Бюджетное профессиональное образовательное учреждение Вологодской области

«Череповецкий лесомеханический техникум им. В.П. Чкалова»

 Специальность 09.02.07 «Информационные системы и программирование»

ОТЧЕТ ПО ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ

ПП по ПМ.02 Осуществление интеграции программных модулей

Выполнил студент 2 курса группы ИС-\_\_\_\_

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

подпись \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

место

практики \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

наименование юридического лица, ФИО ИП

Период прохождения:

с  «\_\_\_» \_\_\_\_\_\_\_ 2024 г.

по «\_\_\_» \_\_\_\_\_\_\_ 2024 г.

Руководитель практики от

предприятия

должность\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

подпись\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

МП Руководитель практики от

техникума: Материкова А.А.

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

Оценка:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2024 года

г. Череповец

2024

Содержание

1.Общая характеристика организации…………………………………………4

1.1Руководство…………………………………………………………………..5

1.2 Распорядок работы………………………………………………………….6

2.Инфромация про техника……………………………………………………..7

2.1Должностные обязанности…………………………………………………..7

2.2 Права…………………………………………………………………………8

2.3 Обязанности и ответственность……………………………………………9

3 Программные модули……………………………………………………….10

3.1 Интеграция программных модулей………………………………………10

3.2 Инспектирование компонентов программного обеспечения…………..16

4 Описание стандарта кодирования python…………………………………..18

5 Отладка модулей…………………………………………………………….23

6 Тестовый сценарий………………………………………………………….25

7 Заключение…………………………………………………………………..30

8 Литература…………………………………………………………………..31

9 Приложение…………………………………………………………………32

Введение:

Задачами данной практики являются подготовка обучающихся осознанному и углубленному изучению дисциплин, привитие им практических умений и получение первичных профессиональных навыков по выбранной специальности.

Целями производственной практики (по профилю специальности) являются:

закрепление и совершенствование общих и профессиональных

компетенций:

ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам;

ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности;

ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность B профессиональной сфере, использовать знания по финансовой грамотности в различных жизненных ситуациях;

ОК 04. Эффективно взаимодействовать и работать в коллективе и команде; ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;

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;

ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях;

ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности;

ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках.

ПК 2.1. Разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент

ПК 2.2. Выполнять интеграцию модулей в программное обеспечение

ПК 2.3. Выполнять отладку программного модуля с использованием специализированных программных средств использованием

ПК 2.4. Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения

ПК 2.5. Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования

Общая характеристика предприятия (организации)

Малленом Системс – ведущая российская компания в области разработки и внедрения систем компьютерного зрения и промышленной видеоаналитики на основе технологий машинного зрения и искусственного интеллекта (машинное обучение, нейронные сети глубокого обучения).

Компания была создана в 2011 году на базе команды ученых и программистов Санкт-Петербургского политехнического университета Петра Великого при поддержке инвестиционной компании «Малленом».   
Профиль компании – реализация наукоемких IT-проектов в сфере транспорта, машиностроения, нефтегазовой, металлургической, пищевой, фармацевтической, алмазодобывающей, атомной промышленности и других отраслях.   
Продукция Малленом Системс представлена в большинстве регионов РФ, странах СНГ и ЕС. Уникальный опыт внедрений и ноу-хау компании обеспечивают быструю разработку новых высокотехнологичных продуктов и решений.   
На протяжении 10 лет компания является официальным партнером-интегратором в России и СНГ компании Cognex — мирового лидера в сфере машинного зрения и промышленной идентификации.   
С 2022 года Малленом Системс представляет на рынке РФ и ЕАЭС компанию Hikrobot – дочернее подразделение Hikvision по производству оборудования машинного зрения и мобильных роботов.   
Малленом Системс – участник национального рейтинга российских быстрорастущих технологических компаний «ТехУспех», разработанный РВК.

1 Высшее руководство

Генеральный директор - отвечает за сохранность и эффективное использование имущества предприятия, за последствия принимаемых решений

Директор по развитию -  отвечает за стратегическое планирование и реализацию проектов, направленных на рост бизнеса, улучшение конкурентных преимуществ, расширение рынков и увеличение прибыли

Технический директор – отвечает за технологическое развитие

Коммерческий директор – анализирует работу компании

2 Отдел разработки программного обеспечения:

Руководитель отдела разработки

разработка алгоритмов компьютерного зрения, программного обеспечения для анализа видео, интеграция с оборудованием. Может быть разделён на более мелкие команды, специализирующиеся на отдельных аспектах

3 Отдел тестирования и контроля качества

Инженерные службы:

Руководитель направления системной интеграции: отвечает за управление проектами, связанными с интеграцией различных IT-систем и технологий в единую инфраструктуру

