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

Период прохождения:

с «\_\_\_» \_\_\_\_\_\_\_ 2024 г.

по «\_\_\_» \_\_\_\_\_\_\_ 2024 г.

Руководитель практики от

предприятия

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

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

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

Руководитель практики от

техникума: Материкова А.А.

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

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

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

г. Череповец

2024

Содержание

[Введение 3](#_Toc183087903)

[1. Общая характеристика предприятия 4](#_Toc183087904)

[2. Осуществление интеграции программных модулей 5](#_Toc183087905)

[3. Выполняемые задания 7](#_Toc183087906)

[Заключение 16](#_Toc183087907)

[Список использованных источников 17](#_Toc183087908)

[Приложения 18](#_Toc183087909)

# Введение

Место прохождения производственной практики – ООО «Малленом Системс»

Сроки прохождения производственного практики – с 10.11.24 по 23.11.24

Цель производственной практики: достижение повышенного уровня подготовки студента в сфере деятельности, знакомстве с особенностями профессии в реальных условиях производства, приобретении и закреплении практических навыков, а также развитии профессиональных данных и ориентации в профессии.

Задачи производственной практики:

Закрепление теоретических знаний, полученных в процессе обучения.

Получение практического опыта в рамках выбранной специальности.

Ознакомление с особенностями работы и внутренней деятельностью профильных предприятий.

Усвоение должностных обязанностей по специальности.

Овладение знаниями нормативов по основным направлениям производственной деятельности.

# 1. Общая характеристика предприятия

1.1. Организационная структура предприятия

Генеральный директор – Живиця Анна Эдуардовна

Учредители – 10 физ. лиц

Сотрудники – более 100 физ. лиц

Внутренний распорядок работы предприятия, охрана труда на предприятии

Общие положения

1.2. Внутренний распорядок работы предприятия ООО «Малленом Системс» регулирует порядок приема и увольнения работников, основные права и обязанности сторон трудового договора, режим работы, время отдыха, меры поощрения и взыскания, а также другие вопросы регулирования трудовых отношений.

Режим работы

Пн-пт 9:00-18:00

Охрана труда

Обязанности работодателя:

Обеспечение безопасности работников при эксплуатации зданий, сооружений, оборудования, инструментов, сырья и материалов.

Создание и функционирование системы управления охраной труда.

Применение сертифицированных средств индивидуальной и коллективной защиты.

Обеспечение соответствующих условий труда на каждом рабочем месте.

1.3. Должностные инструкции ИТ-специалистов предприятия

Должностные инструкции ИТ-специалистов предприятия ООО «Малленом Системс» включают организацию и обслуживание информационной инфраструктуры, решение технических проблем пользователей, обновление программного и аппаратного обеспечения, а также обеспечение безопасности информации.

# 2. Осуществление интеграции программных модулей

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

- Анализ проектной документации: Изучение архитектуры системы, функциональных требований и технологии, используемой в проекте.

- Определение требований: Сформулирование четких требований к каждому модулю с учетом их взаимодействия, данных интерфейсов и ожидаемой функциональности.

- Учет совместимости: Проведение анализа совместимости между различными модулями и их зависимостями.

- Документация требований: Оформление требований в виде документа, который будет служить основой для разработки и тестирования.

2.2. Выполнение интеграции модулей в программное обеспечение

- Интеграция на уровне исходного кода: Сборка и компиляция кода модулей, настройка необходимых зависимостей и библиотек.

- Настройка окружения: Обеспечение необходимой инфраструктуры для работы модулей (установка серверов, баз данных и т.д.).

- Финальная сборка программы: Компиляция и связывание всех модулей в единое приложение.

- Первичная проверка работоспособности: Проведение тестов на совместимость интегрированных модулей, выявление и устранение ошибок на начальных этапах.

2.3. Выполнение отладки программного модуля с использованием специализированных программных средств

- Использование отладчиков: Применение инструментов для анализа и диагностики работы модулей (например, GDB, Visual Studio Debugger).

