

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 ПРЕДПРИЯТИЯ (ОРГАНИЗАЦИИ)	5
1.1. Общая информация	5
1.2. Должностные инструкции ИТ-специалистов предприятия	9
2. ОСУЩЕСТВЛЕНИЕ ИНТЕГРАЦИИ ПРОГРАММНЫХ МОДУЛЕЙ	10
2.1. Требования к программным модулям	10
2.2. Выполнение интеграции модулей в программное обеспечение	11
2.3. Выполнение отладки программного модуля с использованием специализированных программных средств	11
2.4. Осуществление разработки тестовых наборов и тестовых сценариев для программного обеспечения	12
2.5. Инспектирование компонентов программного обеспечения на предмет соответствия стандартам кодирования	12
3. ВЫПОЛНЯЕМЫЕ ЗАДАНИЯ	14
3.1. Репозиторий и составление технического задания	14
3.2. Стандарт кодирования на языке Python (PEP 8)	16
3.3. Интеграция модулей друг с другом	18
3.4. Тестирование программных модулей	19
ЗАКЛЮЧЕНИЕ	22
ИСТОЧНИКИ	23
ПРИЛОЖЕНИЯ 1	25

ВВЕДЕНИЕ

Место прохождения производственной практики: ООО "Малленом Системс".

Срок прохождения производственной практики: с 24.11.2024 по 7.12.2024

Специальность: Информационные системы и программирование

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, таких как:

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 В 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.

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.

1. ОБЩАЯ ХАРАКТЕРИСТИКА ПРЕДПРИЯТИЯ (ОРГАНИЗАЦИИ)

1.1. Общая информация

Малленом Системс — это ведущая российская компания, специализирующаяся на разработке и внедрении систем компьютерного зрения и промышленной видео-аналитики, основанных на технологиях машинного зрения и искусственного интеллекта, включая машинное обучение и нейронные сети глубокого обучения.

Компания была основана в 2011 году на базе команды ученых и программистов Санкт-Петербургского политехнического университета Петра Великого при поддержке инвестиционной компании «Малленом».

Основной профиль компании включает реализацию наукоемких IT-проектов в таких сферах, как транспорт, машиностроение, нефтегазовая, металлургическая, пищевая и фармацевтическая промышленности, а также в алмазодобывающей и атомной отраслях.

Продукция Малленом Системс представлена во многих регионах России, странах СНГ и ЕС. Уникальный опыт внедрений и ноу-хау компании позволяют быстро разрабатывать новые высокотехнологичные продукты и решения.

На протяжении десяти лет компания является официальным партнером-интегратором в России и СНГ для Cognex — мирового лидера в области машинного зрения и промышленной идентификации.

С 2022 года Малленом Системс представляет на российском и евразийском рынках компанию Hikrobot — дочернее подразделение Hikvision, занимающееся производством оборудования для машинного зрения и мобильных роботов.

Кроме того, Малленом Системс участвует в национальном рейтинге быстрорастущих технологических компаний России «ТехУспех», разработанном РВК.

Организационная структура предприятия:

1. Высшее руководство

Генеральный директор: является ключевой фигурой в управлении компанией, отвечая за стратегическое направление и общее руководство. Генеральный директор формирует видение и миссию компании, принимает важные решения, касающиеся ее развития и устойчивости на рынке.

Главный технический директор: отвечает за технологическое развитие и внедрение инноваций в компании. Он следит за современными тенденциями в области технологий и обеспечивает их интеграцию в процессы компании, что позволяет поддерживать конкурентоспособность.

Директор по развитию: занимается стратегическим планированием и реализацией инициатив, направленных на рост бизнеса. Его задачи включают анализ новых рыночных возможностей, разработку стратегий для увеличения доли рынка и оптимизацию бизнес-процессов.

2. Отдел тестирования и контроля качества:

Этот отдел отвечает за проверку качества как программного, так и аппаратного обеспечения. Специалисты отдела проводят тестирование на различных этапах разработки, чтобы выявить и устранить возможные дефекты, обеспечивая тем самым высокое качество конечного продукта.

3. Маркетинг и продажи

Отдел маркетинга: занимается анализом рынка, разработкой PR-стратегий и рекламой. Специалисты отдела исследуют потребности клиентов и разрабатывают маркетинговые кампании для продвижения продуктов компании.