Отдел системной интеграции - отвечает за объединение различных подсистем и компонентов в единую работающую систему, обеспечивая их совместимость и взаимодействие

Отдел технической поддержки - отвечает за обеспечение помощи пользователям и клиентам в решении проблем, связанных с использованием продуктов или услуг компании

Маркетинг и продажи

Отдел маркетинга - отвечает за разработку и реализацию стратегий, направленных на продвижение продуктов или услуг компании, а также на привлечение и удержание клиентов

Отдел продаж - отвечает за реализацию продуктов или услуг компании и взаимодействие с клиентами для достижения коммерческих целей

Финансовый и административный отдел

Бухгалтерия занимается - сбором и обработкой полной и достоверной информации о деятельности организации

HR-отдел занимается сотрудниками их подбором и обучением

Юридический отдел

Юридический консультант занимается - предоставление рекомендаций и разъяснений по различным юридическим вопросам

Распорядок работы предприятия

В компании "Малленом Системс" действует график работы 5/2, с 09:00 до 18:00. В штате компании есть отдельный специалист по охране труда, который проводит вводные инструктажи при приеме на работу и практике, а также занимается выдачей пропусков для пусконаладочных работ инженеров. В компании 20.09.2018 г. была проведена специальная оценка условий труда, согласно которой рабочие места, на территории которых установлены вредные производственные факторы, отсутствуют

Техник

2. Должностные обязанности

Техник выполняет следующие должностные обязанности:

2.1 Выполняет работу по проведению необходимых технических расчетов;

2.2 Осуществляет наладку, настройку, регулировку и опытную проверку оборудования и систем, следит за его исправным состоянием;

2.3 Принимает участие в проведение экспериментов и испытаний;

2.4 Принимает участие в разработке программ, инструкций и другой технической документации, в изготовлении макетов, а также в испытаниях и экспериментальных работах;

2.5 Выполняет работу по сбору, обработке и накоплению исходных материалов, данных статистической отчетности, научно-технической информации;

2.6 Составляет описания проводимых работ, необходимые спецификации, диаграммы, таблицы, графики и другую техническую документацию;

2.7 Выполняет работу по оформлению плановой и отчетной документации, вносит необходимые изменения и исправления в техническую документацию в соответствии с решениями, принятыми при рассмотрении и обсуждении выполняемой работы;

2.8 Систематизирует, обрабатывает и подготавливает данные для составления отчетов о работе;

2.9 Принимает необходимые меры по использованию в работе современных технических средств.

3. Права

Техник имеет право:

3.1. Участвовать в обсуждении проектов решений, в совещаниях по их подготовке и выполнению.

3.2. Запрашивать у непосредственного руководителя разъяснения и уточнения по данным поручениям, выданным заданиям.

3.3. Запрашивать по поручению непосредственного руководителя и получать от других работников организации необходимую информацию, документы, необходимые для исполнения поручения.

3.4. Знакомиться с проектами решений руководства, касающихся выполняемой им функции, с документами, определяющими его права и обязанности по занимаемой должности, критерии оценки качества исполнения своих трудовых функций.

3.5. Вносить на рассмотрение своего непосредственного руководителя предложения по организации труда в рамках своих трудовых функций.

3.6. Участвовать в обсуждении вопросов, касающихся исполняемых должностных обязанностей.

4. Обязанности и ответственность

Техник обязан:

4.1. Соблюдать локально-нормативные акты Общества.

4.2. Не разглашать информацию и сведения, являющиеся коммерческой тайной.

4.3. Использовать только принятые в Обществе программные инструменты и технологию разработки программного обеспечения.

4.4. Соблюдать трудовую и производственную дисциплину, правила и нормы охраны труда, требования производственной санитарии и гигиены, требования противопожарной безопасности.

Ведущий программист привлекается к ответственности:

4.5. За ненадлежащее исполнение или неисполнение своих должностных обязанностей, предусмотренных настоящей должностной инструкцией, в порядке, установленном действующим трудовым законодательством Российской Федерации;

4.6. За правонарушения и преступления, совершенные в процессе своей деятельности, в порядке, установленном действующим административным, уголовным и гражданским законодательством Российской Федерации;

4.7. За причинение ущерба организации в порядке, установленном действующим трудовым законодательством Российской Федерации.

Программные модули

Интеграция программных модулей

Интеграция программных модулей — это процесс объединения различных программных компонентов или систем для обеспечения их совместной работы. Это может включать в себя как интеграцию на уровне кода, так и на уровне архитектуры.

Разработка требований к программным модулям

Определение ключевых компонентов: Выделите основные компоненты системы и их функции, а также взаимодействия между ними.

Технические спецификации: Проанализируйте, какие технологии, платформы и инструменты будут использоваться в проекте.

Функциональные требования: Опишите функциональность, которую должны обеспечивать модули, включая сценарии использования и бизнес-правила.