- Логирование и профилирование: Настройка логирования для получения данных о работе модуля и его производительности.

- Поиск и исправление ошибок: Идентификация и устранение проблем, возникающих при выполнении модулей, включая неочевидные ошибки и утечки памяти.

- Тестирование на крайних значениях: Проверка работы модулей с нештатными входными данными и сценариями.

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

- Определение типов тестов: Выбор и разработка юнит-тестов, интеграционных тестов и системных тестов.

- Создание тестовых наборов: Формирование набора тестовых данных, необходимых для проверки функциональности модулей.

- Документация тестов: Оформление тестовых случаев (test cases) и сценариев (test scenarios) с описанием ожидаемых результатов.

- Автоматизация тестирования: При необходимости, разработка автоматизированных тестов для повышения эффективности проверки.

2.5. Инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования

- Проверка стандартов кодирования: Сравнение написанного кода с установленными стандартами кодирования (PEP8 для Python).

- Проведение код-ревью: Оценка качества кода другими разработчиками для выявления потенциальных проблем и улучшения читаемости.

- Использование статического анализа: Применение инструментов для статического анализа кода, которые автоматически находят нарушения стандартов.

- Подготовка отчета о соответствии: Составление отчета о проведенных инспекциях с выводами и рекомендациями по улучшению качества кода.

# 3. Выполняемые задания

Для начала работы был создан GIT-репозиторий для производственной практики с необходимой структурой.

Структура репозитория:

- Отчет

- Задания

- Документы

Ссылка на репозиторий:

Иванов Артём Вячеславович, ссылка на репозиторий - https://github.com/Wensday1337/Ivanov

Добавление отчета и документов на практику:

- В репозиторий добавлены:

- Отчет о выполненной практике, включающий подробное описание первого задания.

- Документы, связанные с практикой.

В рамках производственной практики были разработаны два модуля для обработки изображений и взаимодействия с пользователем, а также техническое задание. Заказчиком выступила компания ООО «Малленом Системс». Данный отчет содержит описание выполненных заданий, а также основные моменты, связанные с проектированием, реализацией и тестированием модулей.

Техническое задание на разработку модулей

Данное техническое задание описывает требования к разработке двух модулей на языке Python, предназначенных для обработки изображений и взаимодействия с пользователем с учетом потребностей заказчика ООО «Малленом Системс».

Участники проекта

- Заказчик: ООО «Малленом Системс»

- Разработчик: Иванов Артём

- Тестировщик: Иванов Артём

Модуль обработки и работы с изображениями

Название модуля: image\_processing.py

Описание

Модуль предназначен для выполнения следующих операций над изображениями:

- Изменение размера изображения.

- Поворот изображения.

Функциональные требования:

- Изменение размера изображения

- Поворот изображения

Нефункциональные требования:

- Использование библиотеки Pillow для работы с изображениями.

- Обработка возможных ошибок (например, отсутствующие файлы, неправильные форматы).

Модуль взаимодействия с пользователем и формирования и хранения данных

Название модуля: user\_interface.py

Описание

Модуль отвечает за сбор данных от пользователя и передачу их в модуль обработки изображений. Включает функции для проверки введённых данных.

Функциональные требования:

Получение пользовательского ввода:

- Запрашивать у пользователя путь к изображению.

- Запрашивать у пользователя новые ширину и высоту изображения.

- Запрашивать у пользователя угол поворота изображения.

- Проверять корректность введённого пути к изображению.

Вызов функций обработки изображений:

- Передавать собранные данные в функции модуля image\_processing.py.

- Выводить сообщения об успехе выполнения операций.

Нефункциональные требования:

- Простой и интуитивно понятный интерфейс.

- Обработка исключений, связанных с некорректными входными данными.

Технические требования

- Основной язык программирования: Python

- Библиотеки:

- Pillow для работы с изображениями.

- Операционная система: независима от платформы (Windows, macOS, Linux).

