

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.03 РЕВЬЮИРОВАНИЕ ПРОГРАММНЫХ МОДУЛЕЙ

Выполнил студент 3 курса группы ИС-31

Шевель Милена Александровна

подпись _____

место практики: ООО «Малленом Системс»

Период прохождения:

с «08» декабря 2024 года

по «21» декабря 2024 года

Руководитель практики от

техникума: Материкова А.А.

Оценка: _____

«__» _____ 2024 года

Руководитель практики от

предприятия, должность:

Южакова Н.В., специалист по кадрам

подпись _____

МП

г. Череповец

2024

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
1 ОБЩАЯ ХАРАКТЕРИСТИКА КОМПАНИИ.....	4
1.1 Организационная структура компании	4
1.2 Внутренний распорядок работы компании и охрана труда	5
1.3 Должностные инструкции ИТ-специалиста компании	5
2 РЕВЬЮИРОВАНИЕ ПРОГРАММНЫХ ПРОДУКТОВ.....	6
2.1 Ревьюирование программного кода в соответствии с технической документацией.....	6
2.2 Измерение характеристик компонентов программного продукта	7
2.3 Исследование созданного программного кода с использованием специализированных программных средств	7
2.4 Сравнительный анализ программных продуктов и средств разработки	8
3 ВЫПОЛНЯЕМЫЕ ЗАДАНИЯ.....	9
ЗАКЛЮЧЕНИЕ	17
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ	18
ПРИЛОЖЕНИЯ.....	19

ВВЕДЕНИЕ

Производственная практика является важным этапом в образовательном процессе студентов, позволяющим соединить теоретические знания с реальными производственными процессами. Она предоставляет уникальную возможность получить представление о специфике работы в выбранной области, освоить новые технологии и методики, а также развить профессиональные компетенции. Практика проходила в ООО «Малленом Системс» с 08.12.2024 по 21.12.2024.

Целью практики было освоение основного вида деятельности по направлению «Ревьюирование программных модулей» и формирование соответствующих общих и профессиональных компетенций.

В рамках выполнения поставленной цели, были сформулированы следующие задачи:

1. Осуществить ревьюирование программного кода в соответствии с технической документацией;
2. Выполнить измерение характеристик компонентов программного продукта;
3. Произвести исследование созданного программного кода с использованием специализированных программных средств;
4. Провести сравнительный анализ программных продуктов и средств разработки.

В ходе практики были изучены основные этапы разработки программного обеспечения в компании, а также применены полученные ранее знания в реальных условиях производственного процесса.

1 ОБЩАЯ ХАРАКТЕРИСТИКА КОМПАНИИ

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

Малленом Системс была создана в 2011 году на базе команды ученых и программистов Санкт-Петербургского политехнического университета Петра Великого. Сегодня в компании более 100 сотрудников. Глубокие компетенции в сфере машинного зрения и большой опыт успешной реализации проектов на промышленных предприятиях позволяет успешно решать большой спектр задач в различных отраслях. В Центре исследований и разработки интеллектуальных систем ведется работа по созданию новых решений и развитию продуктов компании. Более 5 лет компания готовит кадры по технологическому стеку компании в Акселераторе ML START.