Отдел продаж: работает с клиентами, формирует коммерческие предложения и заключает контракты. Его задача — наладить долгосрочные отношения с клиентами и обеспечить выполнение планов по продажам.

4. Инженерные службы

Руководитель направления системной интеграции: отвечает за управление проектами, связанными с интеграцией различных IT-систем и технологий в единую инфраструктуру. Он координирует работу команд и обеспечивает выполнение проектов в срок.

Отдел системной интеграции: занимается интеграцией решений в инфраструктуру клиентов, обеспечивая их совместимость и функциональность.

Отдел технической поддержки: обеспечивает поддержку клиентов, решая технические проблемы и отвечая на запросы пользователей. Это важный элемент для поддержания высокого уровня удовлетворенности клиентов.

5. Отдел разработки ПО

Руководитель отдела разработки: управляет процессами разработки программного обеспечения, включая создание алгоритмов компьютерного зрения и программ для анализа видео. Отдел может быть разделен на более мелкие команды, каждая из которых фокусируется на определенных аспектах разработки, таких как обработка изображений, машинное обучение и создание пользовательских интерфейсов. Это позволяет повысить эффективность работы и качество конечного продукта.

6. Финансовый и административный отдел

Бухгалтерия: ведет финансовую отчетность и управление бюджетом компании. Бухгалтерия обеспечивает прозрачность финансовых потоков и соблюдение законодательства в области финансов.

HR-отдел: занимается подбором, обучением и развитием персонала. HR-отдел играет ключевую роль в формировании корпоративной

7. Юридический отдел

Обеспечивает правовую поддержку компании, занимается контрактами и соблюдением законодательства.

Внутренний распорядок работы предприятия, охрана труда на предприятии (организации)

В компании "Малленом Системс" установлен стандартный график работы, который подразумевает пятидневную рабочую неделю с понедельника по пятницу. Рабочие часы начинаются в 09:00 и заканчиваются в 18:00, что позволяет сотрудникам эффективно организовывать свое время и выполнять поставленные задачи в рамках рабочего дня.

Важным аспектом обеспечения безопасности и здоровья работников является наличие специалиста по охране труда. Этот профессионал играет ключевую роль в создании безопасной рабочей среды. Он проводит вводные инструктажи для новых сотрудников, что позволяет им ознакомиться с правилами безопасности, а также с внутренними регламентами компании. Такие инструктажи особенно важны при приеме на работу, так как они помогают новым работникам понять, как избежать потенциальных рисков и обеспечить свою безопасность на рабочем месте.

Кроме того, специалист по охране труда отвечает за выдачу пропусков для пусконаладочных работ инженеров. Это необходимо для контроля доступа на территорию компании, особенно в тех случаях, когда проводятся работы, требующие повышенного внимания к безопасности.

Стоит отметить, что в компании была проведена специальная оценка условий труда 20 сентября 2018 года. Результаты этой оценки показали, что на рабочих местах отсутствуют вредные производственные факторы. Это свидетельствует о том, что компания соблюдает требования законодательства в области охраны труда и здоровья сотрудников. Таким образом, "Малленом Системс" создает безопасные условия для своих работников, что является важным аспектом корпоративной культуры и социальной ответственности.

В целом, подход компании к охране труда и безопасности сотрудников является примером ответственного отношения к своим работникам и стремления создать комфортную и безопасную рабочую среду.

1.2. Должностные инструкции ИТ-специалистов предприятия

Техник:

Техник выполняет следующие должностные обязанности:

- Выполняет работу по проведению необходимых технических расчетов;
- Осуществляет наладку, настройку, регулировку и опытную проверку оборудования и систем, следит за его исправным состоянием;
- Принимает участие в проведение экспериментов и испытаний;
- Принимает участие в разработке программ, инструкций и другой технической документации, в изготовлении макетов, а также в испытаниях и экспериментальных работах;
- Выполняет работу по сбору, обработке и накоплению исходных материалов, данных статистической отчетности, научно-технической информации;
- Составляет описания проводимых работ, необходимые спецификации, диаграммы, таблицы, графики и другую техническую документацию;
- 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;
- Систематизирует, обрабатывает и подготавливает данные для составления отчетов о работе;
- Принимает необходимые меры по использованию в работе современных технических средств.