Документирование требований: Оформите требования в виде документа, который будет служить основой для разработки и тестирования

Процесс изменения требований: Установите процесс для управления изменениями в требованиях на протяжении всего жизненного цикла проекта.

Выполнение интеграции модулей

Планирование интеграции: Определите, какие модули будут интегрированы, и составьте план интеграции, включая сроки и ответственных.

Создание тестовой среды: Подготовьте тестовую среду, которая будет использоваться для интеграции модулей. Это может быть отдельный сервер или виртуальная машина.

Установка необходимых инструментов: Убедитесь, что все инструменты и библиотеки, необходимые для работы модулей, установлены и настроены.

Объединение кода: Объедините код различных модулей в единую кодовую базу. Это может включать в себя копирование файлов, настройку зависимостей и конфигурационных файлов.

Функциональное тестирование: Проведите тестирование функциональности системы, чтобы убедиться, что все модули работают как задумано.

Тестирование взаимодействия: Проверьте взаимодействие между модулями, чтобы убедиться, что данные передаются корректно и без ошибок.

Нагрузочное тестирование: Если это необходимо, проведите нагрузочное тестирование для оценки производительности системы под различными условиями.

Отладка: Если возникают ошибки, используйте отладочные инструменты для их выявления и исправления

Подготовка к развертыванию: Убедитесь, что все модули готовы к развертыванию в рабочей среде.

Развертывание системы: Разверните интегрированное программное обеспечение в производственной среде.

Мониторинг после развертывания: После развертывания следите за системой на предмет возможных проблем или ошибок.

Выполнение отладки программного модуля

Пошаговое выполнение: Используйте функции пошагового выполнения для выполнения кода по строкам. Это позволяет внимательно следить за изменениями в состоянии программы.

Просмотр значений переменных: В процессе отладки можно просматривать значения переменных в реальном времени. Это помогает понять, какие данные обрабатываются в данный момент и выявить возможные несоответствия.

Использование выражений: Многие инструменты позволяют вводить выражения для вычисления значений переменных на лету. Это может быть полезно для проверки логики программы.

Стек вызовов: Изучите стек вызовов чтобы понять последовательность вызовов функций и определить, где могла возникнуть ошибка.

Выявление причин ошибок: Используя собранные данные, проанализируйте, почему возникла ошибка. Это может быть связано с неправильной логикой, ошибками в данных или неправильными условиями.

Внесение изменений: После нахождения причины ошибки внесите необходимые изменения в код.

Перезапуск отладки: Запустите программу снова в режиме отладки, чтобы убедиться, что исправления устранили проблему.

Регрессионное тестирование: Проверьте другие части программы, чтобы убедиться, что внесенные изменения не вызвали новых ошибок.

Осуществление разработки тестовых наборов и тестовых сценариев для программного обеспечения

Тестовый набор — это группа тестов, которые проверяют определенные аспекты системы. Для его разработки:

Идентификация функциональных областей: Разделите приложение на модули или компоненты.

Создание матрицы трассировки: Убедитесь, что все требования покрыты тестами. Это поможет избежать пропусков.

Определение критериев успешности: Установите четкие критерии для каждого теста, чтобы знать, что считается успешным результатом.

Написание тестовых сценариев

Тестовые сценарии описывают конкретные шаги для выполнения теста. Они должны быть четкими и понятными:

Структура тестового сценария:

Название: Краткое и понятное название теста.

Предусловия: Условия, которые должны быть выполнены перед началом тестирования.

Шаги: Пошаговое описание действий, которые необходимо выполнить.

Ожидаемый результат: Четкое описание того, что должно произойти после выполнения шагов.

Выполнение тестирования

После разработки тестовых наборов и сценариев переходите к их выполнению:

Запуск тестов: Выполните тесты вручную или с помощью автоматизированных средств.

Документирование результатов: Записывайте результаты выполнения тестов, включая обнаруженные ошибки.

Автоматизация тестирования

Если это целесообразно, рассмотрите возможность автоматизации некоторых тестов:

Выбор инструментов автоматизации: Определите, какие инструменты лучше всего подходят для вашего проекта

Создание скриптов автоматизации: Напишите скрипты для автоматизации повторяющихся тестов.

Инспектирование компонентов программного обеспечения на предмет соответствия стандартам кодирования

Определение стандартов кодирования

Стандарты: Установите четкие стандарты кодирования, которые должны быть соблюдены. Это могут быть как внутренние стандарты вашей команды, так и общепринятые

Документация: Убедитесь, что стандарты документированы и доступны для всех членов команды.

Инструменты для инспекции

Системы контроля версий: Интеграция с системами контроля версий (например, Git) позволяет проводить ревью кода перед его слиянием в основную ветку.

