, y=500)  
 Button(self.canvas, text="Unit 13", command=lambda: self.start\_theme("13"), bd=1, bg='#D2B48C', width=20,  
 font=("Times New Roman", 12), overrelief="ridge", activebackground="#F5DEB3").place(x=20, y=540)  
 Button(self.canvas, text="Unit 14", command=lambda: self.start\_theme("14"), bd=1, bg='#D2B48C', width=20,  
 font=("Times New Roman", 12), overrelief="ridge", activebackground="#F5DEB3").place(x=20, y=580)  
 Button(self.canvas, text="Dictionary", command=self.start\_dictionary, bd=1, bg='#FFE4B5', width=20,  
 font=("Times New Roman", 12), overrelief="ridge", activebackground="#FFE4C4").place(x=20, y=660)  
 Button(self.canvas, text="Exit", command=window.destroy, bd=1, bg='#8B0000', width=20,  
 font=("Times New Roman", 12), overrelief="ridge", activebackground="#B22222").place(x=20, y=700)  
  
 def start\_theme(self, theme):  
 destroy\_canvas(self.canvas)  
 pdf\_file = f'{abs\_path\_files}pdf{path.sep}{theme}.pdf'  
 TeachBook(theme, pdf\_file)  
  
 def start\_dictionary(self):  
 destroy\_canvas(self.canvas)  
 pdf\_file = f'{abs\_path\_files}pdf{path.sep}vocabulary.pdf'  
 Dictionary(pdf\_file)  
  
  
class Dictionary:  
 def \_\_init\_\_(self, pdf\_file):  
 self.pdf\_file = pdf\_file  
 self.page\_num = 0  
 self.img\_tk = tk.Image  
  
 self.canvas = tk.Canvas(window, bg='#FFDEAD')  
 self.canvas.pack(fill=tk.BOTH, expand=True)  
 window.geometry("700x1000+500+0")  
  
 self.render\_page()  
  
 window.bind("<Left>", self.prev\_page)  
 window.bind("<Right>", self.next\_page)  
  
 self.button\_glav = tk.Button(self.canvas, text="Main Menu", command=self.glav\_men, bd=1, bg='#D2B48C',  
 width=11, font=("Times New Roman", 11), overrelief="ridge",  
 activebackground="#F5DEB3")  
 self.button\_glav.place(x=602, y=10)  
  
 def render\_page(self):  
 pdf\_document = fitz.open(self.pdf\_file)  
 page = pdf\_document[self.page\_num]  
 pix = page.get\_pixmap()  
 img = Image.frombytes("RGB", (pix.width, pix.height), pix.samples)  
 img = img.resize((600, 800))  
 img\_tk = ImageTk.PhotoImage(img)  
 self.canvas.create\_image(0, 0, anchor=tk.NW, image=img\_tk)  
 self.img\_tk = img\_tk  
  
 def prev\_page(self, event):  
 if self.page\_num > 0:  
 self.page\_num -= 1  
 self.render\_page()  
  
 def next\_page(self, event):  
 pdf\_document = fitz.open(self.pdf\_file)  
 if self.page\_num < pdf\_document.page\_count - 1:  
 self.page\_num += 1  
 self.render\_page()  
  
 def glav\_men(self):  
 destroy\_canvas(self.canvas)  
 MainMenu()  
  
  
class TeachBook:  
 def \_\_init\_\_(self, theme, pdf\_file):  
 self.theme = theme  
 self.pdf\_file = pdf\_file  
 self.page\_num = 0  
 self.img\_tk = tk.Image  
  
 self.root = window  
 self.canvas = tk.Canvas(self.root, bg='#FFDEAD')  
 self.canvas.pack(fill=tk.BOTH, expand=True)  
  
 self.root.geometry("700x1000+500+0")  
  
 self.render\_page()  
  
 self.root.bind("<Left>", self.prev\_page)  
 self.root.bind("<Right>", self.next\_page)  
  
 self.button = tk.Button(self.canvas, text="Testing", command=self.start\_test, bd=1, bg='#D2B48C', width=15,  
 font=("Times New Roman", 12), overrelief="ridge", activebackground="#F5DEB3")  
 self.button.place(x=553, y=740)  
  
 self.button\_glav = tk.Button(self.canvas, text="Main Menu", command=self.glav\_men, bd=1, bg='#D2B48C',  
 width=15, font=("Times New Roman", 12), overrelief="ridge",  
 activebackground="#F5DEB3")  
 self.button\_glav.place(x=553, y=10)  
  