2. ОСУЩЕСТВЛЕНИЕ ИНТЕГРАЦИИ ПРОГРАММНЫХ МОДУЛЕЙ

Интеграция программных модулей — это ключевой этап в разработке программного обеспечения, который обеспечивает совместную работу различных компонентов системы. Этот процесс требует тщательного планирования и внимательного выполнения, чтобы гарантировать, что все части системы функционируют как единое целое.

2.1. Требования к программным модулям

Разработка требований к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонентов:

Анализ документации. Начинается с тщательного изучения проектной и технической документации. Это включает в себя архитектурные схемы, описания интерфейсов, спецификации API и любые другие документы, которые могут дать представление о том, как модули должны взаимодействовать.

Выявление зависимостей. Определяется зависимости между модулями. Какие модули требуют данных от других? Какой формат данных будет использоваться при обмене? Эти аспекты критически важны для успешной интеграции.

Согласование функциональных и нефункциональных требований. Определяется не только функциональные требования (например, что модуль должен делать), но и нефункциональные (например, производительность, безопасность, масштабируемость). Это поможет создать более полное представление о том, как модули должны работать вместе.

Документация требований. Создается документ с четко сформулированными требованиями, который будет служить основой для

дальнейшей работы. Это поможет избежать недопонимания между членами команды и обеспечит единый подход к разработке.

2.2. Выполнение интеграции модулей в программное обеспечение

Подготовка среды интеграции. Создается отдельную среду для интеграции, где можно будет тестировать взаимодействие модулей без влияния на основную продуктивную среду. Убедитесь, что все необходимые инструменты и зависимости установлены.

Физическая интеграция. На этом этапе происходит фактическая интеграция модулей в единую систему. Это может включать в себя настройку конфигурационных файлов, подключение баз данных и настройку API.

Проверка корректности взаимодействия. После интеграции необходимо провести тестирование на предмет корректности взаимодействия между модулями. Убедитесь, что данные передаются правильно и что модули реагируют на события так, как это было предусмотрено.

2.3. Выполнение отладки программного модуля с использованием специализированных программных средств

Логирование. Настраивается логирование для отслеживания работы модулей. Логи могут содержать информацию о входных данных, выходных результатах и возникающих ошибках, что значительно упростит процесс отладки.

Инструменты отладки. Используются современные инструменты отладки (например, IDE с встроенными отладчиками, профайлеры) для выявления и устранения ошибок в коде. Эти инструменты могут помочь вам отслеживать выполнение программы и выявлять проблемные участки.

Тестирование. Выполняется отладку как на уровне отдельных модулей (юнит-тестирование), так и на уровне интеграции (интеграционное тестирование). Это поможет выявить ошибки как в логике отдельных компонентов, так и в их взаимодействии.

2.4. Осуществление разработки тестовых наборов и тестовых сценариев для программного обеспечения

Создание тестовых наборов. Определите наборы тестов для проверки каждой функциональности системы. Убедитесь, что тесты покрывают все важные сценарии использования.

Разработка тестовых сценариев. Напишите детализированные сценарии тестирования, которые описывают шаги, необходимые для проверки каждого требования. Сценарии должны включать как позитивные тесты (проверка правильного поведения системы), так и негативные (проверка обработки ошибок).

Автоматизация. Рассмотрите возможность автоматизации тестирования с использованием инструментов для автоматизированного тестирования. Это позволит значительно ускорить процесс тестирования и повысить его эффективность.

2.5. Инспектирование компонентов программного обеспечения на предмет соответствия стандартам кодирования

Определение стандартов кодирования. Установите четкие стандарты кодирования для команды (например, использование определенного стиля написания кода, правил именования переменных и функций). Это поможет обеспечить единообразие и читаемость кода.

Проведение ревью кода. Организуйте регулярные проверки кода (код-ревью) среди членов команды для выявления несоответствий стандартам.

Это не только помогает улучшить качество кода, но и способствует обмену знаниями внутри команды.

Использование статического анализа кода. Применяйте инструменты статического анализа кода для автоматической проверки на соответствие стандартам. Эти инструменты могут выявлять потенциальные проблемы еще до запуска программы.

3. ВЫПОЛНЯЕМЫЕ ЗАДАНИЯ

Список выполненных заданий во время прохождения практики необходимых для заполнения отчёта:

- Репозиторий.
- Техническое задание на разработку модулей.
- Стандарт кодирования PEP 8.
- Инспектирование разрабатываемого программного кода.
- Интеграция Модулей.
- Отладка модулей.
- Тестирование программных модулей

3.1. Репозиторий и составление технического задания

В ходе производственной практики был создан репозиторий в Github для предоставления отчёта, документов практики и всех выполненных заданий. Содержит следующую структуру:

1) Report. В данной папке расположен отчёт по всей производственной практике.

