



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА – Российский технологический университет»
РТУ МИРЭА

Институт информационных технологий (ИИТ)
Кафедра вычислительной техники (ВТ)

ОТЧЁТ ПО УЧЕБНОЙ ПРАКТИКЕ
Ознакомительная практика

приказ Университета о направлении на практику от «13» февраля 2024 г. №1457-С

Отчет представлен к
рассмотрению:

Студент группы ИКБО-44-23

«31» мая 2024 г.


Закиров И.А.
(подпись и расшифровка подписи)

Отчет утвержден.
Допущен к защите:

Руководитель практики
от кафедры

«31» мая 2024 г.


Холмогоров В.В.
(подпись и расшифровка подписи)

Руководитель практики
от профильной организации

«31» мая 2024 г.


Карлов О.О.
(подпись и расшифровка подписи)

Москва 2024 г.



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА – Российский технологический университет»
РТУ МИРЭА

Институт информационных технологий (ИИТ)
Кафедра вычислительной техники (ВТ)

ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ НА УЧЕБНУЮ ПРАКТИКУ
Ознакомительная практика

Студенту 1 курса учебной группы ИКБО-44-23

Закирову Искандарходже Акрамовичу

Место и время практики: ООО «СНС Волга» на производственном участке сервисных и пуско-наладочных работ систем ЧПУ, с 13 февраля 2024 г. по 30 мая 2024 г.

Должность на практике: программист

1. СОДЕРЖАНИЕ ПРАКТИКИ:

1.1. Изучить: предметную область, существующие решения и технологии разработки программных систем для конвертации изображений в ASCII-art.

1.2. Практически выполнить: анализ предметной области, анализ полных или частичных аналогов разрабатываемой системы, обоснование выбранных инструментов разработки системы, разработку рабочего макета/прототипа системы или полностью готового модуля системы в соответствии с выбранной моделью жизненного цикла.

1.3. Ознакомиться: с технологиями разработки программных систем для конвертации изображения в ASCII-art.


2. ДОПОЛНИТЕЛЬНОЕ ЗАДАНИЕ: нет

3. ОРГАНИЗАЦИОННО-МЕТОДИЧЕСКИЕ УКАЗАНИЯ: в соответствии с методическими указаниями по учебной практике по направлению 09.03.04 «Программная инженерия»

Руководитель практики от кафедры
«09» февраля 2024 г.


Подпись (Холмогоров В.В.)

Руководитель практики от профильной организации
«09» февраля 2024 г.


Подпись (Карлов О.О.)



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА – Российский технологический университет»
РТУ МИРЭА

**РАБОЧИЙ ГРАФИК
ПРОВЕДЕНИЯ ОЗНАКОМИТЕЛЬНОЙ ПРАКТИКИ**

студента Закирова И.А. 1 курса группы ИКБО-44-23очной формы обучения, обучающегося по направлению подготовки 09.03.04 Программная инженерия

Неделя	Сроки выполнения	Этап	Отметка о выполнении
1	09.02.2024	Подготовительный этап, включающий в себя организационное собрание (Вводная лекция о порядке организации и прохождения производственной практики, инструктаж по технике безопасности, получение задания на практику)	
5	23.02.2024	Исследовательский этап (формирование введения, включающего постановку проблемы, описание общего метода решения, предмета и объекта исследования и постановку задачи на практику, а также исследовательского раздела отчёта с описанием предметной области, анализа аналогов, постановкой задачи на разработку, а также требованиями к системе)	
5	10.03.2024	Аналитический этап (формирование аналитического раздела отчёта, обоснование выбранных инструментов разработки, обоснование методов разработки и алгоритмической базы системы) Представление и защита руководителю структурированного материала: введения и исследовательского раздела отчёта и презентационного материала к нему	
8	26.04.2024	Технологический этап (формирование технологического раздела отчёта, обоснование выбранных инструментов разработки, обоснование методов разработки и алгоритмической базы системы, описание алгоритма работы системы и руководство пользователя)	
8	17.05.2024	Представление и защита руководителю структурированного материала: технологического раздела отчёта и презентационного материала к нему	
11	31.05.2024	Подготовка и защита окончательной версии отчёта, презентации и разрабатываемой системы	

Задание получил
«09» февраля 2024 г.


Подпись

(Закиров И.А.)

СОГЛАСОВАНО:
Заведующий кафедрой:

«09» февраля 2024 г.


Подпись

(Платонова О.В.)

Проведенные инструктажи:

Охрана труда:

Инструктирующий


Подпись

«09» февраля 2024 г.

Карлов О.О., руководитель
Департамента
станкостроения
Закиров И.А.

Инструктируемый


Подпись

Техника безопасности:

Инструктирующий


Подпись

«09» февраля 2024 г.

Карлов О.О., руководитель
Департамента
станкостроения
Закиров И.А.

Инструктируемый


Подпись

Пожарная безопасность:

Инструктирующий


Подпись

«09» февраля 2024 г.


Карлов О.О., руководитель
Департамента
станкостроения
Закиров И.А.

Инструктируемый


Подпись

С правилами внутреннего распорядка ознакомлен:

«09» февраля 2024 г.


Подпись

Закиров И.А.

Содержание практики и планируемые результаты согласованы с руководителем практики от профильной организации

Руководитель практики от кафедры

 /Холмогоров В.В., аспирант/

Руководитель практики от профильной организации Департамента станкостроения

 /Карлов О.О., руководитель

Обучающийся

 /Закиров И.А./

Согласовано:

Заведующий кафедрой

 /Платонова О.В., к.т.н., доцент/

АННОТАЦИЯ

В данной практической работе по ознакомительной практике была реализована программная система, решающая проблему конвертации изображений в ASCII-art, на основе следующей функциональности: конвертация изображения, отправка получившегося изображения пользователю.

Данная работа посвящена теме «Программная система конвертации изображений в ASCII-art». Работа состоит из 27 страниц, включает 10 рисунков и 2 таблицы.

Во Введении обоснована актуальность выбранной темы, определены объект и предмет исследования.

В разделе 1 представлены результаты анализа предметной области, обзора и анализа существующих решений, постановки задачи на разработку программной системы и определения основных требований к системе.

В разделе 2 представлены обоснование средств проектирования, модель процессов системы, блок-схем алгоритма взаимодействия пользователя с системой и модель общей структуры программной системы, обоснование выбора среды и средств разработки программной системы, общее описание системы и её основные функции

В разделе 3 представлены результаты описания программной сущности, а также приведено руководство пользователя по работе с данной системой, а именно скриншоты каждого действия программы и результат ее работы.