 def render\_page(self):  
 pdf\_document = fitz.open(self.pdf\_file)  
 page = pdf\_document[self.page\_num]  
 pix = page.get\_pixmap()  
 img = Image.frombytes("RGB", (pix.width, pix.height), pix.samples)  
 img = img.resize((550, 800))  
 img\_tk = ImageTk.PhotoImage(img)  
 self.canvas.create\_image(0, 0, anchor=tk.NW, image=img\_tk)  
 self.img\_tk = img\_tk  
  
 def prev\_page(self, event):  
 if self.page\_num > 0:  
 self.page\_num -= 1  
 self.render\_page()  
  
 def next\_page(self, event):  
 pdf\_document = fitz.open(self.pdf\_file)  
 if self.page\_num < pdf\_document.page\_count - 1:  
 self.page\_num += 1  
 self.render\_page()  
  
 def start\_test(self):  
 test\_file = f'{abs\_path\_files}txt{path.sep}Tester{path.sep}{self.theme}.txt'  
 if not path.exists(test\_file):  
 messagebox.showerror("Внимание!", "Файл с тестированием отсутствует!")  
 return # Stay in TeachBook  
 destroy\_canvas(self.canvas)  
 TestApp(self.theme)  
  
 def glav\_men(self):  
 destroy\_canvas(self.canvas)  
 MainMenu()  
  
  
class Tester:  
 def \_\_init\_\_(self, theme, score=0, all\_questions=0):  
 self.root = window  
 self.theme = theme  
 self.questions = self.load\_questions(f'{abs\_path\_files}txt{path.sep}Tester{path.sep}{self.theme}.txt')  
 self.current\_question = 0  
 self.score = 0  
 self.false = 0  
 self.first\_test\_score = score  
 self.first\_test\_questions = all\_questions  
 self.history = []  
 self.load\_file()  
 self.setup\_ui()  
  
 def load\_file(self):  
 self.file = open(f'{abs\_path\_files}txt{path.sep}Tester{path.sep}{self.theme}.txt', "r")  
  
 def load\_questions(self, filename):  
 questions = []  
 with open(filename, "r") as file:  
 lines = file.readlines()  
 i = 0  
 while i < len(lines):  
 if i + 5 >= len(lines):  
 break  
 theme = lines[i].strip()  
 i += 1  
 question\_text = lines[i].strip()  
 i += 1  
 answers = []  
 for \_ in range(4):  
 if i < len(lines):  
 answers.append(lines[i].strip())  
 i += 1  
 if i < len(lines):  
 correct\_answer = lines[i].strip()  
 i += 1  
 question = {  
 "theme": theme,  
 "question\_text": question\_text,  
 "answers": answers,  
 "correct\_answer": correct\_answer  
 }  
 questions.append(question)  
 else:  
 break  
 return questions  
  
 def setup\_ui(self):  
 self.canvas = tk.Canvas(self.root, width=400, height=400, bg='#FFDEAD', highlightthickness=0)  
 self.canvas.pack(side=tk.TOP)  
  
 self.buttons\_frame = tk.Frame(self.root, bg='#FFDEAD', highlightthickness=0, bd=0)  
 self.buttons\_frame.pack(side=tk.TOP)  
  
 self.assignment\_label = Label(self.canvas, text="", font=("Times New Roman", 16, "bold"),  
 wraplength=380, bg='#FFDEAD', highlightthickness=0, bd=0)  
 self.assignment\_label.pack(fill=tk.BOTH, expand=True, pady=(10, 5))  
  
 self.question\_label = Label(self.canvas, text="", font=("Times New Roman", 14), wraplength=380, bg='#FFDEAD',  
 highlightthickness=0, bd=0)  
 self.question\_label.pack(fill=tk.BOTH, expand=True, pady=(20, 10))  
  
 self.back\_button = Button(self.buttons\_frame, text="Back", command=self.prev\_question, bd=1, bg='#D2B48C',  
 width=15,  
 font=("Times New Roman", 11), overrelief="ridge", activebackground="#F5DEB3")  
 self.back\_button.pack(side=tk.BOTTOM, pady=10)  
  
 self.buttons = []  
 for j in range(4):  
 button = Button(self.buttons\_frame, text=f"Answer {chr(65 + j)}",  
 command=lambda i=j: self.check\_answer(chr(65 + i)), bd=1, bg='#D2B48C', width=15,  
 font=("Times New Roman", 11), overrelief="ridge", activebackground="#F5DEB3")  
 button.pack(side=tk.LEFT, padx=5)  
  