- Документация: комментарии в коде, описания функций, инструкции по запуску.

Тестирование и приемка

Способы тестирования

- Тестирование каждой функции модуля обработки изображений отдельно.

- Тестирование модуля взаимодействия с пользователем на различные сценарии ввода.

Критерии приемки

- Все функции работают согласно требованиям.

- Нет критических ошибок и сбоев.

- Актуальная документация к коду.

Заказчик: ООО «Малленом Системс»

Дата реализации: Срок производственной практики

Разработка модулей

1. Модуль обработки и работы с изображениями

Модуль был назван image\_processing.py и включает в себя следующие функции:

1.1 Изменение размера изображения:

- Функция принимает путь к входному изображению, путь для сохранения измененного изображения, а также новые размеры (ширину и высоту).

- Используется библиотека Pillow для выполнения данной операции.

1.2 Поворот изображения:

- Функция принимает путь к входному изображению, путь для сохранения повёрнутого изображения и угол поворота.

- Также реализована обработка ошибок для случаев, когда файл не найден или имеет неподдерживаемый формат.

Пример реализации функции изменения размера (Рис.1)

2. Модуль взаимодействия с пользователем

Модуль был назван user\_interface.py и обеспечивает сбор пользовательского ввода и взаимодействие с модулем обработки изображений.

Основные функции модуля:

2.1 Получение пользовательского ввода:

- Модуль запрашивает у пользователя путь к изображению, а также новые размеры и угол поворота.

- Реализована проверка корректности введенных данных, необходимо отображать ошибки, если данные введены неправильно.

2.2 Вызов функций обработки изображений:

- После получения данных от пользователя модуль вызывает соответствующие функции обработки изображений и выводит сообщения об успешной операции или ошибках при выполнении.

Пример реализации функции для получения ввода от пользователя (Рис.2)

Далее код был проанализирован в соответствии со стандартом кодирования PEP 8 для Python

PEP 8 — это стиль кодирования для языка Python, который помогает поддерживать единообразие и читаемость кода. Ниже приводятся основные рекомендации PEP 8:

Отступы:

- Используйте 4 пробела на уровень отступа. Не используйте табуляцию.

Максимальная длина строки:

- Строки кода не должны превышать 79 символов. Для комментариев и документации — 72 символа.

Пропуски:

- Разделяйте функции и классы с двумя пустыми строками.

- Разделяйте методы внутри классов одной пустой строкой.

- Используйте один пробел перед и после операторов (например, a = b + c).

Импортируемые модули:

- Импортируйте стандартные библиотеки, сторонние библиотеки и локальные модули в следующем порядке, каждая группа должна быть разделена пустой строкой.

Именование:

- Используйте snake\_case для имен переменных и функций.

- Используйте CamelCase для имен классов.

- Константы должны именоваться UPPER\_SNAKE\_CASE.

Комментирование:

- Комментарии должны быть ясными и объяснять "почему" код делает что-то, а не "что" он делает.

- Однострочные комментарии должны начинаться с заглавной буквы и быть отделены от кода двумя пробелами.

- Многострочные комментарии должны быть отделены пустыми строками.

Строки документации (docstrings):

- Используйте строки документации для описания функций и классов. Они должны быть в тройных кавычках и пояснять назначение функции, ее параметры и возвращаемые значения.

Сравнение с None:

- Для сравнения с None используйте is и is not, а не == или != (например, if obj is None:).

Использование пробелов в выражениях:

- Не добавляйте пробелы перед запятыми, точками с запятой или двоеточиями (например, func(a, b) вместо func(a , b)).

Тернарные операторы:

- Используйте явные конструкции if-else для многострочных условий. Тенарные операторы должны использоваться аккуратно.

Инспектирование модуля image\_processing.py по стандарту PEP 8:

1. Отступы: В функции отсутствует правильное форматирование отступов (4 пробела). Следует исправить отступы для комментариев и кода.