1.1 Организационная структура компании

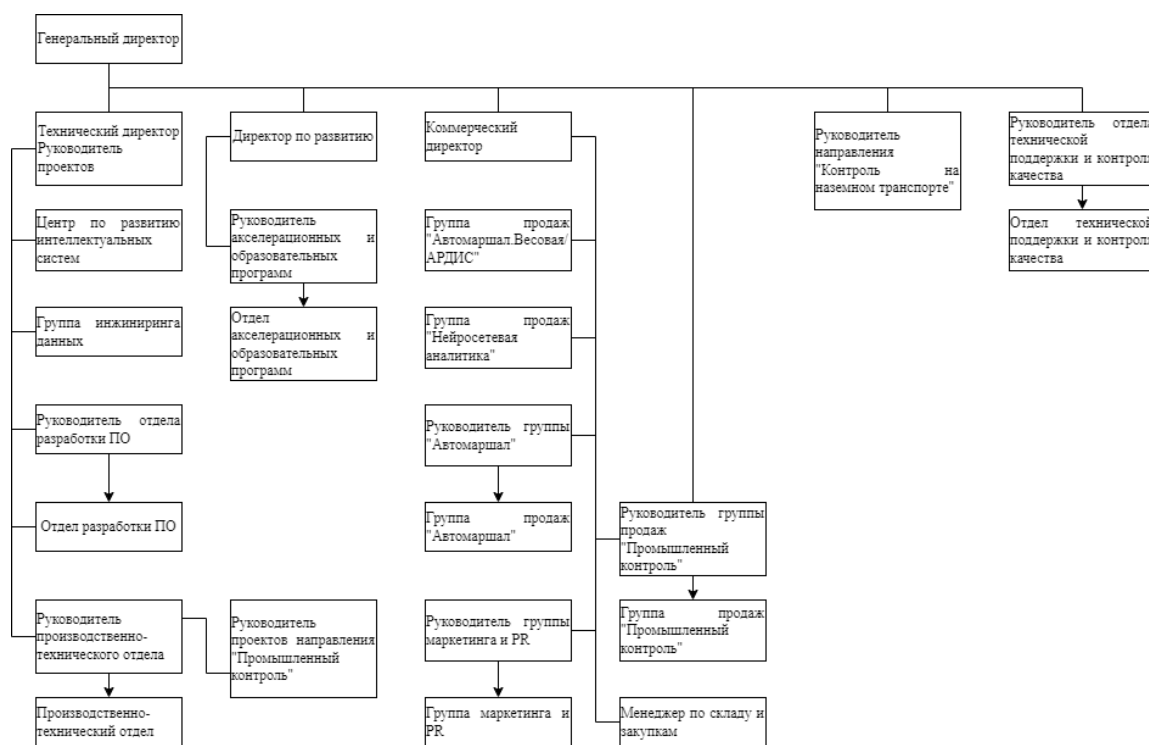


Рисунок 1 – организационная структура компании

1.2 Внутренний распорядок работы компании и охрана труда

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.

1.3 Должностные инструкции ИТ-специалиста компании

- 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 РЕВЬЮИРОВАНИЕ ПРОГРАММНЫХ ПРОДУКТОВ

Ревьюирование программных продуктов включает систематическую проверку кода для выявления ошибок и соответствия техническим требованиям. Основные задачи:

1. Осуществление ревью: Проверка кода на соответствие документации и стандартам.
2. Измерение характеристик: Оценка компонентов программного продукта для определения их соответствия заданным критериям. Исследование кода: Использование специализированных средств для выявления ошибок и отклонений от алгоритма.
3. Сравнительный анализ: Оценка различных программных решений для выбора оптимального.

Методы включают использование систем контроля версий и инструментов автоматизации, что обеспечивает эффективность и качество процесса ревьюирования.

2.1 Ревьюирование программного кода в соответствии с технической документацией

Основной целью ревью является: проверка соответствия кода технической документации, обнаружение ошибок и улучшение качества кода, оценка соблюдения стандартов кодирования и архитектуры.

Основные проблемы:

1. Несоответствие требованиям: В некоторых частях кода отсутствуют реализации описанных в технической документации функций.
2. Код-стайл: Зафиксированы отклонения от стандартов кодирования (например, несоответствие имен переменных, отступы).
3. Недостаточная документация: Некоторые методы и классы не содержат комментариев, что затрудняет понимание логики работы.
4. Проблемы с производительностью: Выявлены участки кода, которые могут быть оптимизированы для улучшения быстродействия.

Ревьюирование кода — это не только способ обнаружить ошибки, но и возможность для обучения и обмена знаниями. Хорошее ревью может значительно улучшить качество продукта и способствовать развитию команды.

2.2 Измерение характеристик компонентов программного продукта

Определение характеристик элементов программного продукта проводится с использованием метрик, которые помогают оценить качество программного обеспечения. В соответствии со стандартом ISO 25010:2011 качество ПО характеризуется такими атрибутами, как функциональность, надежность и производительность. Метрики могут быть как внутренними (например, размер кода), так и внешними (например, время отклика), и они служат для количественной оценки свойств программного продукта. Важно учитывать связь между внутренними и внешними характеристиками для более точной оценки качества.

Регулярное измерение характеристик элементов программного продукта позволяет выявлять узкие места, улучшать качество и адаптировать продукт к требованиям пользователей и бизнеса.

2.3 Исследование созданного программного кода с использованием специализированных программных средств

Анализ программного кода, созданного с использованием специализированных программных средств, включает в себя статический и динамический анализ. Статический анализ проводится без выполнения программы и дает возможность выявить уязвимости на стадии разработки. Инструменты, такие как SonarQube и Pylint, проверяют код на соответствие стандартам и наличие ошибок. В отличие от этого, динамический анализ требует выполнения кода и помогает находить проблемы, которые могут возникнуть во время работы.