 self.buttons.append(button)  
  
 self.show\_question()  
  
 def show\_question(self):  
 self.canvas.delete("all")  
 if self.current\_question < len(self.questions):  
 question = self.questions[self.current\_question]  
  
 self.assignment\_label.config(text=question["theme"])  
 self.canvas.create\_window(200, 50, window=self.assignment\_label)  
  
 self.question\_label.config(text=question["question\_text"])  
 self.canvas.create\_window(200, 100, window=self.question\_label)  
  
 for i, answer in enumerate(question["answers"]):  
 self.buttons[i].config(text=answer)  
 for i in range(len(question["answers"]), 4):  
 self.buttons[i].config(text="", state=tk.DISABLED)  
 else:  
 self.show\_results()  
 self.buttons\_frame.destroy()  
  
 def show\_results(self):  
 self.allq = self.score + self.false  
  
 result\_label = tk.Label(self.canvas, text='Correct answers '  
 + str(self.score + self.first\_test\_score) + ' ', font='Arial 14',  
 bg='#FFDEAD')  
 result\_label.pack()  
 proc = tk.Label(self.canvas, text='Percentage of correct answers = '  
 + str(int(((self.score + self.first\_test\_score) /  
 (self.allq + self.first\_test\_questions)) \* 100)) + ' %',  
 font='Arial 14', bg='#FFDEAD')  
 proc.pack()  
  
 def glav\_men():  
 destroy\_canvas(self.canvas)  
 self.file.seek(0)  
 MainMenu()  
  
 def restart():  
 destroy\_canvas(self.canvas)  
 self.file.seek(0)  
 TestApp(self.theme)  
  
 Glav\_menu = tk.Button(self.canvas, text='Go to the main menu', command=glav\_men, bd=1, bg='#D2B48C',  
 width=20, font=("Times New Roman", 12), overrelief="ridge", activebackground="#F5DEB3")  
 Glav\_menu.pack(side=tk.LEFT, padx=10)  
  
 restart\_button = tk.Button(self.canvas, text='Start the test again', command=restart, bd=1, bg='#D2B48C',  
 width=20, font=("Times New Roman", 12), overrelief="ridge",  
 activebackground="#F5DEB3")  
 restart\_button.pack(side=tk.LEFT, padx=5)  
  
 def check\_answer(self, answer):  
 question = self.questions[self.current\_question]  
 self.history.append((self.current\_question, self.score, self.false))  
 if answer == question["correct\_answer"]:  
 self.score += 1  
 else:  
 self.false += 1  
 self.current\_question += 1  
 self.show\_question()  
  
 def prev\_question(self):  
 if self.history:  
 self.current\_question, self.score, self.false = self.history.pop()  
 self.show\_question()  
  
  
class TestApp:  
 def \_\_init\_\_(self, theme):  
 self.root = window  
 self.root.configure(bg='#FFDEAD')  
 self.theme = theme  
 self.current\_block = 0  
 self.questions\_blocks = []  
 self.answers\_blocks = []  
 self.user\_answers = []  
 self.scores = []  
 self.block\_titles = []   
 self.load\_questions\_from\_file(f'{abs\_path\_files}txt{path.sep}TestApp{path.sep}{self.theme}.txt')  
 self.init\_ui()  
 self.total\_score = 0  
 self.total\_questions = sum(len(block) for block in self.questions\_blocks)  
 self.display\_questions()  
  
 def init\_ui(self):  
 self.canvas = tk.Canvas(self.root, width=400, height=400, bg='#FFDEAD', highlightthickness=0)  
 self.canvas.pack(side=tk.TOP)  
  
 self.frame = tk.Frame(self.canvas, bg='#FFDEAD', highlightthickness=0, bd=0)  
 self.frame.pack(side=tk.TOP)  
  
 self.next\_button = tk.Button(self.canvas, text="The next block", command=self.next\_block, bd=1, bg='#D2B48C',  
 width=20, font=("Times New Roman", 12), overrelief="ridge",  
 activebackground="#F5DEB3")  
 self.next\_button.pack(side=tk.TOP, pady=10)  
  
 self.prev\_button = tk.Button(self.canvas, text="Back", command=self.prev\_block, bd=1, bg='#D2B48C', width=20,  
 font=("Times New Roman", 12), overrelief="ridge", activebackground="#F5DEB3")  
 self.prev\_button.pack(side=tk.TOP, pady=10)  
  
