

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

Введение	3
Введение	10
Основания	10
Цели и задачи	10
Технологические требования:	10
Общие Требования к модулям:	11
Тестирование	11
Сроки выполнения	11
Предварительные стадии разработки	11
Модуль обработки и работы с изображениями	11
Условия использования:.....	12
Модуль взаимодействия с пользователем и формирование и хранения данных	12
Заключение	13
Источники	13
Приложения	15

Введение

Задачи:

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

Цели:

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 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;

1

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 компании

Малленом Системс — это одна из ведущих российских организаций, сосредоточенная на создании и внедрении систем компьютерного зрения и промышленной видеоаналитики, использующих технологии машинного зрения и искусственного интеллекта, включая машинное обучение и глубокие нейронные сети.

Компания была основана в 2011 году группой ученых и программистов СанктПетербургского политехнического университета Петра Великого при поддержке инвестиционной компании «Малленом».

Основная деятельность компании включает реализацию высокотехнологичных IT-проектов в таких отраслях, как транспорт, машиностроение, нефтегазовая, металлургическая, пищевая и фармацевтическая промышленности, а также в алмазодобывающем и атомном секторах.

Продукция Малленом Системс доступна в различных регионах России, странах СНГ и ЕС. Уникальный опыт и ноу-хау компании позволяют быстро разрабатывать новые высокотехнологичные решения.

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

С 2022 года Малленом Системс представляет на российском и евразийском рынках компанию Hikrobot — дочернюю организацию Hikvision, которая занимается производством оборудования для машинного зрения и мобильных роботов.

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

| 1.1 Организационная структура компании

1. Высшее руководство

- **Генеральный директор:** Является центральной фигурой в управлении компанией, отвечая за стратегическое направление и общее руководство. Генеральный директор формирует видение и миссию организации, а также принимает ключевые решения, касающиеся ее развития и устойчивости на рынке.
- **Главный технический директор:** Отвечает за технологическое развитие и внедрение инноваций. Он следит за современными трендами в технологии и обеспечивает их интеграцию в процессы компании для поддержания конкурентоспособности.
- **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 программного и аппаратного обеспечения. Специалисты проводят тестирование на различных этапах разработки, выявляя и устраняя возможные дефекты для обеспечения высокого качества конечного продукта.

3. Инженерные службы

- Руководитель направления системной интеграции: Отвечает за управление проектами по интеграции различных IT-систем и технологий в единую инфраструктуру, координируя работу команд и обеспечивая 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, что является важным аспектом для поддержания высокого уровня удовлетворенности клиентов.

4. Отдел разработки ПО

- Руководитель отдела разработки: Управляет процессами создания программного обеспечения, включая разработку алгоритмов.

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

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 пуска наладочных 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.

1.3 Должностные инструкции ИТ-специалистов предприятия

1. Общие положения

1.1. Настоящая должностная инструкция определяет должностные обязанности, права и ответственность Техника Общества с ограниченной ответственностью «Малленом Системс» (далее – Техник, Общество).

1.2. Техник относится к категории специалистов.

1.3. Техник принимается на работу и увольняется приказом генерального директора или уполномоченным им лицом.

1.4. На должность Техника назначается лицо, без предъявления требований к образованию и опыту работы.

1.5. Техник подчиняется непосредственно руководителю структурного подразделения, ведущему программисту и/или руководителю направления/проекта, в котором работает в настоящее время.

1.6. Техник должен знать:

- методы автоматической и автоматизированной проверки работоспособности программного обеспечения;
- основные виды диагностических данных и способы их представления; – языки, утилиты и среды программирования, и средства пакетного выполнения процедур;
- типовые метрики программного обеспечения;
- основные методы измерения и оценки характеристик программного обеспечения; – методы создания и документирования контрольных примеров и тестовых наборов данных;
- правила, алгоритмы и технологии создания тестовых наборов данных;
- требования к структуре и форматам хранения тестовых наборов данных;
- методы и средства проверки работоспособности программного обеспечения;
- среду проверки работоспособности и отладки программного обеспечения;
- внутренние нормативные документы, регламентирующие порядок документирования результатов проверки работоспособности программного обеспечения;
- методы и средства рефакторинга и оптимизации программного кода;
- языки программирования и среды разработки;
- внутренние нормативные документы, регламентирующие требования к программному коду, порядок отражения изменений в системе контроля версий; – внутренние нормативные документы, регламентирующие порядок отражения результатов рефакторинга и оптимизации в коллективной базе знаний;
- методы и приемы отладки программного кода;
- типовые ошибки, возникающие при разработке программного обеспечения, и методы их диагностики и исправления;
- локально-нормативные акты Общества, касающиеся выполнения его должностных обязанностей;
- требования охраны труда и правила пожарной безопасности.

1.4. Техник должен знать и уметь:

- писать программный код процедур проверки работоспособности программного обеспечения на выбранном языке программирования под руководством наставника;
- использовать выбранную среду программирования для разработки процедур проверки работоспособности программного обеспечения на выбранном языке программирования;

- применять методы и средства проверки работоспособности программного обеспечения;
- анализировать значения полученных характеристик программного обеспечения; – документировать результаты проверки работоспособности программного обеспечения;
- применять методы, средства для рефакторинга и оптимизации;
- публиковать результаты рефакторинга и оптимизации в коллективной базе знаний в виде лучших практик;
- использовать систему контроля версий для регистрации произведенных изменений;
- применять методы и приемы отладки дефектного программного кода;
- интерпретировать сообщения об ошибках, предупреждения, записи технологических журналов, возникающих при выполнении дефектного кода.

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.

Техническое задание на разработку модулей.

Заказчик:

«Малленом Системс»

Выполняющий:

Студент Группы ИС-23 Куликов Сергей Артёмович

Введение

Это техническое задание (ТЗ) определяет требования к разработке двух модулей для обработки изображений: модуля обработки изображений (изменение размера, склеивание) и модуля взаимодействия с пользователем.

Основания

Разработка модулей обусловлена заданием от компании, где проходит практику студент. Цель проекта — приобретение студентом практических навыков программирования и создание интуитивно понятного инструмента для обработки изображений, учитывающего его текущий уровень знаний.

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).

Форматы изображений: смотреть от условия.

Общие Требования к модулям:

1.Модуль обработки и работы с изображениями:

- Поддерживается изменение размера изображений с заданием ширины и высоты, а также склеивание нескольких изображений в одно без потери качества.

2.Модуль взаимодействия с пользователем:

- Пользователь указывает пути к двум изображениям. Система передает эти пути модулю обработки для изменения размера и склеивания. Пользователь получает уведомление об успешном сохранении результирующего изображения.

Тестирование

- Юнит-тестирование функций модуля обработки изображений.
- Тестирование модуля взаимодействия с пользователем с различными входными данными.
- Верификацию соответствия всех функций установленным требованиям.
- Проверку на отсутствие критических ошибок и сбоев.
- Обновление документации

Сроки выполнения

Срок производной практики.

Предварительные стадии разработки

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

Модуль обработки и работы с изображениями

Как работает код:

- 1) `resize_image`: Изменяет размер изображения, используя интерполяцию LANCZOS для лучшего качества. Вы можете указать другие методы интерполяции из документации Pillow.

- 2) `join_images`: Склеивает изображения горизонтально. Он вычисляет общую ширину и максимальную высоту, чтобы создать новое изображение нужного размера, а затем вставляет исходные изображения.

Условия использования:

- 1) Установка Pillow: `pip install Pillow`
- 2) Подготовка изображения: Создайте файлы `image1.jpg` и `image2.jpg` (или замените на ваши имена файлов) в той же директории, где находится ваш скрипт Python.

Модуль взаимодействия с пользователем и формирование и хранения данных Код запрашивает у пользователя пути к двум изображениям, проверяет их существование, склеивает их горизонтально и сохраняет результат в ту же директорию, что и первое изображение. Обработка исключений делает код более устойчивым к ошибкам. Имя выходного файла формируется таким образом, чтобы избежать конфликтов с именами исходных файлов.

Результаты тестирования программных модулей

Название модуля(примеры)	Вводимые данные пользователем	Ожидаемый результат	Конечный результат
1) Неудачный результат Модуль обработки и работы с изображениями	Ширина: 444 Высота: 111 Изображение не в указанной директории	Изменения размера изображения и поворот изображения	Ошибка: Файл <code>image.jpg</code> не найден Ошибка: Файл не найден: [Errno 2] 2 such file or directory
2) Удачный результат Модуль обработки и работы с изображениями	Ширина 800 Высота 450 Изображение есть в указанной директории	Изменение размера изображения и поворот изображения	Изображение <code>image1.jpg</code> изменено и сохранено как <code>image1_resized.jpg</code> Изображение повернуто и

			сохранены как joined_image.jpg
--	--	--	-----------------------------------

Модуль взаимодействия с пользователем и формирование и хранения данных

Название модуля	Вводные данные пользователем	Ожидаемый Результат	Конечный результат
1) Неудачный результат Модуль взаимодействия с пользователем и формирование, и хранения данных	Пользователь указал Неправильные пути к изображениям	Изображения сохранены и повернуты	Один или оба файла не найдены
2) Удачный результат Модуль взаимодействия с пользователем и формирование, и хранения данных	Пользователь указал правильные пути к изображениям	Изображения сохранены и повернуты	Изображения повернуты и сохранены как c:\Users\image1_joined.jpg Повернутое изображение сохранено по пути: c:\Users\image1_joined.jpg

Заключение

Производственная практика позволила мне существенно расширить знания Python, результатом чего стало создание двух взаимодействующих модулей: модуля обработки изображений и модуля взаимодействия с пользователем.

Источники

Сайт Компании по практике <https://www.mallenom.ru>

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 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>

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>

Модуль os <https://docs.python.org/3/library/os.html>

Приложения

1) Модуль обработки и работы с изображениям

```
1 from PIL import Image
2
3 class ImageProcessor:
4     def __init__(self, image_path):
5         try:
6             self.image = Image.open(image_path)
7         except FileNotFoundError:
8             raise FileNotFoundError(f"Image not found at path: {image_path}")
9         except Exception as e:
10            raise Exception(f"Error opening image: {e}")
11
12
13     def resize_image(self, width, height):
14         """Изменяет размер изображения. Сохраняет пропорции, если один из параметров равен -1."""
15         if width == -1:
16             width = int(self.image.width * (height / self.image.height))
17         elif height == -1:
18             height = int(self.image.height * (width / self.image.width))
19         self.image = self.image.resize((width, height))
20
21
22     def rotate_image(self, degrees):
23         """Поворачивает изображение на заданный угол."""
24         self.image = self.image.rotate(degrees)
25
26
27     def save_image(self, output_path):
28         """Сохраняет обработанное изображение."""
29         try:
30             self.image.save(output_path)
31             print(f"Image saved to {output_path}")
32         except Exception as e:
33             raise Exception(f"Error saving image: {e}")
34
35
36 # Пример использования:
37
38 try:
39     processor = ImageProcessor("input.jpg") # Замените "input.jpg" на путь к вашему изображению
40
41     processor.resize_image(400, -1) # Изменяем ширину на 400 пикселей, высота масштабируется пропорционально
42     processor.rotate_image(90) # Поворачиваем на 90 градусов
43
44     processor.save_image("output.jpg") # Сохраняем обработанное изображение
45
46 except FileNotFoundError as e:
47     print(f"Ошибка: {e}")
48 except Exception as e:
49     print(f"Ошибка: {e}")
```

2) Модуль взаимодействия с пользователем и формирование и хранения данных

```
1 from PIL import Image
2 import argparse
3
4 class ImageProcessor:
5     # ... (код класса ImageProcessor из предыдущего примера) ...
6
7
8 def main():
9     parser = argparse.ArgumentParser(description="Resize and rotate images.")
10    parser.add_argument("input_path", help="Path to the input image")
11    parser.add_argument("width", type=int, help="New width of the image (-1 to keep aspect ratio)")
12    parser.add_argument("height", type=int, help="New height of the image (-1 to keep aspect ratio)")
13    parser.add_argument("output_path", help="Path to save the output image")
14    args = parser.parse_args()
15
16    try:
17        processor = ImageProcessor('input_path', 'output_path')
18        processor.resize_image(args.width, args.height)
19        processor.save_image(args.output_path)
20
21    except FileNotFoundError as e:
22        print(f"Error: {e}")
23    except Exception as e:
24        print(f"An error occurred: {e}")
25
26
27 if __name__ == "__main__":
28     main()
```