Выполняемая работа

Для работы было необходимо создать репозиторий git

Репозиторий включает в себя

Отчет(Report) в этой папке находиться сам отчет

Задания(Src) тут находиться код который нам дали написать для практики

Документы(Docs) в данной папке расположены все необходимые документы

-Репозиторий

-Интеграция Модулей

-Отладка Модулей

-Стандарт кодировки pep 8

-Тестирование программных модулей

-Техническое задание

Описание стандарта кодирования на языке Python

PEP 8 — это руководство по стилю кода для языка Python, которое помогает разработчикам писать код, который будет читаться и поддерживаться другими программистами. Основные рекомендации PEP 8 включают: PEP 8 — это официальный стиль кодирования для языка Python, который обеспечивает единообразие и читаемость кода. Основные рекомендации PEP 8 включают:

Отступы: используйте 4 пробела для каждого уровня отступа.

Максимальная длина строки: Ограничьте длину строк 79 символами.

Пробелы: используйте пробелы вокруг операторов и после запятых, но не используйте пробелы перед запятыми и скобками.

предоставлять функции для получения обработанных изображений, которые затем могут быть использованы в пользовательском интерфейсе.

Для проведения инспектирования кода, который выполняет изменение расширения файла (конвертацию изображений между форматами PNG и JPG) и перемещение изображений в другие папки, нужно рассмотреть несколько аспектов: читаемость кода, обработка ошибок, структура и функциональность.

Основные аспекты для инспектирования

1. Читаемость кода:

Код должен быть хорошо структурирован и легко читаться.

Используйте понятные имена переменных и функций.

Добавьте комментарии, объясняющие сложные участки кода.

2. Обработка ошибок:

Убедитесь, что код корректно обрабатывает ошибки, такие как отсутствие файла, неверный путь или проблемы с форматом изображения.

Используйте исключения для обработки неожиданных ситуаций.

3. Функциональность:

Проверьте, что конвертация изображений работает корректно и сохраняет файлы в нужном формате.

Убедитесь, что перемещение файлов выполняется правильно.

Для интеграции модулей, которые выполняют изменение расширения файла (конвертацию изображений между форматами PNG и JPG) и перемещение изображений в другие папки, мы можем создать единый файл, который будет объединять функциональность обоих модулей. Это позволит пользователю конвертировать изображения и перемещать их в удобном интерфейсе.

Полная интеграция модулей

Для интеграции модулей, которые выполняют изменения формата изображений (например, PNG в JPG и наоборот) и перемещение изображений в другие папки с использованием API, можно использовать Python с библиотеками, такими как Flask для создания API и Pillow для работы с изображениями. Ниже приведен пример реализации.

Шаги реализации

1. Установка необходимых библиотек:

Убедитесь, что у вас установлены необходимые библиотеки. Вы можете установить их с помощью pip install Flask Pillow

2. Создание API с использованием Flask:

Мы создадим простой API, который будет обрабатывать запросы на изменение формата изображения и перемещение файлов.

Объяснение кода

1. Импорт библиотек: Импортируем необходимые библиотеки для работы с Flask и изображениями.

2. Создание API:

/convert: Этот маршрут принимает POST-запрос с изображением и целевым форматом (jpg или png). Он сохраняет изображение во временной папке, изменяет его формат и возвращает имя нового файла.

/move: Этот маршрут принимает POST-запрос с именем файла и целевой папкой. Он перемещает указанный файл в новую папку.

Для интеграции модулей, которые выполняют изменения формата изображений (например, PNG в JPG и наоборот) и перемещение изображений в другие папки с использованием API, можно использовать Python с библиотеками, такими как Flask для создания API и Pillow для работы с изображениями. Ниже приведен пример реализации.

Модуль взаимодействия с пользователем и формирование и хранений данных api

Шаги реализации

1. Установка необходимых библиотек:

Убедитесь, что у вас установлены необходимые библиотеки. Вы можете установить их с помощью pip install Flask Pillow\

2. Создание API с использованием Flask:

Мы создадим простой API, который будет обрабатывать запросы на изменение формата изображения и перемещение файлов.

3. Реализация функционала:

Объяснение кода

1. Импорт библиотек: Импортируем необходимые библиотеки для работы с Flask и изображениями.

2. Создание API:

/convert: Этот маршрут принимает POST-запрос с изображением и целевым форматом (jpg или png). Он сохраняет изображение во временной папке, изменяет его формат и возвращает имя нового файла.

/move: Этот маршрут принимает POST-запрос с именем файла и целевой папкой. Он перемещает указанный файл в новую папку.

Отладка модулей

Для отладки модулей, отвечающих за изменение расширения файла (конвертацию изображений между PNG и JPG) и перемещение изображений в другие папки, мы можем использовать несколько подходов. Я предложу вам несколько шагов, которые помогут выявить и устранить возможные проблемы в коде.