2) Src. В данной папке расположен код разрабатываемых модулей.

3) Docs. Содержит все документы практики.

Техническое задание на разработку модулей

Заказчик: ООО «Малленом Системс»,

Выполняющий: Студент Группы ИС-23 Ткаченко Тихон Александрович

Введение:

Данное техническое задание (ТЗ) описывает требования по практике к разработке двух модулей: модуля обработки и работы с изображениями и модуля взаимодействия с пользователем. Модули предназначены для обработки изображений, включая изменение их размера и склеивание.

Основания:

Разработка модулей вызвана запросом от компании в которой студент проходит практику. В рамках данного проекта мы стремимся обучиться и построить интуитивно понятный инструмент для работы с изображениями в рамках знаний обучающегося студента.

Цели и задачи:

Цель разработки заключается в создании двух модулей:

1. Модуль обработки и работы с изображениями, который будет выполнять следующие функции:

- Изменение размера изображения.
- Склеивание нескольких изображений в одно.

2. Модуль взаимодействия с пользователем и формирования данных, который будет:

- Обеспечивать ввод данных от пользователя.
- Передавать данные в модуль обработки изображений.
- Выдавать результат, сохраняя объединенное изображение в указанной директории.

Технологические требования:

- Язык программирования: Python.
- Используемые библиотеки: Pillow (для работы с изображениями).
- Формат выходных изображений: JPEG, PNG (возможны другие).

Pillow – это, бесплатная библиотека с открытым исходным кодом для работы с изображениями и их обработки на языке Python.

Общие Требования к модулям:

1. Модуль обработки и работы с изображениями:

- Возможность изменения размера изображения с указанием новых параметров (ширина и высота).
- Возможность склеивания двух или более изображений в одно с сохранением качества.

2. Модуль взаимодействия с пользователем:

- Пользователь вводит путь к двум изображениям.
- Передача путей к изображениям в модуль обработки для выполнения операций (изменение размера, склеивание).
- Уведомление пользователя о результате выполнения операций (успешное сохранение объединенного изображения).
- Тестирование:
 - Проведение тестирования каждой функции модуля обработки изображений индивидуально.
 - Тестирование модуля взаимодействия с пользователем по различным сценариям ввода.
 - Все функции функционируют в соответствии с установленными требованиями.
 - Отсутствие критических ошибок и сбоев.
 - Обновленная документация к коду.

Сроки выполнения: срок производственной практики

Предварительные стадии разработки:

1. Сбор информации
2. Анализ задания
3. Подготовка приложения для разработки
4. Проектирование системы
5. Тестирование

3.2. Стандарт кодирования на языке Python (PEP 8)

PEP 8 — это стиль кодирования для языка Python, который служит набором рекомендаций и стандартов для написания читаемого и поддерживаемого кода. Он был разработан для того, чтобы помочь программистам следовать единым принципам, что, в свою очередь, облегчает совместную работу над проектами и улучшает качество кода. Следование

PEP 8 позволяет создавать код, который будет понятен не только вам, но и другим разработчикам, которые могут работать с вашим кодом в будущем. Основные принципы PEP представлены ниже.

Максимальная длина строки. Рекомендуется ограничивать длину строк до 79 символов. Это помогает избежать горизонтальной прокрутки в редакторах и делает код более читаемым. Для комментариев и документации рекомендуется использовать строки длиной до 72 символов.

Именованние переменных и функций. Имена переменных должны быть понятными и описательными. Для переменных и функций рекомендуется использовать стиль `snake_case` (например, `my_variable`, `calculate_sum`). Для классов применяется стиль `CamelCase` (например, `MyClass`).

Импорт модулей. Импортируйте модули в начале файла, группируя их по стандартным библиотекам, сторонним библиотекам и вашим собственным модулям. Каждый импорт должен находиться на отдельной строке.

Пробелы в выражениях и инструкциях. PEP 8 рекомендует использовать пробелы вокруг операторов и после запятых, чтобы улучшить читаемость. Избегайте излишнего использования пробелов, например перед запятой или после открывающей скобки.

Докстринги и комментарии. Документация к функциям и классам должна быть написана в виде строк документации (docstrings). Они должны начинаться с краткого описания и содержать информацию о параметрах и возвращаемых значениях.

Пробелы между функциями и классами. Используйте два пустых ряда между определениями классов и функциями, а один пустой ряд между методами внутри класса. Это помогает структурировать код и улучшает его читаемость.

Отступы. В PEP 8 рекомендуется использовать 4 пробела для каждого уровня отступа. Это делает структуру кода более очевидной и облегчает его восприятие. Избегайте использования табуляции, так как это может привести к несоответствию в отображении кода в разных редакторах.

Инспектирование модулей (Модуль обработки и работы с изображениями и Модуль взаимодействия с пользователем и формирование и хранения данных)

1. Импорт модулей. Модули следует импортировать в начале файла, перед определением функций.

2. Использование `if __name__ == "__main__"`. При импорте кода провести обязательную проверку на двойные подчеркивания

3. Форматирование. Применяется следующее:

– Рекомендуется использовать 4 пробела для отступов.

– В комментариях перед строками кода следует добавлять два пробела перед текстом комментария.

4. Длина строки. Старайтесь придерживаться максимальной длины строки в 79 символов. Если ваши строки длиннее, разбивайте их на несколько строк.

5. Докстринги. Докстринги функций содержат описание, но можно добавить больше информации о параметрах и возвращаемых значениях.

6. Код после вызова функций. Вызовы функций в блоке `if __name__ == "__main__"`: должны быть оформлены более аккуратно, чтобы было легче читать и поддерживать.

7. По возможности добавить более конкретные сообщения об ошибках для разных типов исключений, чтобы пользователь мог легче понять, что произошло.

3.3. Интеграция модулей друг с другом

Метод Импорт и использование функций:

Для интеграции модулей обработки изображений и взаимодействия с пользователем в одном проекте с использованием импорта, мы можем

организовать код в несколько файлов. Один файл будет содержать функции для обработки изображений. Затем мы объединим его в основном файле.

1. Модуль `image_utils.py` (Приложение 1). Содержит функции `size_image` для изменения размера изображений и `join_images` для склеивания нескольких изображений.

2. Модуль `main.py` (Рис.4). Содержит следующие функции:

- Запрашивает у пользователя пути к двум изображениям.
- Проверяет существование файлов.
- Вызывает функцию `size_image` для изменения размера каждого изображения.
- Вызывает функцию `join_images` для склеивания измененных изображений и сохранения результата.

3.4. Тестирование программных модулей

При отладке были проверены 2 модуля (Модуль обработки и работы с изображениями и Модуль взаимодействия с пользователем и формирование и хранения данных)

1. Проверка корректности ввода:

- Пользователь вводит корректные пути к изображениям.
- Проверку на существование файлов перед их обработкой.

2. Обработка ошибок:

Функции корректно обрабатывают исключения. Например, если файл изображения не существует, программа должна информировать пользователя об этом.

3. Тестирование функций:

Для 1 модуля (Модуль обработки и работы с изображениями) было переданы существующий и несуществующий путь.

Для 2 модуля (Модуль взаимодействия с пользователем и формирование и хранения данных) проверена работа с двумя существующими изображениями и с одним несуществующим.

Ход тестирования представлен в таблице 1.

Таблица 1 - Результаты тестирований программных модулей.

Наименование теста	Тестовые данные	Ожидаемый результат	Фактический результат	Результат тестирования	Комментарий
Тест 1 Модуль обработки и работы с изображениями	изображения: image1 image2 Ширина 800 Высота 450 В директории C:\Users\Username (изображения не в указанной директории)	Изменение размера изображения и склеивание его с другим изображением	Ошибка: Файл image1.jpg не найден. Ошибка: Файл не найден: [Errno 2] No such file or directory:	Код работает с ошибкой поскольку пользователь не импортировал исходные изображения в нужную директорию	Для функционирования кода пользователь должен импортировать изображения в нужную директорию
Тест 2 Модуль обработки и работы с изображениями	изображения: image1 image2 Ширина 800 Высота 450 В директории C:\Users\Username (изображения в указанной директории)	Изменение размера изображения и склеивание его с другим изображением	Изображение image1.jpg изменено и сохранено как image1_resized.jpg Изображения склеены и сохранены как joined_image.jpg	Код работает без ошибок	-
Наименование теста	Тестовые данные	Ожидаемый результат	Фактический результат	Результат тестирования	Комментарий
Тест 3 Модуль взаимодействия с пользователем и формирование и хранения данных	изображения: image1 image2 Пользователь указал неправильные пути к изображениям	Изображения сохранены и склеены по пути.	Один или оба файла не найдены.	Код работает с ошибкой поскольку Пользователь указал неправильные пути к изображениям	Для функционирования кода пользователь должен указать правильное расположение изображений
Тест 4 Модуль взаимодействия с	изображения: image1 image2 Пользователь	Изображения сохранены и склеены по пути	Изображения склеены и сохранены как c:\Users\image1_joi	Код работает без ошибок	-