В Заключение приведены краткие результаты выполненной работы по каждому из подразделов и вклад системы в решение проблемы конвертации изображений.

В приложении представлены изображения – слайды презентации для данной практики.

СОДЕРЖАНИЕ

АННОТАЦИЯ.....	1
СОДЕРЖАНИЕ	7
ВВЕДЕНИЕ.....	9
1 ИССЛЕДОВАТЕЛЬСКИЙ РАЗДЕЛ	10
1.1 Анализ предметной области.....	10
1.2 Характеристика предмета и объекта исследования.....	11
1.3 Обзор и анализ существующих решений в области конвертации изображений в ASCII-art.....	11
1.4 Постановка цели и задач на разработку.....	12
1.5 Определение основных требований к программной системе	13
1.5.1 Требования к функциональным характеристикам	13
1.5.1.1 Требования к взаимодействию между пользовательской частью и базой данных	13
1.5.1.2 Требования к пользовательской части.....	13
1.5.2 Требования к составу и параметрам технических средств	14
1.5.3 Требования к информационной и программной совместимости.....	14
1.5.4 Требования к эргономике	14
1.6 Выводы по разделу 1.....	14
2 АНАЛИТИЧЕСКИЙ РАЗДЕЛ	16
2.1 Модель работы программной системы	16
2.2 Выбор инструментов для разработки.....	18
2.3 Описание функций, реализуемых программной системой.....	18
2.4 Выводы по разделу 2.....	18

3. ТЕХНОЛОГИЧЕСКИЙ РАЗДЕЛ	20
3.1. Программные сущности	20
3.2 Руководство пользователя по работе с системой	21
3.3 Выводы по разделу 3.....	25
ЗАКЛЮЧЕНИЕ	26
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	27

ВВЕДЕНИЕ

Темой проектной работы для ознакомительной практики является «Программная система конвертации изображений в ASCII-art». Сегодня в мире информационных технологий все больше людей стремятся к творческому самовыражению и оригинальным способам обработки изображений. ASCII-art – это один из таких способов, который позволяет создать уникальные композиции из текстовых символов. Из-за невозможности стандартных графических редакторов преобразовывать обычные изображения в изображения из символов ASCII, пользователи не могут легко превратить привычные изображения в ASCII-арты и поделиться ими со своими друзьями или использовать их в развлекательных целях.

В связи с вышеописанными проблемами возникает большая необходимость создания такой системы, которая может помочь людям легко и удобно преобразовывать изображения в ASCII-арты. Самым удобным и доступным вариантом такой системы является консольное приложение.

Для разработки указанной программной системы требуется решить следующие основные задачи:

- провести анализ предметной области;
- провести анализ существующих решений в рассматриваемой предметной области;
- определить основные требования к программной системе;
- провести моделирование процессов, происходящих при работе системы;
- спроектировать структуру программной системы;
- разработать программную систему.

Объектом исследования данного проекта является консольное приложение, которое будет конвертировать изображения в ASCII-art.

Предметом исследования являются существующие методы и подходы разработки, используемые в аналогах.

1 ИССЛЕДОВАТЕЛЬСКИЙ РАЗДЕЛ

1.1 Анализ предметной области

Предметная область программной системы конвертации изображений объединяет в себе элементы двух областей знания – разработка интерактивного программного приложения и системы преобразование изображения. Так как требуемое преобразование изображения возможно только с помощью технологий компьютерного зрения, то также предметная область содержит элементы области знания компьютерного зрения. Проведён анализ каждой из указанных областей. Процесс преобразования изображения, являющийся основным элементом данной системы и целью её создания имеет следующие пункты: конвертация изображений в ASCII-art; возможность создания запроса на конвертацию изображения и его получение.

Процесс взаимодействия пользователя с изображением является простым: пользователь посылает изображение на консоль и получает в ответ конечный результат.

Далее рассмотрим процесс разработки программной системы. Для её разработки используется язык программирования Python, так как он является простым и имеет широкий набор библиотек и фреймворков, а также поддерживает процедурный подход программирования.

«Процедурное программирование» – это программирование на императивном языке, при котором последовательно выполняемые операторы можно собрать в подпрограммы, то есть более крупные целостные единицы кода, с помощью механизмов самого языка [1].

«Библиотека» – сборник подпрограмм или объектов, используемых для разработки программного обеспечения [2].

«ASCII» — стандарт кодирования знаков латинского алфавита, цифр, некоторых специальных знаков и управляющих последовательностей принятый в 1963 году, как основной способ представления текстовых данных в ЭВМ [3].

«Фреймворк» — это готовый набор инструментов, помогающий разработчику быстро создать продукт [4].

Электронная вычислительная машина (ЭВМ) — это комплекс технических, аппаратных и программных средств, предназначенных для автоматической обработки информации, вычислений и автоматического управления [5].

Данный язык программирования и методы разработки соответствуют основным требованиям, которые были описаны в разделе 1.5 настоящей работы и являются достаточными для создания различных промышленных программ. Большое количество документации, включая русскоязычную, а также множество доступных видеоуроков делают изучение этого стека технологий легким и быстрым.

1.2 Характеристика предмета и объекта исследования

По результатам исследования предметной области было определено, что объектом изучения станет программная система, включающая в себя программную систему и консольное приложение. Главной целью данного исследования является создание программного продукта с понятным интерфейсом, обеспечивающего эффективное взаимодействие с пользователями. Кроме того, важно предусмотреть функционал, который максимально увеличит вероятность успешного выполнения поставленных задач. Также отмечается необходимость принципа открытых информационных систем и свободного доступа к ней.

1.3 Обзор и анализ существующих решений в области конвертации изображений в ASCII-art

В ходе проведенного исследования было обнаружено несколько систем, которые могут представлять собой потенциальных конкурентов для разрабатываемой в рамках данной стажировки системы. Эти конкуренты включают в себя сайт ASCII-Generator.site и сайт asciiart.eu Для того чтобы провести более

детальный анализ, были определены следующие критерии сравнения: возможность преобразования требуемых изображений в ASCII-art, удобство использования и использование программы без подключения к интернету. Далее в таблице 1.1 представлен список выявленных конкурентов с уникальными порядковыми номерами: ASCII-Generator.site - 1, asciiart.eu- 2.

Таблица 1.1 – Сравнительный анализ аналогов разрабатываемой системы

Решение	Критерии		
	возможность преобразования требуемых изображений в ASCII-art	удобство использования	использование программы без подключения к интернету
1	да	нет	нет
2	да	да	нет

1.4 Постановка цели и задач на разработку

После проведения анализа предметной области и исследования существующих решений стало понятно, что не существует такой системы для преобразования изображений в ASCII-art, которая соответствовала бы критериям выше, поэтому есть необходимость создать такой продукт. Для выполнения данной задачи необходимо написать программу, которая будет преобразовывать изображения в ASCII-art. Она должна соответствовать следующим требованиям:

- возможность преобразования требуемых изображений в ASCII-art;
- удобство использования приложения;
- возможность использования приложения без подключения к интернету;

1.5 Определение основных требований к программной системе

1.5.1 Требования к функциональным характеристикам

Программа состоит из пользовательской части, а также базы данных, находящейся на сервере.

1.5.1.1 Требования к взаимодействию между пользовательской частью и базой данных

Для отправки адреса изображения на консоль, пользователю необходимо ввести команду, которая позволит передать указанный адрес на сервер. После этого, сервер извлечет из базы данных соответствующее изображение и передаст его обратно пользователю для отображения на консоли. Важно убедиться, что адрес изображения указан корректно и доступен для сервера, чтобы избежать ошибок при получении изображения.

1.5.1.2 Требования к пользовательской части

Консольное приложение, которое конвертирует изображения в ASCII-арт, должно иметь следующую функциональность:

- возможность загружать изображение с локального диска или по URL
- возможность выбора размера и разрешения ASCII-арта.
- возможность сохранения ASCII-арта в виде текстового файла или его вывод на консоль.
- возможность обработки ошибок и отображение сообщений об ошибках пользователю.

Это основные функциональные требования, которые обеспечат эффективную работу и удобство использования консольного приложения для конвертации изображений в ASCII-арт.

1.5.2 Требования к составу и параметрам технических средств

К компьютерной технике, на которой будет эксплуатироваться система, предъявляются следующие требования к аппаратному и программному окружениям, на основании минимальных системных требований к большинству современных программных средств:

- электричество;
- операционная система: Microsoft Windows 7 и выше, Mac OS X 10.6 или более поздней версии, операционная система семейства Linux версии ядра 4.0 и выше.

1.5.3 Требования к информационной и программной совместимости

При разработке системы должен быть применён язык программирования высокого уровня для скриптовых сценариев: Python.

1.5.4 Требования к эргономике

Интерфейс консольного приложения не должен быть перегружен большим количеством цветов и элементов. Пользователю должно быть понятно куда ведёт то или иное поле, чтобы ему было легче ориентироваться в приложении, чтобы он тратил время не на изучение интерфейса, а на решение своей проблемы.

1.6 Выводы по разделу 1

Был проведён анализ предметной области, а также сбор и анализ существующих решений, поставлена задача на разработку и определены основные требования.

Анализ предметной области показал, что консольное приложение, способное конвертировать изображения в ASCII-art, находится на пересечении двух предметных областей: программной системы и консольного приложения.

Обзор и анализ существующих решений по конвертации изображений в ASCII-art показал, что существуют аналоги и частичные аналоги, реализующие систему организации для конвертации изображений в ASCII-art. Их сравнительный анализ позволил выявить наилучшее сочетание элементов будущей программной системы. Это позволит избежать недостатков аналогов.

По результатам анализа предметной области, а также обзора и анализа существующих решений поставлена задача на разработку программной системы для конвертации изображений в ASCII-art.

На основе поставленной задачи к разрабатываемой программной системе предъявлены требования к функциональным характеристикам, к надёжности, к условиям эксплуатации, а также требования к информационной и программной совместимости и к составу и параметрам технических средств.

В результате, для эффективной работы консольного приложения по конвертации изображений в ASCII-арт необходимо уделять внимание как функциональным характеристикам, так и надёжности приложения. Важно, чтобы база данных хранила необходимую информацию и взаимодействие между пользовательской частью и базой данных происходило без ошибок. Пользовательская часть должна быть интуитивно понятной, с возможностью загружать, изменять и сохранять изображения. Надёжность приложения играет ключевую роль, поэтому необходимо обеспечить безопасность, стабильность и быструю работу программы, а также обработку ошибок. В целом, соблюдение всех требований к функциональным характеристикам и надёжности приложения обеспечит его эффективную работу и удобство использования для пользователей.

2 АНАЛИТИЧЕСКИЙ РАЗДЕЛ

2.1 Модель работы программной системы

Модель процессов работы программной системы позволяет понять, как друг с другом взаимодействуют её конкретные элементы и как с ней взаимодействует человек. Для понимания, в какой последовательности работают эти элементы и что нужно делать пользователю для правильной работы программы, необходимо знать алгоритм работы системы и алгоритм работы с системой. Для полного понимания алгоритма работы системы следует рассмотреть процесс с алгоритмической точки зрения: последовательность действий человека, который хочет конвертировать изображение в ASCII-art.

Для того, чтобы начать работу с приложением пользователю необходимо сначала перенести все изображения, которые он хочет преобразовать в ASCII, в папку, где находится само приложение. Также необходимо, чтобы файл был в формате .PNG, иначе конвертация не получится. Далее пользователь открывает приложение и ему выводится следующий текст: «Enter the name of the png file you want convert:». Пользователь должен вставить полное название файла. В случае, если имя файла было введено неправильно или же сам файл не существует, то выводится следующий текст: «Incorrect data entry!». Далее пользователю дается возможность повторить попытку: «Entry the path to the image field again!». Это будет продолжаться пока введенные данные не будут корректными.

Далее пользователь должен будет выбрать вес изображения от 100 до 1000. Для того, чтобы пользователь понял это выводится следующий текст: «Enter a value from 100 to 1000:». В случае, если пользователь превысил допустимое значение или же наоборот ввел меньшее значение, то программа выведет «Incorrect data entry! Please, repeat the input! ». Данное сообщение также выведется в случае, если были введены любые другие символы, кроме арабских цифр. Далее также дается возможность повторно ввести вес изображения: «Enter a value from 100 to 1000 again:».

Следующим шагом пользователь должен будет назвать файл, в который он хочет сохранить получившийся результат. Для этого ему выведется следующее сообщение: «Enter the name of the file you want to save the image to». После сохранения результата программа завершается и выводит следующий текст: «Your image has been saved to a file ***.txt».