1. Обновленный код API

2. Объяснение изменений

Конвертация изображений:

В функции convert\_image добавлена проверка для сохранения в нужном формате: если формат jpg, изображение конвертируется в RGB и сохраняется как JPEG; если png, то сохраняется как PNG.

Перемещение изображений:

Добавлен новый маршрут /move, который принимает JSON с image\_path (путь к изображению) и target\_folder (папка назначения).

Проверяет существование файла, создает целевую папку (если она не существует) и перемещает файл с помощью функции shutil.move.

Теперь у вас есть API, который позволяет конвертировать изображения между форматами PNG и JPG, а также перемещать изображения в другие папки. Вы можете дополнительно расширять функциональность по мере необходимости, добавляя дополнительные проверки и обработку ошибок.

Для реализации функциональности, где пользователь задает путь к изображению и тип конвертации (JPG или PNG), нам нужно создать два модуля: один для обработки пользовательского ввода и другой для работы с изображениями.

Модуль работы с изображениями (image\_processor.py)

Этот модуль будет отвечать за конвертацию изображений.

3. Основной файл приложения (app.py)

Этот файл будет отвечать за взаимодействие с пользователем и передачу данных в модуль обработки изображений.

4. Пример работы приложения

После запуска вы увидите приглашение ввести путь к изображению.

Затем введите тип конвертации (например, jpg или png).

Если все прошло успешно, приложение сообщит вам о новом пути к конвертированному изображению.

Тестовый Сценарий

Модуль обработки и работы с изображениями

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Наименование теста | Тестовые данные | Ожидаемый результат | Фактический результат | Результат тестирования | Комментарий |
| Изменение Расширения файла (PNG to JPG, JPG to PNG) | Исходный файл: изображение в формате PNG  Целевой файл: изображение в формате JPG  Размер файла: 1 МБ  Качество изображения: Высокое | Конвертация PNG в JPG:  Файл должен быть успешно преобразован в JPG.  Размер файла должен уменьшиться (обычно на 30-50%).  Качество изображения должно оставаться приемлемым, без значительных потерь.  Конвертация JPG в PNG:  Файл должен быть успешно преобразован в PNG.  Размер файла может увеличиться.  Изображение должно поддерживать прозрачность (если она была в оригинале). | Конвертация PNG в JPG:  Файл был успешно преобразован  Размер файла уменьшился до 450 КБ.  Качество изображения осталось высоким, с минимальными потерями.  Конвертация JPG в PNG:  Файл был успешно преобразован  Размер файла увеличился до 1.2 МБ.  Прозрачность не поддерживалась, так как исходное изображение не имело прозрачных областей. | Конвертация PNG в JPG прошла успешно с ожидаемыми результатами.  Конвертация JPG в PNG также прошла успешно, но без поддержки прозрачности. | Эти инструменты обеспечивают высокое качество конвертации и простоту использования |
| Перемещение изображения в другие папки | Исходное изображение: image1.jpg  Исходная папка: C:\Users\Username\Pictures  Целевая папка: C:\Users\Username\Documents\Images  Ожидаемый результат | Изображение image1.jpg должно быть успешно перемещено из папки Pictures в папку Documents\Images. После перемещения в исходной папке файл должен отсутствовать, а в целевой папке он должен быть доступен. | Файл успешно перемещен в целевую папку, отсутствует в исходной папке | Тест пройден | Процесс перемещения занял 2 секунды |

Пользователь задает путь к изображению, далее указывает тип конвертации (JPG или PNG), модуль передает данные модулю работы с изображением и выдает результат.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Наименование теста | Тестовые данные | Ожидаемый результат | Фактический результат | Результат тестирования | Комментарий |
| Пользователь задает путь к изображению, далее указывает тип конвертации (JPG или PNG), модуль передает данные модулю работы с изображением и выдает результат. | Замените  path/to/your/image.png на реальный путь к вашему изображению и укажите желаемый формат конвертации. | Формат файла: Изображение будет преобразовано в формат PNG.  Качество: Качество изображения должно оставаться высоким, так как PNG использует сжатие без потерь.  Прозрачность: Если исходное изображение имеет прозрачные области, они должны быть сохранены в новом файле PNG. | Качество изображения: Обычно сохраняется высокое качество, однако если JPG был сильно сжат, улучшения могут быть незначительными.  Поддержка прозрачности: PNG поддерживает прозрачные фоны, что является преимуществом по сравнению с JPG, который не поддерживает прозрачность. | Позволяет конвертировать до 50 файлов одновременно.  Использует технологии для создания прозрачных PNG | Удобство интерфейса: Некоторые сервисы предлагают интуитивно понятный интерфейс, что упрощает процесс.  Функциональность: Возможность пакетной обработки и поддержка различных форматов может значительно ускорить работу.  Безопасность: Убедитесь, что сервис удаляет загруженные файлы через определенное время для защиты вашей конфиденциальности. |

Заключение

На этой практике я улучшил свои познания в питоне.