2.4 Сравнительный анализ программных продуктов и средств разработки

Сравнительный анализ программных продуктов и средств разработки необходим для оценки их функциональности, производительности и удобства использования. Он помогает выявить сильные и слабые стороны различных решений, что позволяет выбрать наиболее подходящий инструмент для конкретных задач. Такой анализ также способствует оптимизации затрат, упрощает процесс обучения пользователей и помогает в принятии обоснованных решений при внедрении новых технологий в организации. Кроме того, он может способствовать выявлению трендов на рынке и помощи в разработке стратегий развития продуктов.

Проведение сравнительного анализа программных продуктов и средств разработки помогает избежать ошибок при выборе инструментов, оптимизировать затраты и повысить эффективность разработки.

3 ВЫПОЛНЯЕМЫЕ ЗАДАНИЯ

3.1 Задания первой недели

Первым делом я создала GIT репозиторий «Practice_PP03_IS31» для производственной практики, в котором расположила выполненные задания, отчеты и прочие документы текущей практики.

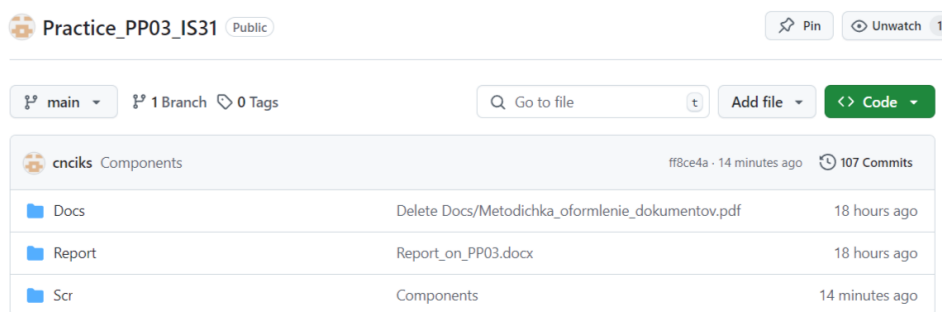


Рисунок 2 – репозиторий «Practice_PP03_IS31»

Далее нам дали задание разработать модуль обработки и работы с изображениями и модуля взаимодействия с пользователем, формирования и хранения данных: Изменение размера изображения.

Код программы представлен в приложении А.

Программный код нужно было разделить на следующие модули:

AppModul.py – модуль, который содержит классы приложения и связывает все компоненты. Представлен в приложении Б.

ImageModul.py – модуль, который содержит логику обработки изображений.

```
1 from PIL import Image
2
3 class ImageProcessor:
4     def __init__(self):
5         self.image = None
6         self.image_path = None
7
8     def load_image(self, path):
9         if path:
10             self.image_path = path
11             self.image = Image.open(self.image_path)
12
13     def resize_image(self, new_width, new_height):
14         if self.image is not None:
15             resized_image = self.image.resize((new_width, new_height), Image.LANCZOS)
16             return resized_image
17         return None
```

Рисунок 5 – модуль ImageModul.py программному коду «Изменение размеров изображения»

MainModul.py - основной файл, который запускает приложение.

```
1  from tkinter import Tk
2  from app import ImageUtilityApp
3
4  if __name__ == "__main__":
5      root = Tk()
6      app = ImageUtilityApp(root)
7      root.mainloop()
```

Рисунок 6 – модуль MainModul.py к программному коду «Изменение размеров изображения»

UiModul.py - в этом модуле описан пользовательский интерфейс. Представлен в приложении В.

3.2 Задания второй недели

Следующим заданием было: Проведение обратного проектирования, используя графический язык UML . Создание и описание диаграммы Компонентов, Сценариев использования, Последовательностей, Деятельности для программного кода «Изменение размеров изображения».



Рисунок 8 – диаграмма компонентов к программному коду «Изменение размеров изображения»

- Пользовательский интерфейс взаимодействует с обработчиком изображений, чтобы отправлять выбранные пользователем параметры.
- Компонент загрузки изображений передает загруженное изображение в обработчик изображений.



Рисунок 9 – диаграмма сценариев использования к программному коду «Изменение размеров изображения»

- Загрузить изображение - Пользователь загружает изображение в систему.
- Выбрать размер - Пользователь выбирает новый размер для изображения (ширину и высоту).
- Пользователь взаимодействует со всеми вышеуказанными сценариями использования.



Рисунок 10 – диаграмма последовательностей к программному коду «Изменение размеров изображения»

- Пользователь выбирает изображение.
- Система загружает выбранное изображение.
- Пользователь вводит новые размеры (ширина и высота).
- Система проверяет допустимость новых размеров.
- Система применяет изменения к изображению.
- Система уведомляет пользователя об успешном изменении размера.

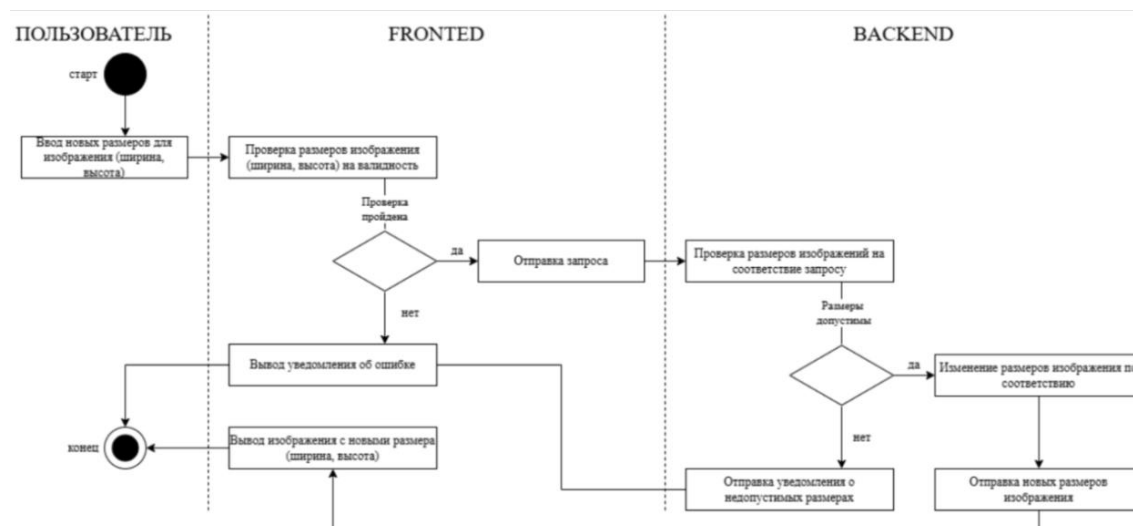


Рисунок 11 – диаграмма деятельности к программному коду «Изменение размеров изображения»

- Пользователь вводит новые размеры для изображения (ширину и высоту).
- Fronted проверяет размеры изображения (ширину и высоту) на валидность.
- Если проверка не удалась:
- Fronted выводит уведомление об ошибке.
- Если проверка удалась:
- Fronted отправляет запрос Backend.
- Backend проверяет размеры изображения на соответствие запросу.
- Если размеры не допустимы:
- Backend отправляет уведомление о недопустимых размерах.
- Fronted выводит уведомление об ошибке.
- Если размеры допустимы:
- Backend изменяет размеры изображения по соответствию.
- Backend отправляет новые размеры изображения.
- Fronted выводит изображение с новыми размерами (шириной и высотой).

Также нужно было провести скоростные показатели программного кода «Изменение размеров изображения» и его модулей.

Таблица 1 – Скоростные показатели программного кода и его модулей

Программный код «Изменение размеров изображения»	Время загрузки изображения 0,032576 секунд.
Модуль AppModuleTime.py	Время выбора файла: 5,5707 секунд.
Модуль ImageModuleTime.py	Время получения размера (с): 0.00015687942504882812, 'Время получения разрешения (с): 9.5367431640625e-07, Время получения даты создания (с): 6.771087646484375e-05
Модуль MainModuleTime.py	Время выполнения load_image: 0.021446 секунд. Время выполнения get_creation_date: 0.000068 секунд. Время выполнения get_image_info: 0.012225 секунд.
Модуль UiModuleTime.py	Время открытия изображения: 0,028281. Время получения информации: 0,027827.

Таблица 2 – Размеры программного кода и его модулей

Программный код «Изменение размеров изображения»	Размер: 876254. Размер модуля sys 72 байт. Размер модуля PIL_Image 72 байт.
Модуль AppModuleTime.py	Размер объекта 48 байт
Модуль ImageModuleTime.py	Размер объекта 48 байт
Модуль MainModuleTime.py	Размер объекта 76640 байт.
Модуль UiModuleTime.py	Размер объекта 48 байт

Я провела анализ средств разработки программ на языке программирования Python.