 def load\_questions\_from\_file(self, filename):  
 with open(filename, "r", encoding="utf-8") as file:  
 lines = file.readlines()  
 current\_questions = []  
 current\_answers = []  
 current\_title = None  
 for line in lines:  
 line = line.strip()  
 if line.startswith("Название:"):  
 current\_title = line[len("Название:"):].strip()  
 elif line.startswith("Вопрос:"):  
 current\_questions.append({"question": line[len("Вопрос:"):].strip(), "options": []})  
 elif line.startswith("Ответ:"):  
 current\_answers.append(line[len("Ответ:"):].strip().lower())  
 elif line == "Next":  
 if current\_questions and current\_answers and current\_title:  
 self.questions\_blocks.append(current\_questions)  
 self.answers\_blocks.append(current\_answers)  
 self.block\_titles.append(current\_title)  
 self.user\_answers.append([""] \* len(current\_questions))  
 self.scores.append(0)  
 current\_questions = []  
 current\_answers = []  
 current\_title = None  
 if current\_questions and current\_answers and current\_title:  
 self.questions\_blocks.append(current\_questions)  
 self.answers\_blocks.append(current\_answers)  
 self.block\_titles.append(current\_title)  
 self.user\_answers.append([""] \* len(current\_questions))  
 self.scores.append(0)  
  
 def display\_questions(self):  
 for widget in self.frame.winfo\_children():  
 widget.destroy()  
  
 self.question\_labels = []  
 self.answer\_entries = []  
  
 block\_title = self.block\_titles[self.current\_block]  
 block\_label = tk.Label(self.frame, text=block\_title, font='Arial 12 bold', bg='#FFDEAD', wraplength=400)  
 block\_label.pack(pady=(0, 10))  
  
 for i, question in enumerate(self.questions\_blocks[self.current\_block]):  
 question\_label = tk.Label(self.frame, text=question["question"], font='Arial 10', bg='#FFDEAD',  
 wraplength=600)  
 question\_label.pack()  
 self.question\_labels.append(question\_label)  
  
 answer\_entry = tk.Entry(self.frame)  
 answer\_entry.pack()  
 answer\_entry.insert(0, self.user\_answers[self.current\_block][i])  
 self.answer\_entries.append(answer\_entry)  
  
 def next\_block(self):  
 self.save\_current\_answers()  
 if self.current\_block < len(self.questions\_blocks) - 1:  
 self.scores[self.current\_block] = self.calculate\_block\_score(self.current\_block)  
 self.total\_score = sum(self.scores)  
 self.current\_block += 1  
 self.display\_questions()  
 else:  
 self.scores[self.current\_block] = self.calculate\_block\_score(self.current\_block)  
 self.total\_score = sum(self.scores)  
 destroy\_canvas(self.canvas)  
 Tester(self.theme, score=self.total\_score, all\_questions=self.total\_questions)  
  
 def prev\_block(self):  
 if self.current\_block > 0:  
 self.save\_current\_answers()  
 self.scores[self.current\_block] = self.calculate\_block\_score(self.current\_block)  
 self.total\_score = sum(self.scores)  
 self.current\_block -= 1  
 self.display\_questions()  
 else:  
 destroy\_canvas(self.canvas)  
 TeachBook(self.theme, f'{abs\_path\_files}pdf{path.sep}{self.theme}.pdf')  
  
 def save\_current\_answers(self):  
 self.user\_answers[self.current\_block] = [entry.get().strip().lower() for entry in self.answer\_entries]  
  
 def calculate\_block\_score(self, block\_index):  
 user\_answers = self.user\_answers[block\_index]  
 correct\_answers = self.answers\_blocks[block\_index]  
 return sum(  
 1 for user\_answer, correct\_answer in zip(user\_answers, correct\_answers) if user\_answer == correct\_answer)  
  
  
def destroy\_canvas(canvas: tk.Canvas) -> None:  
 window.unbind("<Left>")  
 window.unbind("<Right>")  
 canvas.destroy()  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 MainMenu()

# **ПРИЛОЖЕНИЕ Б**

**ТЕХНИЧЕСКОЕ ЗАДАНИЕ**

# **ПРИЛОЖЕНИЕ В**

**ПРОГРАММА И МЕТОДИКА ИСПЫТАНИЙ**

# **ПРИЛОЖЕНИЕ Г**

**РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ**