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 «\_\_\_» \_\_\_\_\_\_\_ 2025 г.

по «\_\_\_» \_\_\_\_\_\_\_ 2025 г.

Руководитель практики от

предприятия

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

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

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

МПРуководитель практики от

техникума: Материкова А.А.

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

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

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2025 года

г. Череповец

2025

Содержание

[ВВЕДЕНИЕ 2](#_Toc211351932)

[1 ОБЩАЯ ХАРАКТЕРИСТИКА ПРЕДПРИЯТИЯ 3](#_Toc211351933)

[1.1 Организационная структура предприятия 3](#_Toc211351934)

[1.2 Внутренний распорядок работы предприятия, охрана труда на предприятии 3](#_Toc211351935)

[1.3 Должностные инструкции ИТ-специалистов предприятия 4](#_Toc211351936)

[2 ОСУЩЕСТВЛЕНИЕ ИНТЕГРАЦИИ ПРОГРАММНЫХ МОДУЛЕЙ 4](#_Toc211351937)

[2.1 Разработка требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент 4](#_Toc211351938)

[2.2 Выполнение интеграции модулей в программное обеспечение 4](#_Toc211351939)

[2.3 Выполнение отладки программного модуля с использованием специализированных программных средств 5](#_Toc211351940)

[2.4 Осуществление разработки тестовых наборов и тестовых сценариев для программного обеспечения 6](#_Toc211351941)

[2.5 Инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования 6](#_Toc211351942)

[3 ВЫПОЛНЯЕМЫЕ ЗАДАНИЯ 7](#_Toc211351943)

[ЗАКЛЮЧЕНИЕ 12](#_Toc211351944)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 13](#_Toc211351945)

[ПРИЛОЖЕНИЯ 14](#_Toc211351946)

# ВВЕДЕНИЕ

Производственная практика проходила в ООО «Малленом Системс» в отделе разработки программного обеспечения с 06 октября 2025 года по 19 октября 2025 года.

Цели и задачи практики:

1. Разрабатывать требования к программным модулям на основе проектной и технической документации на предмет взаимодействия компонент
2. Выполнять интеграцию модулей в программное обеспечение
3. Выполнять отладку программного модуля с использованием специализированных программных средств
4. Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения
5. Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования

В ходе практики были разработаны программные модули для обработки изображений, включающие функции изменения размера и поворота графических файлов, а также модуль взаимодействия с пользователем для ввода параметров обработки и вывода результатов.

# 1 ОБЩАЯ ХАРАКТЕРИСТИКА ПРЕДПРИЯТИЯ

## Организационная структура предприятия

Руководство компании:

Генеральный директор: Живиця Анна Эдуардовна

Учредители: 10 физических лиц

Численность сотрудников: более 100 человек

## Внутренний распорядок работы предприятия, охрана труда на предприятии

Общие положения

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:

Понедельник-пятница: с 9:00 до 18:00

Обеденный перерыв: с 13:00 до 14:00

Технические перерывы: каждые 2 часа по 10 минут

Охрана труда

Использование сертифицированного оборудования

Регулярные перерывы для снижения нагрузки на зрение

Соблюдение правильной осанки при работе за компьютером

Соблюдение противопожарной безопасности

## 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.

# 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, 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.

В процессе отладки использовался отладчик PyCharm для:

— Пошагового выполнения функций load\_image(), resize\_image(), rotate\_image()

— Анализа значений переменных при обработке изображений

— Выявления и исправления ошибок обработки исключений

— Проверки корректности передачи данных между модулями

Были обнаружены и исправлены следующие проблемы:

— Некорректная обработка путей к файлам с кириллическими символами

— Отсутствие проверки существования папки "Результаты" перед сохранением

## 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

— В процессе разработки проводилось инспектирование исходного кода на соответствие стандарту PEP 8. Проверка включала:

Анализ именования:

— Все переменные и функции используют snake\_case (например, image\_path, get\_user\_input)

Проверка отступов:

— Использованы стандартные отступы в 4 пробела

Длина строк:

— Строки кода не превышают 79 символов

Импорты:

— Модули импортируются в начале файлов, каждый на отдельной строке

Пробелы вокруг операторов:

— Добавлены пробелы для улучшения читаемости

# ВЫПОЛНЯЕМЫЕ ЗАДАНИЯ

Создание рабочей среды и репозитория

Для выполнения работ по производственной практике был создан Git-репозиторий с очищенной структурой, содержащей только необходимые для проекта файлы.

Структура репозитория:

* Docs/
* Report/
* SRC/

-- image\_processor.py

-- user\_interface.py

-- main.py

- README.md

Ссылка на репозиторий:

Новоселов Алексей Алексеевич —<https://github.com/SayYo48/proizvodstvennaya-praktika>

Разработка модуля обработки изображений (image\_processor.py)

— Модуль реализует три основные функции для работы с изображениями:

Функция load\_image (image\_path)

Назначение:

— Загрузка изображения из файла

Логика работы:

— Использует метод Image.open из библиотеки Pillow. При возникновении ошибок (файл не найден, повреждённый формат) возвращает None и выводит сообщение об ошибке.

Функция resize\_image (image, new\_size)

Назначение:

— Изменение размера изображения

Логика работы:

— Принимает объект изображения и кортеж с новыми размерами (width, height). Использует метод image.resize. Обрабатывает исключения при некорректных параметрах.

Функция rotate\_image (image, angle)

Назначение:

— Поворот изображения на заданный угол

Логика работы:

— Использует метод image.rotate с указанием угла поворота. Обеспечивает обработку ошибок выполнения операции.

Разработка модуля пользовательского интерфейса (user\_interface.py)

— Модуль обеспечивает взаимодействие с пользователем и включает функции:

Функция get\_user\_input

Назначение:

— Получение входных данных от пользователя

Логика работы:

— Последовательно запрашивает путь к изображению, новые размеры и угол поворота. Выполняет базовое преобразование типов данных.

Функция save\_and\_show\_result (image, original\_path)

Назначение:

— Сохранение и отображение результатов обработки

Логика работы:

— Создает папку "Результаты" при её отсутствии, генерирует имя для выходного файла, сохраняет изображение и выводит информацию о выполненной операции.

Разработка основного модуля (main.py)

— Модуль осуществляет интеграцию всех компонентов системы:

Логика работы main.py:

— Вывод приветственного сообщения

— Получение входных данных через user\_interface\get\_user\_input

— Загрузка изображения через image\_processor\load\_image

— Последовательное применение операций изменения размера и поворота

— Сохранение и отображение результатов через user\_interface\save\_and\_show\_result

— Вывод сообщения о завершении обработки

Интеграция модулей выполнена через импорт функций в основной файл main.py:

— Модуль user\_interface.py передает данные в модуль image\_processor.py

— Модуль main.py координирует взаимодействие между всеми компонентами

— Обработка ошибок обеспечивает стабильную работу при некорректных входных данных

— Создана единая точка входа в приложение через main.py

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Наименование теста | Тестовые данные | Ожидаемый результат | Фактический результат | Результат тестирования | Комментарий |
| Test1 | Ширина: 800 Высота: 600 Угол: 45° | Создание обработанного изображения | Создание изображения Преорбазованный\_image.jpg | Код работает без ошибок | — |
| Test2 | Ширина: 1920 Высота: 1080 Угол: 90° | Создание полноразмерного повернутого изображения | Создание изображения с поворотом на 90° |
| Test3 | Ширина: 100 Высота: 100 Угол: 180° | Создание маленького перевернутого изображения | Корректное создание уменьшенного изображения |
| Test4 | Несуществующий путь к файлу | Сообщение об ошибке | "Ошибка: файл не найден" |
| Test5 | Ширина: 0 Высота: 0 Угол: 0° | Сообщение об ошибке | "Ошибка при изменении размера" |

# 

# ЗАКЛЮЧЕНИЕ

В период с 06 октября по 19 октября 2025 года в ООО «Малленом Системс» мной были успешно освоены профессиональные компетенции по интеграции программных модулей. Разработаны требования к модулям системы обработки изображений. Выполнена интеграция трех программных модулей: обработки изображений, пользовательского интерфейса и главного управляющего модуля. Проведена отладка программы с устранением ошибок обработки изображений. Созданы тестовые сценарии проверки функциональности.

Практические результаты:

— Разработана система обработки изображений на Python

— Освоены инструменты Git, VSCode, библиотека Pillow

— Создана модульная архитектура приложения

— Приобретены навыки промышленной разработки ПО

Практика позволила закрепить теоретические знания и получить практический опыт, необходимый для профессиональной деятельности в сфере разработки программного обеспечения.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Техническое задание на производственную практику ПП.02 «Осуществление интеграции программных модулей»
2. Официальная документация Python 3.13 [Электронный ресурс]. — режим доступа: <https://docs.python.org/3/>
3. Документация библиотеки Pillow [Электронный ресурс]. — режим доступа: <https://pillow.readthedocs.io/>
4. Стандарт PEP 8 - Style Guide for Python Code [Электронный ресурс]. —режим доступа: <https://peps.python.org/pep-0008/>
5. Система контроля версий Git [Электронный ресурс]. — режим доступа: <https://git-scm.com/>
6. ГОСТ 7.32-2001 «Отчет о научно-исследовательской работе. Структура и правила оформления»
7. Visual Studio Code Documentation [Электронный ресурс]. — режим доступа: <https://code.visualstudio.com/docs>
8. Методические рекомендации по разработке программной документации

# ПРИЛОЖЕНИЯ

Рисунок 1 — Главный модуль программы (main.py)

Изображение выглядит как текст, снимок экрана, Шрифт, число

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 2 — Модуль взаимодействия с пользователем (user\_interface.py)



Рисунок 3 — Модуль обработки изображений (image\_processor.py)

Изображение выглядит как текст, снимок экрана, Шрифт, документ

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 4 — Принцип работы программы

Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.