Помимо вышеприведённых возможностей пользователю даётся возможность завершить программу на любом ее этапе. Для этого в начале всей программы выводится сообщение «If you want to stop the program, you can enter the text: exit». Слово «exit» является ключевым и выполняет функцию завершения программы на любом ее этапе. После завершения пользователю на экран выводится следующий текст: «The program has been stopped!».

Блок-схема данной модели работы программной системы показана на рисунке 2.1.

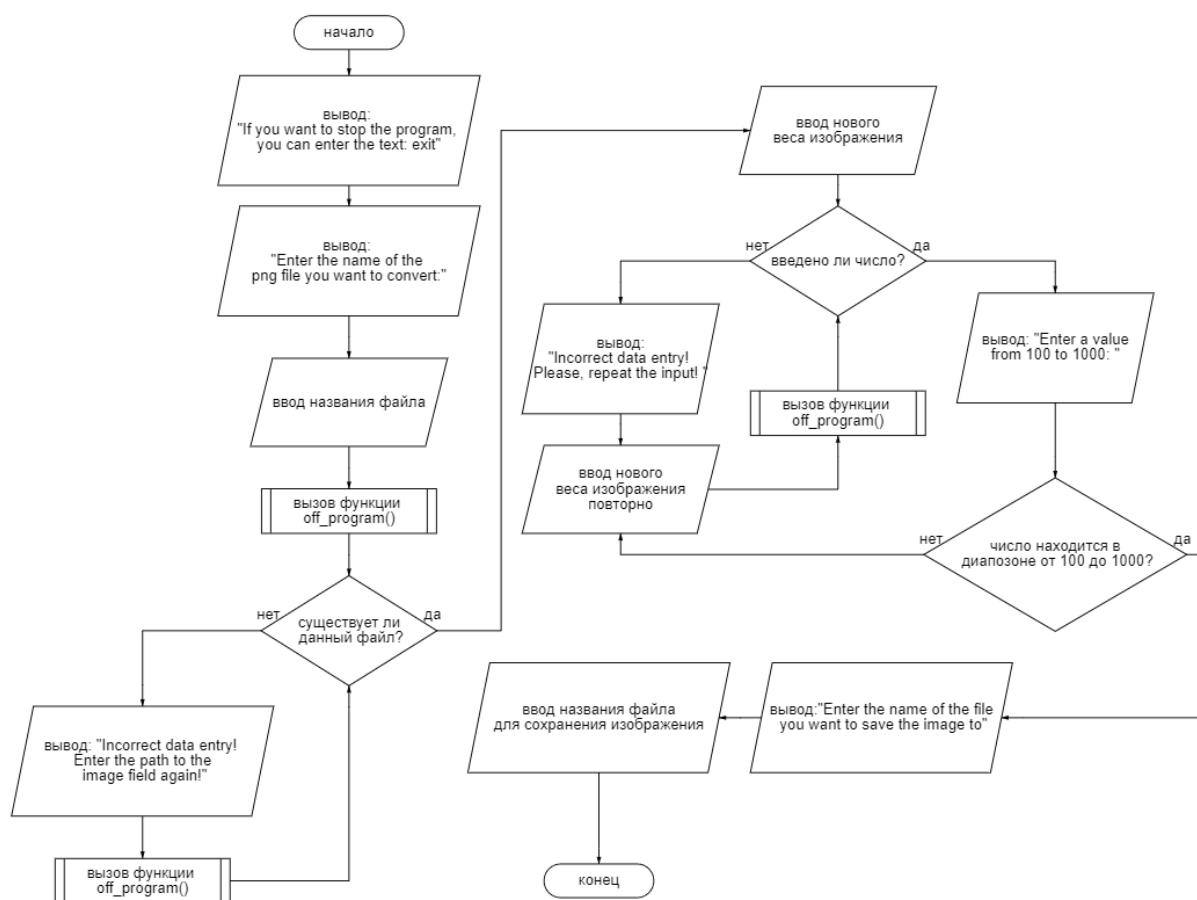


Рисунок 2.1 – Модель использования системы

2.2 Выбор инструментов для разработки

Для разработки данной программной системы, как было сказано в разделе 1.1, будет использоваться язык программирования Python. На данный момент существует несколько сред разработки, основным из которых являются Dreamweaver, Wing, Thonny и PyCharm.

При сравнении данных сред разработки по их удобству и возможностям, был сделан выбор в пользу PyCharm. Данная среда разработки является удобной для разработки. Также стоит отметить хороший дизайн интерфейса. К тому же я к ней привык и уже установил все необходимые библиотеки. Также данный продукт является бесплатным. Так как я студент и у меня ограниченные финансовые возможности, это для меня самый оптимальный вариант.

2.3 Описание функций, реализуемых программной системой

В данной системе реализованы следующие функциональные возможности:

- конвертация: конвертация изображения;
- создание: создание текстового документа;
- сохранение: сохранение получившегося результата в текстовый документ;
- обработка запросов: получение команд от пользователя.

Вышеописанные функциональные возможности в программной системе реализуются совокупностью взаимодействующих объектов и функций, входящих в состав консольного приложения. При этом, для каждой функции описаны сценарии поведения при выходе приложения или сервера из строя.

2.4 Выводы по разделу 2

В результате выполнения аналитического раздела был определен общий алгоритм работы системы, а также определена структура и модель системы. Кроме этого, были изучены и выбраны среда и средства реализации программной

системы, приведено общее описание разрабатываемой программной системы, описание реализуемых функций и описание работы пользователя с системой.

В общем описании программной системы был приведен основной сценарий использования данной системы с описанием действий в нём.

В описании реализуемых функций были приведены функциональные блоки, обеспечивающие работу пользователя с системой.

В описании работы пользователей с программной системой был приведён фрагмент руководства оператора, в котором были подробно изложены все этапы работы пользователя с системой.

3. ТЕХНОЛОГИЧЕСКИЙ РАЗДЕЛ

3.1. Программные сущности

Для создания данной программной системы был написан код на языке программирования Python. В этом коде были использованы функции и переменные.

В качестве функций была написана функция `off_program`, при помощи которой пользователь может отключить программу и выйти из нее. Для этого пользователем вводится ключевое слово «exit», которая передается переменной `data_`, параметру функции.

Также в коде используются следующие переменные:

- `img_flag` – хранит в себе значение `True`;
- `path` – принимает в себя название файла;
- `img` – в этой переменной хранится файл с изображением, открытый при помощи функции `PIL.Image.open`;
- `width`, `height` – принимают в себя ширину и высоту изображения соответственно;
- `aspect_ratio` – расчет соотношения сторон изображения путем деления высоты на ширину;
- `new_width` – принимает от пользователя новый вес изображения;
- `chars` – список символов, которые будут использоваться для отображения пикселей;
- `pixels` – получение данных о пикселях изображения;
- `new_pixels` – преобразование данных пикселей в текстовые символы;
- `new_pixels_count` – подсчет количества символов;
- `ascii_image` – разделение строки символов на строки для каждой строки изображения;
- `name_file` – ввод имени файла для сохранения ASCII рисунка;

Далее опишем использованные методы:

- `exists` – возвращает `True`, если аргумент `path` ссылается на существующий путь в файловой системе или дескриптор открытого файла. Возвращает `False` для неработающих символических ссылок [6];
- `size` – функция для получения ширины и высоты изображения;
- `resize` – применяется для изменения разрешения обрезанного изображения;
- `getdata` – возвращает содержимое изображения в последовательности [7];
- `join` – это строковый метод, который применяется для объединения элементов итерируемого объекта (например, списка или кортежа) в одну строку [8];
- `write` – используется для записи в файлы, открытые в режиме записи [9];

3.2 Руководство пользователя по работе с системой

В данном разделе приведено руководство пользователя по работе с данной системой. При входе в приложение на консоль выводится запись. Ее можно увидеть на рисунке 3.1.

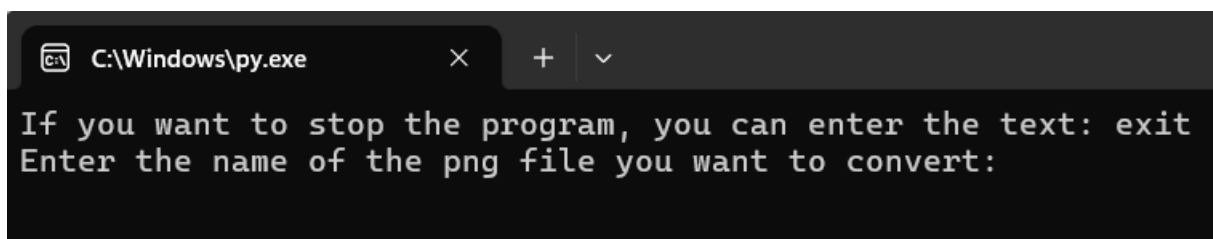


Рисунок 3.1 – начальная запись на консоли

Далее после ввода названия файла программа переходит к следующей итерации. Она изображена на рисунке 3.2.

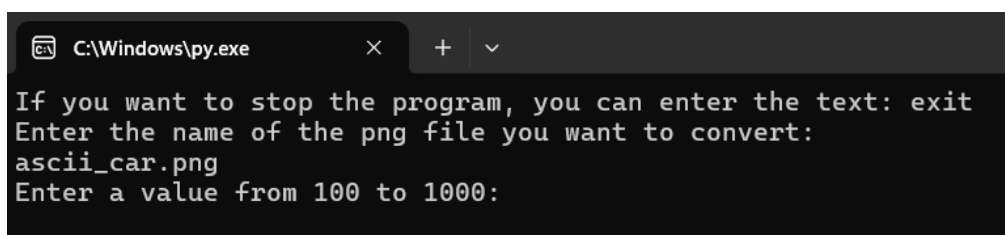
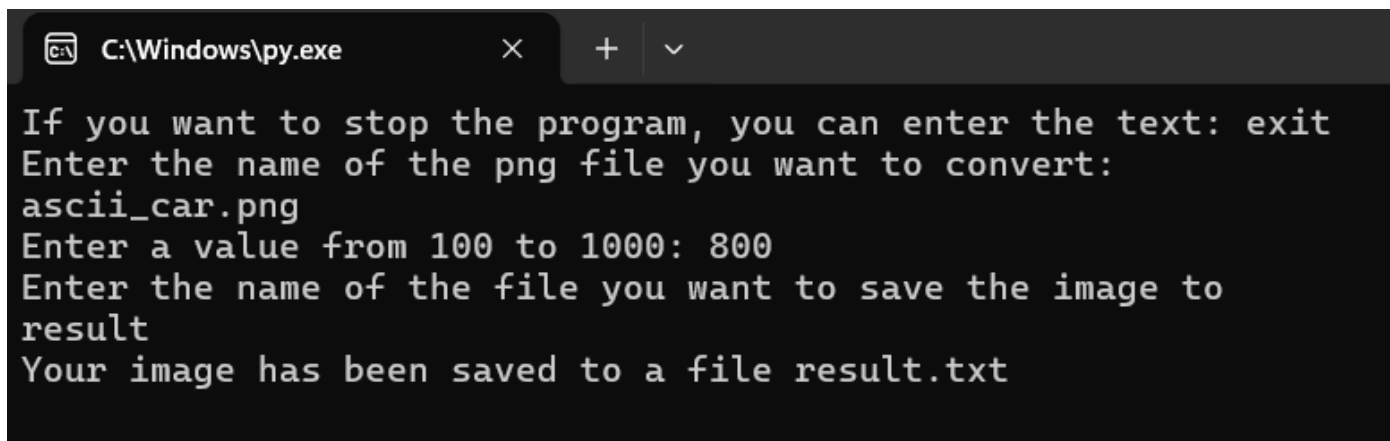


Рисунок 3.2 – Ввод названия изображения

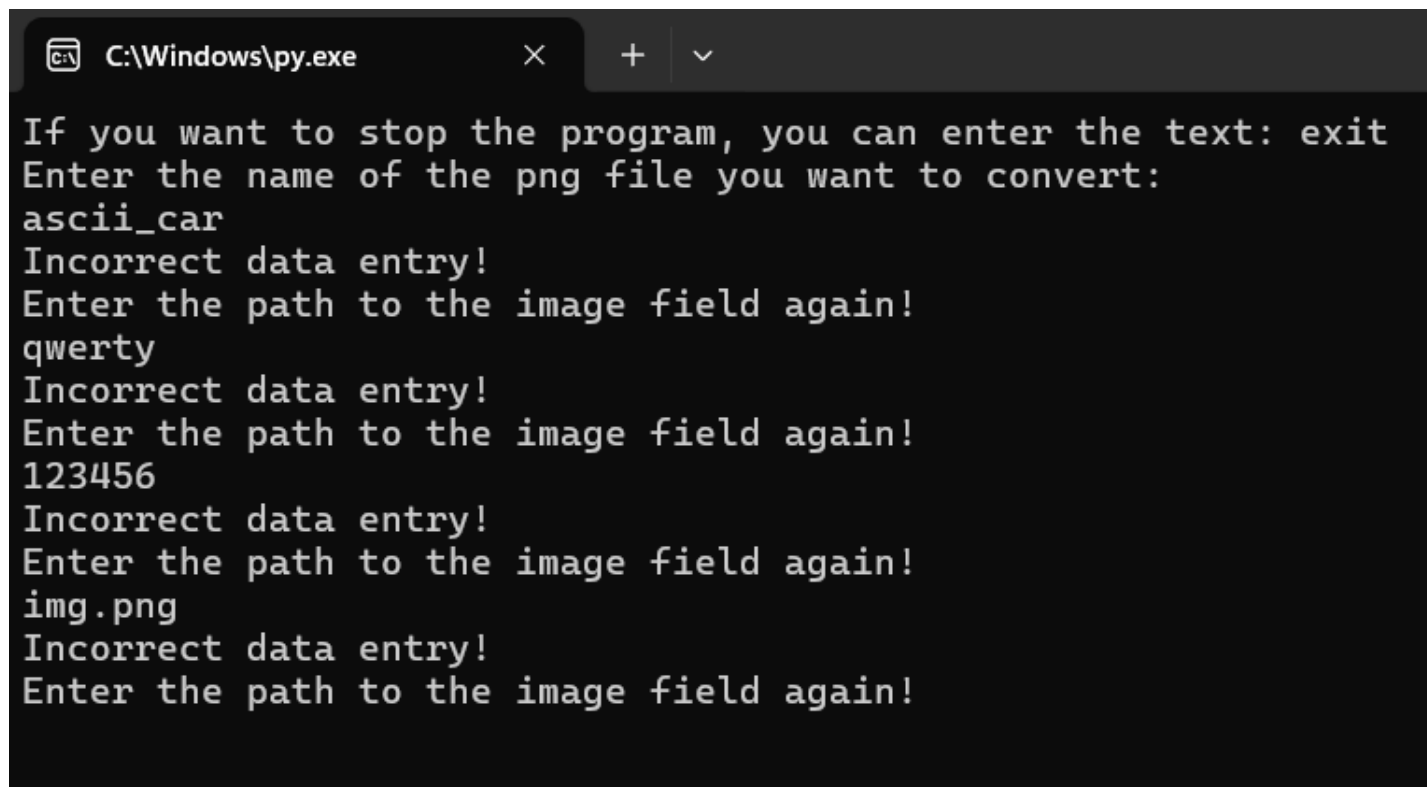
Теперь введем случайное число из нужного диапазона. После этого программа запрашивает название файла, в который хотим сохранить итоговый результат. После того как мы назвали файл, программа говорит, куда было сохранено изображение. Итоговый результат работы программы показан на рисунке 3.3.



```
C:\Windows\py.exe
If you want to stop the program, you can enter the text: exit
Enter the name of the png file you want to convert:
ascii_car.png
Enter a value from 100 to 1000: 800
Enter the name of the file you want to save the image to
result
Your image has been saved to a file result.txt
```

Рисунок 3.3 – Ввод веса изображения и конечный результат

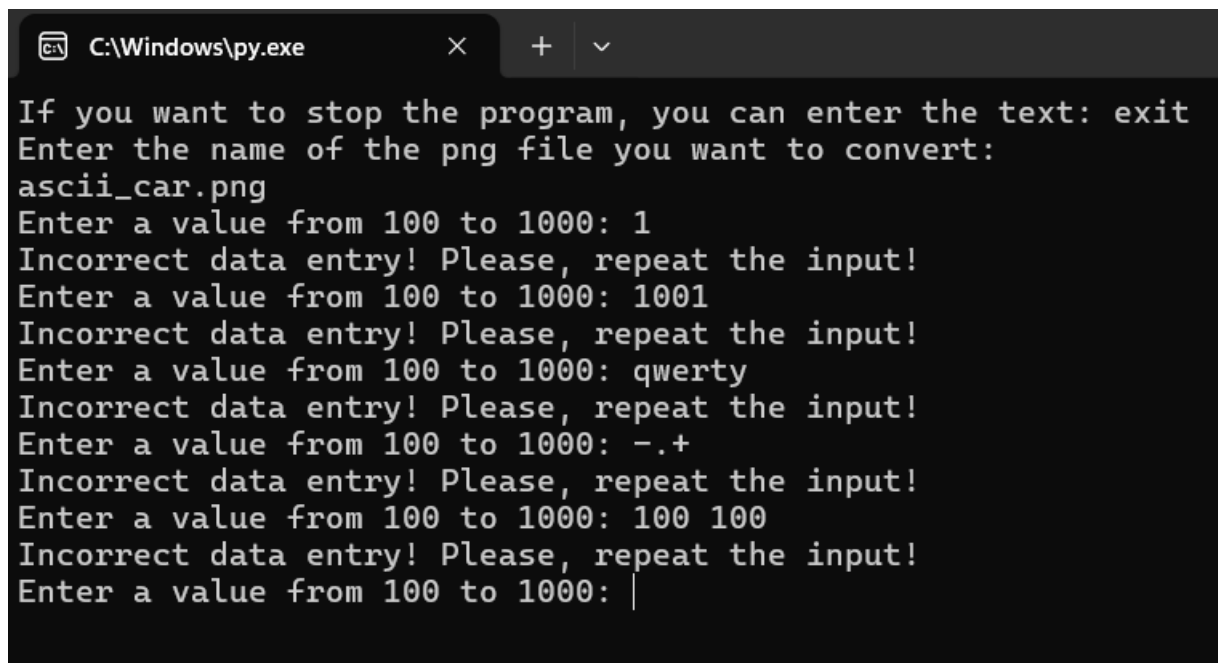
Но стоит отметить, что программа может не сработать, если ввести некорректные данные. Пример такого случая можно увидеть на рисунке 3.4.



```
C:\Windows\py.exe
If you want to stop the program, you can enter the text: exit
Enter the name of the png file you want to convert:
ascii_car
Incorrect data entry!
Enter the path to the image field again!
qwerty
Incorrect data entry!
Enter the path to the image field again!
123456
Incorrect data entry!
Enter the path to the image field again!
img.png
Incorrect data entry!
Enter the path to the image field again!
```

Рисунок 3.4 – Неправильный ввод названия файла

На рисунке 3.5 изображен ввод неправильных данных.

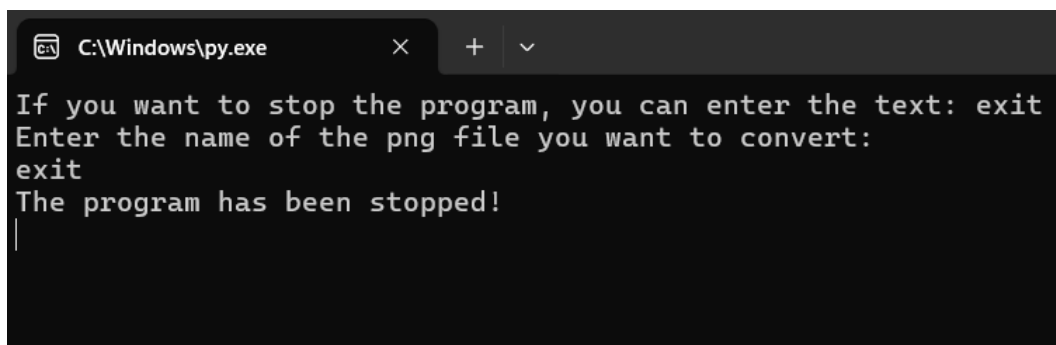


```
C:\Windows\py.exe
If you want to stop the program, you can enter the text: exit
Enter the name of the png file you want to convert:
ascii_car.png
Enter a value from 100 to 1000: 1
Incorrect data entry! Please, repeat the input!
Enter a value from 100 to 1000: 1001
Incorrect data entry! Please, repeat the input!
Enter a value from 100 to 1000: qwerty
Incorrect data entry! Please, repeat the input!
Enter a value from 100 to 1000: -.+
Incorrect data entry! Please, repeat the input!
Enter a value from 100 to 1000: 100 100
Incorrect data entry! Please, repeat the input!
Enter a value from 100 to 1000: |
```

Рисунок 3.5 – Неправильный ввод веса изображения

Выше мы описали все возможные некорректные вводы данных, которые пользователь может случайно ввести.

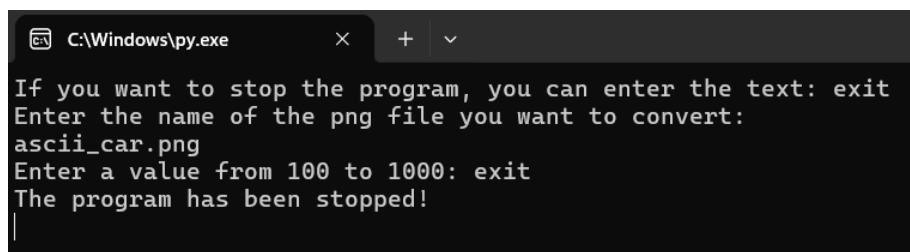
А далее мы покажем случай, когда можно будет досрочно завершить работу программы (рис. 3.6).



```
C:\Windows\py.exe
If you want to stop the program, you can enter the text: exit
Enter the name of the png file you want to convert:
exit
The program has been stopped!
|
```

Рисунок 3.6 – Досрочное завершение программы №1

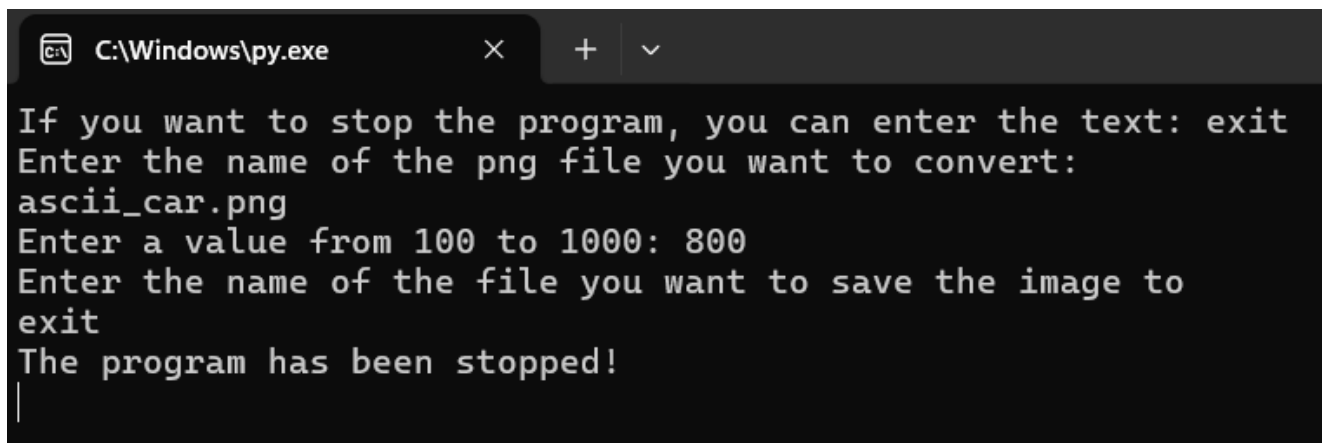
А далее приведено досрочное завершение программы №2 (рис. 3.7).



```
C:\Windows\py.exe
If you want to stop the program, you can enter the text: exit
Enter the name of the png file you want to convert:
ascii_car.png
Enter a value from 100 to 1000: exit
The program has been stopped!
|
```

Рисунок 3.7 – Досрочное завершение программы №2

А далее приведено досрочное завершение программы №3 (рис. 3.8).



```
C:\Windows\py.exe
If you want to stop the program, you can enter the text: exit
Enter the name of the png file you want to convert:
ascii_car.png
Enter a value from 100 to 1000: 800
Enter the name of the file you want to save the image to
exit
The program has been stopped!
|
```

Рисунок 3.8 – Досрочное завершение программы №3

И в конце покажем результат преобразования изображения, который у нас в итоге получился (рис. 3.9).