пользовател ем и формирован ие и хранений данных	указал правильные пути к изображениям		ned.jpg Склеенное изображение сохранено по пути: с:\Users\image1_joi ned.jpg		
---	--	--	---	--	--

ЗАКЛЮЧЕНИЕ

Во время производственной практики я научился эффективно работать с Python и библиотекой Pillow для обработки изображений, что значительно расширило мои навыки программирования. Я освоил основные функции, такие как изменение размера изображений и их склеивание, что позволило мне глубже понять принципы работы с графическими данными.

ИСТОЧНИКИ

- 1) 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>
- 2) 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>
- 3) ГОСТ 7.1-2003 № 332-ст «Система стандартов по информации, библиотечному и издательскому делу. Библиографическая запись. Библиографическое описание.Электронный правовой и нормативно-технической документации- режим доступа: <http://docs.cntd.ru/document/gost-7-1-2003-sibid>
- 4) 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
- 5) 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>
- 6) Сайт Компании ООО «Малленом Системс». Официальный сайт Python [Электронный ресурс] - режим доступа: -<https://www.mallenom.ru>
- 7) Работы с Pillow. Официальный сайт Python [Электронный ресурс] - режим доступа: - <https://python-scripts.com/pillow>

- 8) Модули в Python. Официальный сайт Python [Электронный ресурс] - режим доступа: - <https://docs.python.org/3/tutorial/modules.html>
- 9) Модуль os. Официальный сайт Python [Электронный ресурс] - режим доступа: - <https://docs.python.org/3/library/os.html>
- 10) Стандарт PEP 8. Официальный сайт Python [Электронный ресурс] - режим доступа: - <https://peps.python.org/pep-0008/>
- 11) Отладка. Официальный сайт Хабр [Электронный ресурс] - режим доступа: - <https://habr.com/ru/companies/vk/articles/205426/>

ПРИЛОЖЕНИЕ 1

```
from PIL import Image
import os

def resize_image(input_path, output_path, new_width, new_height):

    try:
        img = Image.open(input_path)
        img = img.resize((new_width, new_height), Image.LANCZOS) # LANCZOS для высокого качества
        img.save(output_path)
        print(f"Изображение {input_path} изменено и сохранено как {output_path}")
    except FileNotFoundError:
        print(f"Ошибка: Файл {input_path} не найден.")
    except Exception as e:
        print(f"Произошла ошибка при изменении размера: {e}")

def join_images(image_paths, output_path):

    try:
        images = [Image.open(path) for path in image_paths]
        widths, heights = zip(*(i.size for i in images))

        total_width = sum(widths)
        max_height = max(heights)

        new_im = Image.new('RGB', (total_width, max_height))

        x_offset = 0
        for im in images:
            new_im.paste(im, (x_offset, 0))
            x_offset += im.size[0]

        new_im.save(output_path)
        print(f"Изображения склеены и сохранены как {output_path}")

    except FileNotFoundError as e:
        print(f"Ошибка: Файл не найден: {e}")
    except Exception as e:
        print(f"Произошла ошибка при склеивании: {e}")

if __name__ == "__main__":
    resize_image("image1.jpg", "image1_resized.jpg", 200, 150)
    image_paths = ["image1_resized.jpg", "image2.jpg"]
    output_path = "joined_image.jpg"
    join_images(image_paths, output_path)
```

Рисунок 1 - Модуль обработки и работы с изображениями

```

from PIL import Image, UnidentifiedImageError
import os

def join_images(image_paths):
    try:
        images = [Image.open(path) for path in image_paths]
        widths, heights = zip(*(i.size for i in images))

        total_width = sum(widths)
        max_height = max(heights)

        new_im = Image.new('RGB', (total_width, max_height))

        x_offset = 0
        for im in images:
            new_im.paste(im, (x_offset, 0))
            x_offset += im.size[0]

        base_name1 = os.path.splitext(os.path.basename(image_paths[0]))[0]
        directory = os.path.dirname(image_paths[0])
        output_path = os.path.join(directory, f"{base_name1}_joined.jpg")
        new_im.save(output_path)
        print(f"Изображения склеены и сохранены как {output_path}")
        return output_path

    except FileNotFoundError as e:
        print(f"Ошибка: Файл не найден: {e}")
        return None
    except UnidentifiedImageError:
        print("Ошибка: Не удалось открыть один из файлов как изображение.")
        return None
    except Exception as e:
        print(f"Произошла непредвиденная ошибка: {e}")
        return None

def get_image_paths_from_user():
    while True:
        try:
            path1 = input("Введите полный путь к первому изображению: ")
            path2 = input("Введите полный путь ко второму изображению: ")
            if not (os.path.exists(path1) and os.path.exists(path2)):
                raise FileNotFoundError("Один или оба файла не найдены.")
            return [path1, path2]
        except FileNotFoundError as e:
            print(e)
        except Exception as e:
            print(f"Ошибка ввода: {e}")
if __name__ == "__main__":
    image_paths = get_image_paths_from_user()
    if image_paths:
        result_path = join_images(image_paths)
        if result_path:
            print(f"Склеенное изображение сохранено по пути: {result_path}")

```

Рисунок 2 - Модуль взаимодействия с пользователем и формирование и хранения данных

```

# image_utils.py
from PIL import Image

def size_image(input_path, output_path, new_width, new_height):
    """Изменяет размер изображения и сохраняет его в указанном пути."""
    try:
        img = Image.open(input_path)
        img = img.resize((new_width, new_height), Image.LANCZOS)
        img.save(output_path)
        print(f"Изображение {input_path} изменено и сохранено как {output_path}")
    except FileNotFoundError:
        print(f"Ошибка: Файл {input_path} не найден.")
    except Exception as e:
        print(f"Произошла ошибка при изменении размера: {e}")

def join_images(image_paths, output_path):
    """Склеивает несколько изображений в одно и сохраняет его в указанном пути."""
    try:
        images = [Image.open(path) for path in image_paths]
        widths, heights = zip(*(i.size for i in images))

        total_width = sum(widths)
        max_height = max(heights)

        new_im = Image.new('RGB', (total_width, max_height))

        x_offset = 0
        for im in images:
            new_im.paste(im, (x_offset, 0))
            x_offset += im.size[0]

        new_im.save(output_path)
        print(f"Изображения склеены и сохранены как {output_path}")

    except FileNotFoundError as e:
        print(f"Ошибка: Файл не найден: {e}")
    except Exception as e:
        print(f"Произошла ошибка при склеивании: {e}")

```

Рисунок 3 - Модуль image_utils.py

```

# main.py
import os
from image_utils import size_image, join_images

def get_image_paths_from_user():
    """Запрашивает у пользователя пути к изображениям."""
    while True:
        try:
            path1 = input("Введите полный путь к первому изображению: ")
            path2 = input("Введите полный путь ко второму изображению: ")
            if not (os.path.exists(path1) and os.path.exists(path2)):
                raise FileNotFoundError("Один или оба файла не найдены.")
            return [path1, path2]
        except FileNotFoundError as e:
            print(e)
        except Exception as e:
            print(f"Ошибка ввода: {e}")

if __name__ == "__main__":
    image_paths = get_image_paths_from_user()

    if image_paths:
        # Изменяем размер изображений перед склеиванием
        resized_paths = []
        for i, path in enumerate(image_paths):
            base_name = os.path.splitext(os.path.basename(path))[0]
            directory = os.path.dirname(path)
            resized_path = os.path.join(directory, f"{base_name}_resized.jpg")
            size_image(path, resized_path, 800, 600) # Изменяем размер
            resized_paths.append(resized_path)

        # Склеивание изображений
        base_name1 = os.path.splitext(os.path.basename(resized_paths[0]))[0]
        output_path = os.path.join(os.path.dirname(resized_paths[0]), f"{base_name1}_joined.jpg")

        join_images(resized_paths, output_path)

```

Рисунок 4 - Главный модуль main.py