Также мне стал более понятен git репозиторий что расширило мои навыки программирования.

Литература

ГОСТ 7.32-2001 «Отчет о научно-исследовательской работе. Структура

и правила оформления» [Электронный ресурс]/ Электронный правовой и нормативно-технической документации- режим доступа: <http://docs.cntd.ru/document/gost-7-32-2001-sibid>

ГОСТ 7.1-2003 № 332-ст «Система стандартов по информации,

библиотечному и издательскому делу. Библиографическая запись. Библиографическое описание. Общие требования и правила составления» [Электронный ресурс]/ Электронный правовой и нормативно-технической документации- режим доступа: <http://docs.cntd.ru/document/gost-7-1-2003-sibid>

Работы с Pillow <https://python-scripts.com/pillow>

ГОСТ 7.80-2000 «Библиографическая запись. Заголовок. Общие

требования и правила составления» [Электронный ресурс]/ Электронный правовой и нормативно-технической документации- режим доступа: <http://docs.cntd.ru/document/gost-7-80-2000>

ГОСТ Р 7.0.5-2008 «Система стандартов по информации,

библиотечному и издательскому делу. Библиографическая ссылка. Общие требования и правила составления» [Электронный ресурс]/ Библиотека ГОСТов стандартов и нормативов- режим доступа: <http://www.standartov.ru/norma_doc/53/53649/index.htm>

ГОСТ Р 7.0.11-2011 «Система стандартов по информации,

библиотечному и издательскому делу. Диссертация и автореферат диссертации. Структура и правила оформления» [Электронный ресурс]/ Электронный правовой и нормативно-технической документации- режим доступа: <http://docs.cntd.ru/document/gost-r-7-0-11-2011>

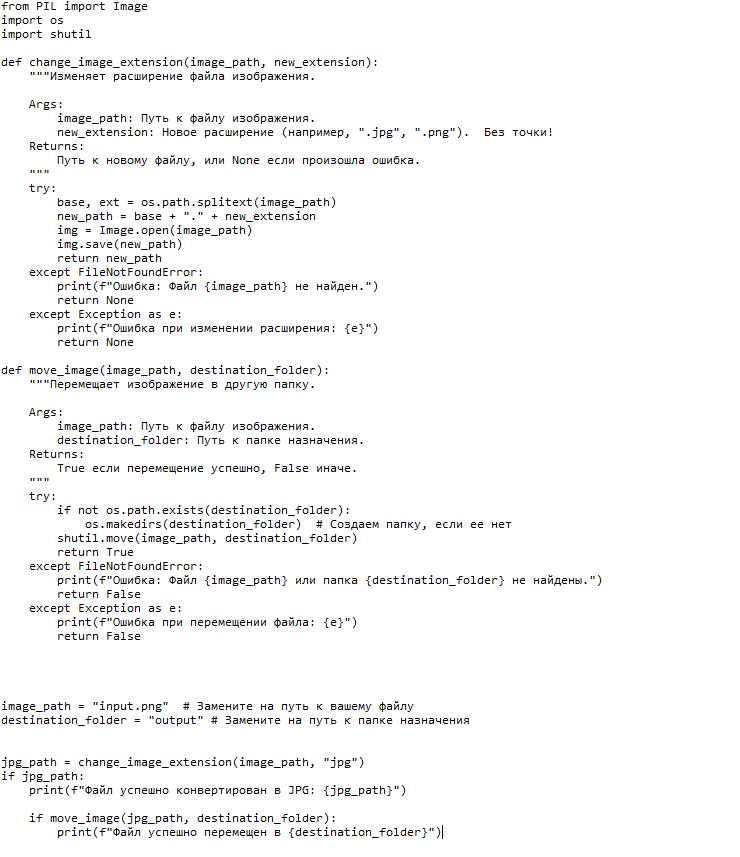
Модули в Python [https://docs.python.org/3/tutorial/modules.html](http://docs.python.org/3/tutorial/modules.html)

Модуль os [https://docs.python.org/3/library/os.html](http://docs.python.org/3/library/os.html)

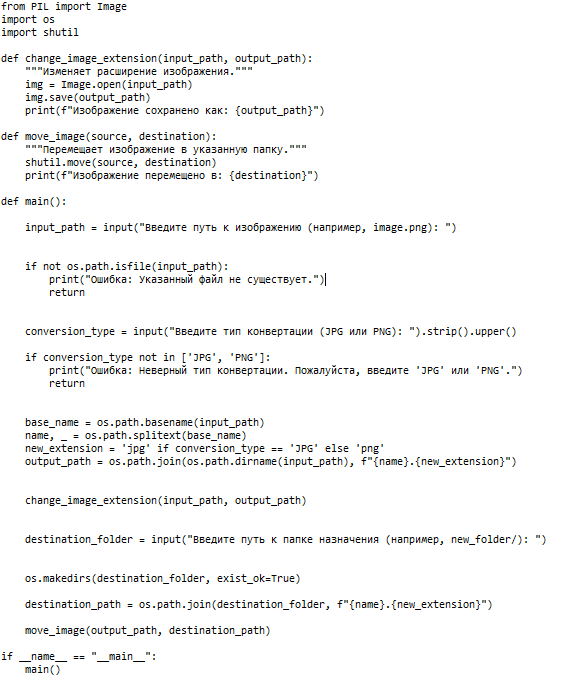
Сайт Компании по практике [https://www.mallenom.ru](http://www.mallenom.ru)

Приложение

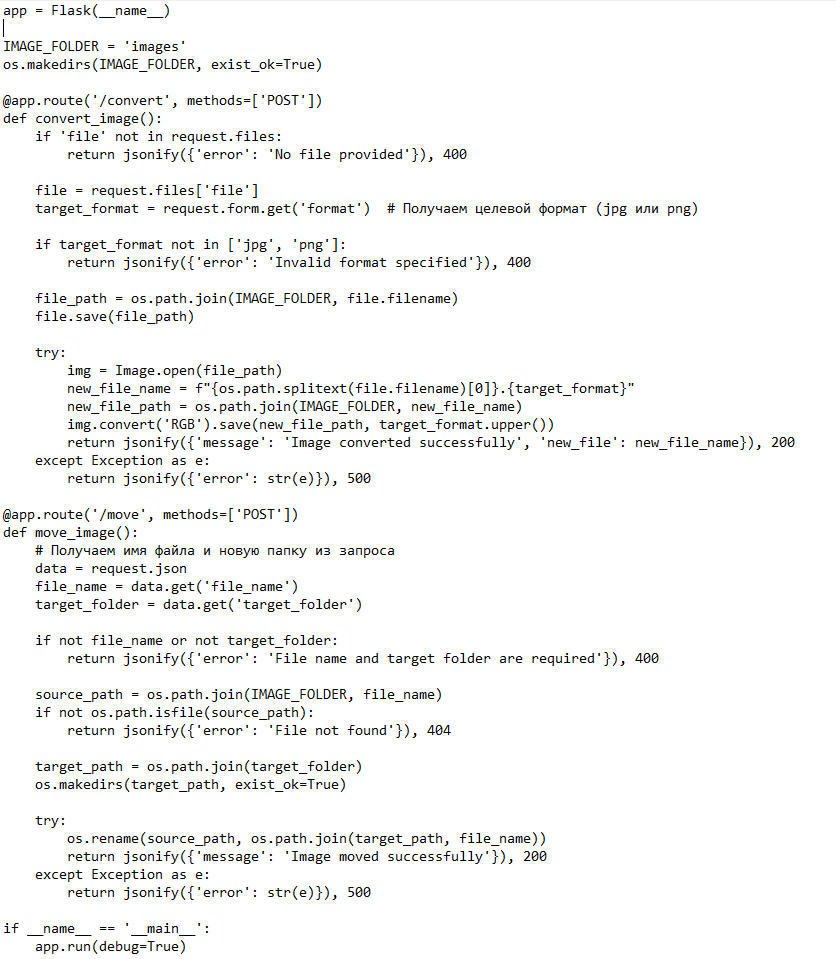
Модуль обработки и работы с изображениями