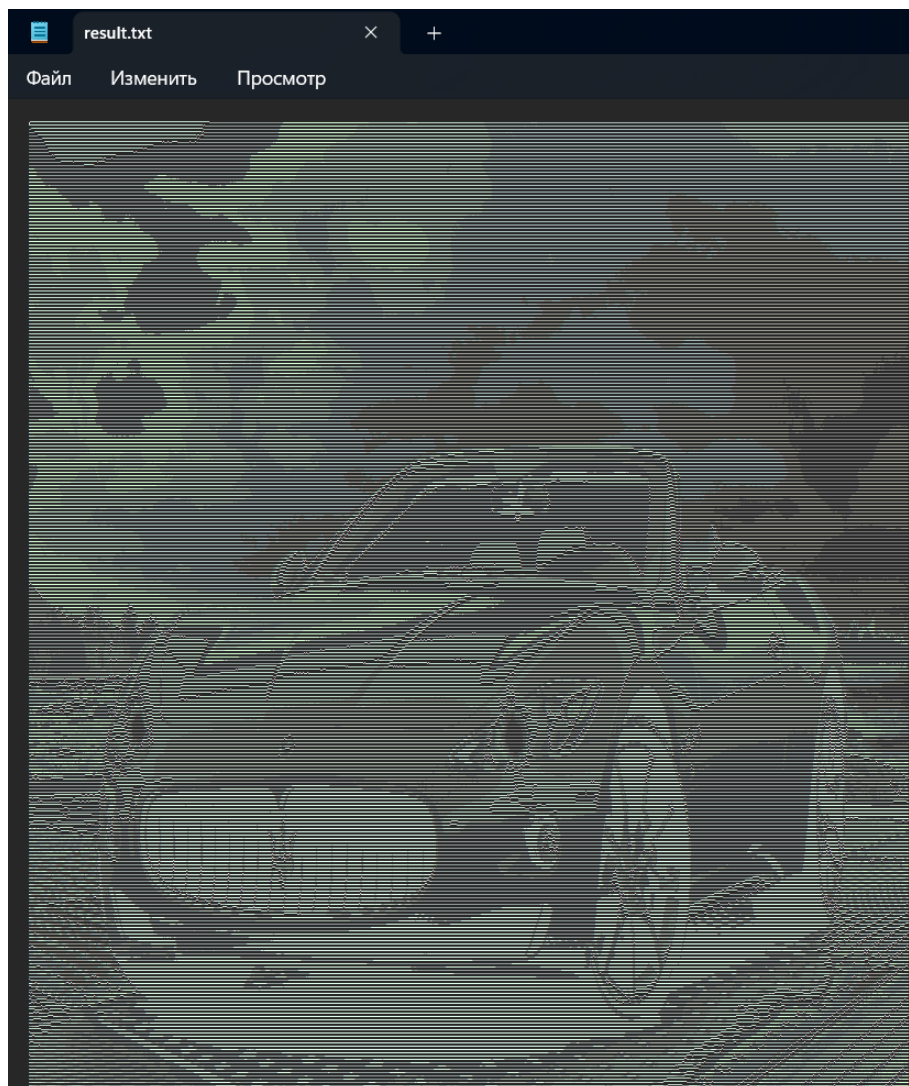


Рисунок 3.9 – Результат конвертации изображения

3.3 Выводы по разделу 3

В результате выполнения технологического раздела были описаны основные программные сущности, было проведено подробное руководство пользователя по работе с системой. Кроме того, был показан результат преобразования изображения в ASCII-art.

В описании программных сущностей были подробно описаны используемые функции, переменные и методы, и их назначение.

В описании руководства пользователя была продемонстрирована работа программы на примерах ввода корректных данных и обработки ошибочных вариантов. Пользователь получил инструкции о том, как правильно воспользоваться каждой функцией и как избежать возможных проблем в процессе работы.

ЗАКЛЮЧЕНИЕ

В ходе выполнения проектной работы решены следующие задачи:

- проведён анализ предметной области с определением объекта исследования;
- проведён обзор и анализ существующих решений, определены частичные аналоги, их достоинства и недостатки;
- поставлена задача на разработку программной системы и определены основные требования к ней;
- обоснованы средства проектирования программной системы;
- разработана модель процессов работы системы;
- спроектирован алгоритм работы системы в виде блок-схемы;
- спроектирована общая структура программной системы;
- обоснованы среда и средства для реализации проекта;
- приведено общее описание программной системы и её функций;
- разработана программная система конвертации изображения в ASCII-art.

Разработанная система призвана внести вклад в проблему конвертации изображения в ASCII-art.

При дальнейшей разработке и масштабировании данная система сможет занять своё место в нише продуктов, направленных на решении проблемы конвертации. Кроме этого, система имеет потенциал для добавления новых функциональных возможностей, которые смогут выгодно отличить её от своих аналогов и поможет ещё эффективнее выполнять поставленную перед системой задачу.

Все задачи, стоящие перед проектной работой, выполнены.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Интернет-энциклопедия Википедия: сайт. — URL: (https://ru.wikipedia.org/wiki/Процедурное_программирование) (дата обращения: 16.05.2024)
- 2 Интернет-энциклопедия Википедия: сайт. — URL: ([https://ru.wikipedia.org/wiki/Библиотека_\(программирование\)](https://ru.wikipedia.org/wiki/Библиотека_(программирование))) (дата обращения: 16.05.2024)
- 3 Интернет-энциклопедия Википедия: сайт. — URL: (<https://ru.wikipedia.org/wiki/ASCII>) (дата обращения: 16.05.2024)
- 4 Яндекс образование: сайт. — URL: (<https://education.yandex.ru/journal/chto-takoe-frejmwork>) (дата обращения: 16.05.2024)
- 5 Интернет-энциклопедия Википедия: сайт. — URL: (https://ru.wikipedia.org/wiki/Электронная_вычислительная_машина) (дата обращения: 16.05.2024)
- 6 Справочная документация по языку Python3: сайт. — URL: (<https://docs-python.ru/standart-library/modul-os-path-python/funktsija-exists-modulja-os-path/>) (дата обращения: 22.05.2024)
- 7 Справочная документация по языку Python3: сайт. — URL: (<https://docs-python.ru/packages/biblioteka-pillow-python/obekt-image/>) (дата обращения: 22.05.2024)
- 8 российский хостинг Timeweb: сайт. — URL: ([https://timeweb.com/ru/community/articles/konkatenaciya-strok-python#:~:text=Метод%20.join%20\(\)%20-%20это,количества%20необходимых%20операций%20выделения%20п%20амяти](https://timeweb.com/ru/community/articles/konkatenaciya-strok-python#:~:text=Метод%20.join%20()%20-%20это,количества%20необходимых%20операций%20выделения%20п%20амяти)) (дата обращения: 22.05.2024)
- 9 образовательный блог о языке программирования Python PythonRu: сайт. — URL: (<https://pythonru.com/osnovy/fajly-v-python-vvod-vyvod>) (дата обращения: 22.05.2024)