2. Пробелы: Необходимо добавлять два пробела перед комментариями для лучшей читаемости.

3. Кодовые строки: Рекомендуется использовать длинные строки комментариев и кода с учетом максимальной длины строки в 79 символов.

4. Импорт: Импортируйте модули в первую очередь перед определением функций.

5. Исключения: Рекомендуется использовать более специфичные исключения вместо общего `Exception`.

Исправленный код (Рис.3)

Инспектирование модуля user\_interface.py по стандарту PEP 8:

1. Отступы: Обратите внимание на правильное использование отступов (4 пробела).

2. Комментарии: Для более ясного выделения комментариев используйте пробелы.

3. Импорт: Импортируйте модули в первой части файла.

4. Проверка имени: Убедитесь, что используете двойные подчеркивания в if \_\_name\_\_ == "\_\_main\_\_":.

Исправленный код (Рис. 4)

Общие рекомендации:

- Старайтесь поддерживать однородность в стиле кода во всех модулях.

- Убедитесь, что все функции и классы имеют соответствующие комментарии и документацию.

- Не забывайте тестировать код после внесения изменений.

Интеграция модулей друг с другом предполагает создание связей и взаимодействия между ними для обеспечения корректной работы всей системы. В моём коде коде интеграцию создает импортирование модулей (Рис. 5)

А также взаимодействие через функции: Когда пользователь вводит данные (путь к изображению, размеры и угол поворота), вы передаёте эти данные в функции resize\_image и rotate\_image. (Рис. 6)

При откладке код не выдал ошибок

Проведение тестирования программных модулей

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Наименование  теста | Тестовые  данные | Ожидаемый  результат | Фактический  результат | Результат  тестирования | Комментарий |
| Test1 | Ширина: 900  Длина: 300  Угол: 90 | Создание 2 изображений, где одно повернуто на 90 градусов, а второе изменило свое разрешение | Создание 2 изображений, где одно повернуто на 90 градусов, а второе изменило свое разрешение | Код работает отлично, без ошибок, но принимает только .png | - |
| Test2 | Ширина: 1920  Длина: 200  Угол: 1 | Создание 2 изображений, где одно повернуто на 1 градус, а второе изменило свое разрешение | Создание 2 изображений, где одно повернуто на 1 градус, а второе изменило свое разрешение |

# Заключение

При прохождении производственной практики я углубил свои знания в области написания программного кода на языке Python, создал 2 модуля, которые взаимодействуют друг с другом - модуль обработки и работы с изображениями и модуль взаимодействия с пользователем и формирования и хранения данных