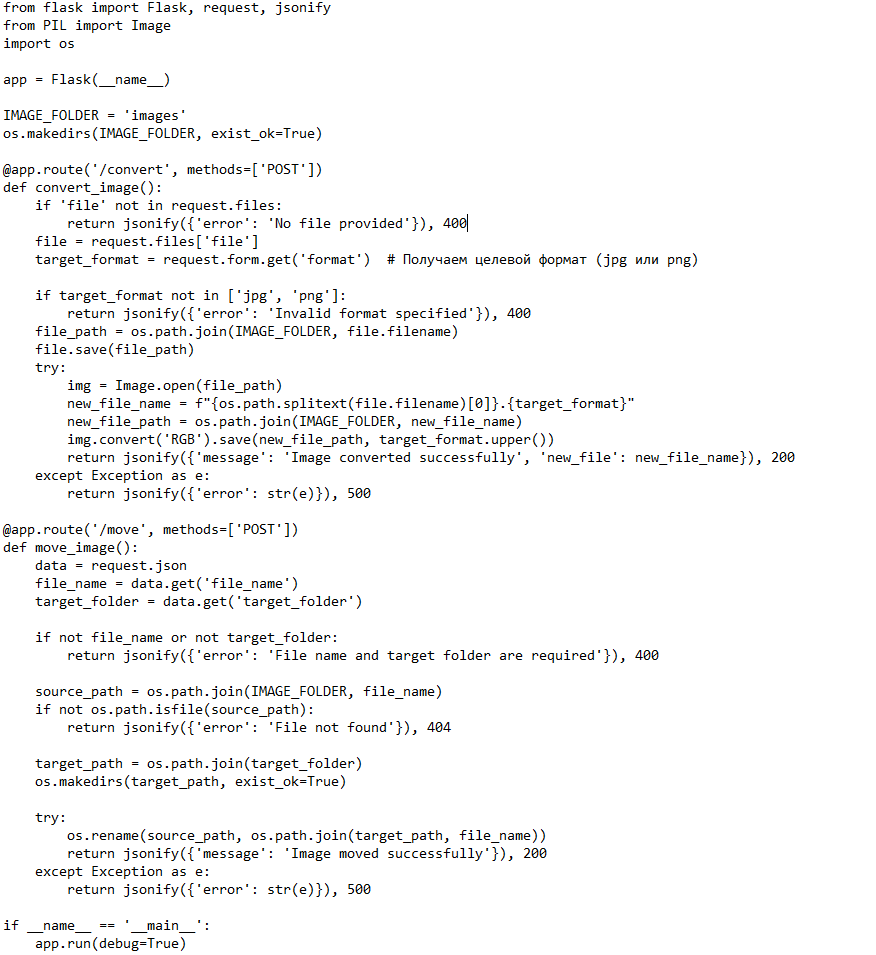
Модуль взаимодействия с пользователем и формирование и хранений данных



Модуль обработки и работы с изображениями API

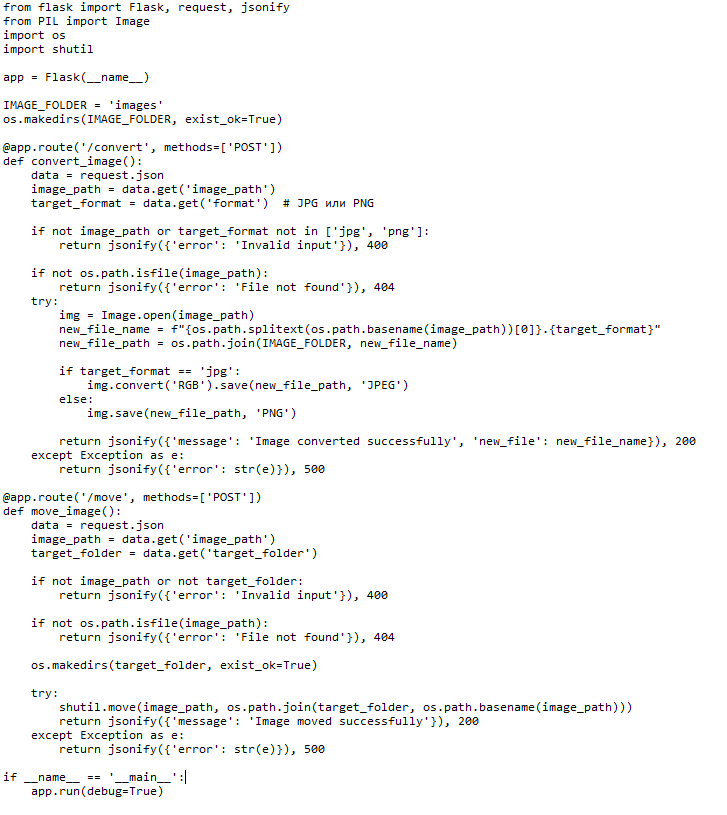


Модуль взаимодействия с пользователем и формирование и хранений данных API

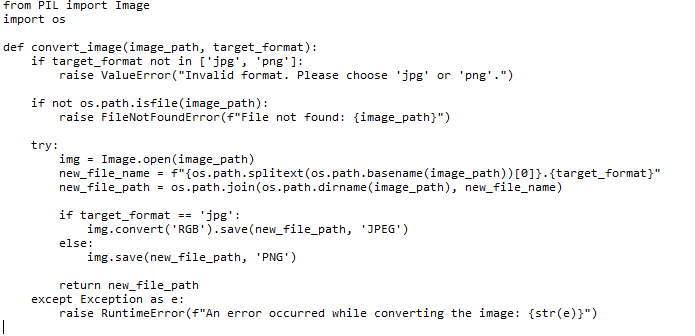


Отладка модулей

Модуль обработки и работы с изображениями API



Модуль работы с изображениями (image\_processor.py)



Основной файл приложения (app.py)