# Список использованных источников

1. Стандарт PEP 8 - <https://peps.python.org/pep-0008/>

2. Отладка - <https://habr.com/ru/companies/vk/articles/205426/>

3. Работа с модулями - <https://metanit.com/python/tutorial/2.10.php>

# Приложения

Рисунок 1

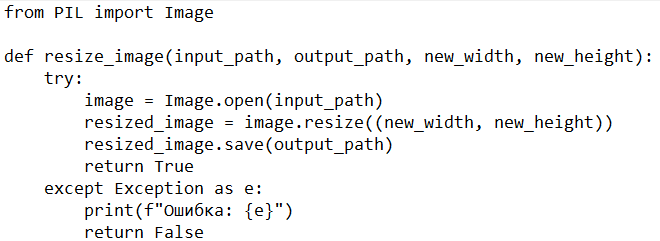


Рисунок 2

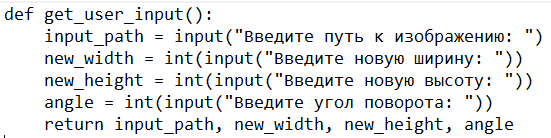


Рисунок 3

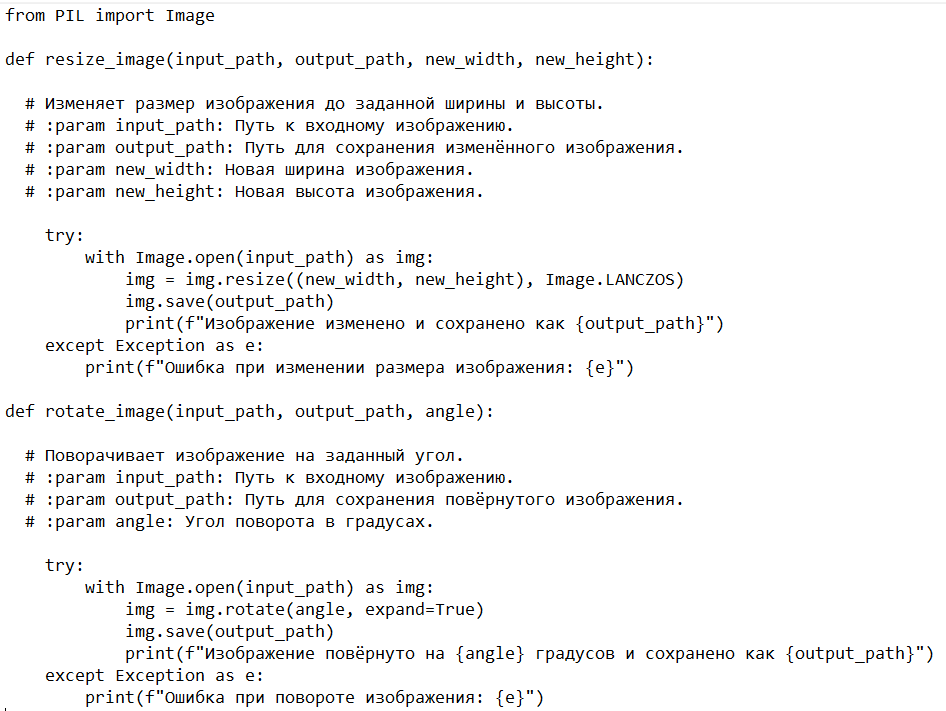


Рисунок 4

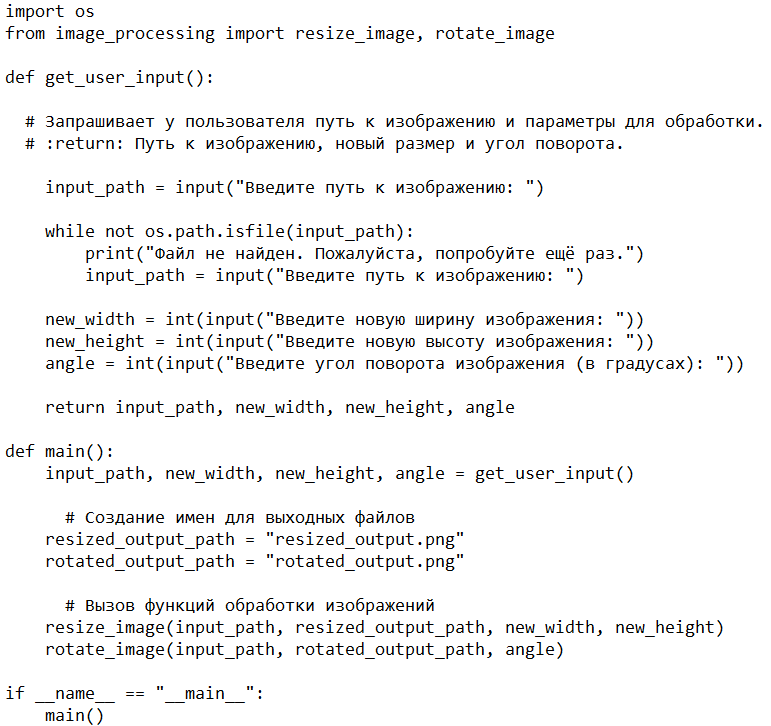


Рисунок 5



Рисунок 6