Visual Studio Code

Плюсы:

- Легкая и быстрая.
- Поддержка множества расширений для различных языков программирования.
- Встроенная поддержка Git и других систем контроля версий.
- Кроссплатформенность: доступна для Windows, macOS и Linux.
- Активное сообщество и регулярные обновления.

Минусы:

- Меньше функциональных возможностей "из коробки" по сравнению с более тяжеловесными IDE.
- Может требовать настройки для оптимизации под специфические технологии.
- Требуется настройка и установка расширений для полноценной работы с Python.
- Некоторые пользователи могут находить его менее интуитивно понятным по сравнению с полными IDE.

Вид приложения представлен в приложении Ж.

PyCharm

Плюсы PyCharm:

- Поддержка работы с виртуальными окружениями и пакетами (pip, conda).
- Отладка и тестирование: Встроенные инструменты для отладки и поддержки юнит-тестирования упрощают процесс нахождения и устранения ошибок.

- Управление зависимостями: PyCharm позволяет легко работать с виртуальными окружениями и управлять зависимостями с помощью `pip` или `conda`.
- Полнофункциональная IDE, специально разработанная для Python.
- Интеграция с системами контроля версий: Поддержка Git.
- Мощные инструменты для анализа кода: PyCharm предоставляет встроенные линтеры и инструменты для анализа кода, что помогает поддерживать качество кода.

Минусы PyCharm:

- Требовательность к ресурсам: PyCharm может быть «тяжелой» по сравнению с более легковесными редакторами кода, такими как VS Code, и может потреблять много оперативной памяти.
- Цена: Хотя существует бесплатная версия PyCharm Community, многие функции доступны только в платной версии Professional, что может быть ограничивающим фактором для некоторых разработчиков.
- Сложность для новичков: Из-за большого количества функций и настроек, начинающим пользователям может быть сложно освоить интерфейс.
- Производительность: В некоторых случаях, особенно на менее мощных компьютерах, PyCharm может работать медленно.

Вид приложения представлен в приложении И.

IDLE (Integrated Development and Learning Environment)

Плюсы IDLE:

- Небольшой размер и быстрая работа. Среда весит всего несколько мегабайт и не требует отдельной установки, так как устанавливается на компьютер вместе с Python.
- Интерактивный режим. Написанный в нём код сразу же выполняется, без сохранения в файл. Это удобно для быстрой проверки кода и обучения языку.

- Подсветка синтаксиса. Чтение и написание кода становится более удобным — каждая конструкция языка подсвечивается определённым цветом.

Минусы IDLE:

- Ограниченная функциональность. Она не подойдёт для крупных проектов с большим количеством файлов, где требуется точно настроить параметры отладки и провести глубокий анализ кода.
- Не подходит для серьёзной разработки. Для серьёзной разработки лучше использовать более функциональные среды, например PyCharm или VS Code.

Вид приложения представлен в приложении К.

Из представленных выше средств разработки программ я выбрала IDLE, потому что:

- IDLE имеет интуитивно понятный интерфейс, что делает его доступным для новичков. Он не перегружен функционалом, что позволяет сосредоточиться на обучении программированию.
- IDLE позволяет легко читать и писать код благодаря подсветке синтаксиса, что помогает лучше понимать структуру программы.
- IDLE работает на различных операционных системах, таких как Windows, macOS и Linux, что позволяет использовать его на любом из этих платформ.
- Так как IDLE является частью Python, существует множество ресурсов и документации, объясняющих, как эффективно использовать эту среду.
- Не требует сложной установки или настройки, что является преимуществом для начинающих разработчиков или людей, которые просто хотят быстро начать писать код.

ЗАКЛЮЧЕНИЕ

На практике я освоила следующее:

- 1 Осуществление ревьюирования программного кода в соответствии с технической документацией;
- 2 Выполнение измерений характеристик компонентов программного продукта;
- 3 Производство исследования созданного программного кода с использованием специализированных программных средств;
- 4 Проведение сравнительного анализа программных продуктов и средств разработки.

Также освоила основной вид деятельности по направлению «Ревьюирование программных модулей» и формирование соответствующих общих и профессиональных компетенций.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

Git репозиторий - https://github.com/cnciks/Practice_PP03_IS31/tree/main

Информация о «Малленом Системс» - <https://www.mallenom.ru/>

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

Работа с библиотеками - <https://metanit.com/sharp/tutorial/3.46.php>

Графический редактор диаграмм - <https://app.diagrams.net/>

UML - <https://practicum.yandex.ru/blog/uml-diagrammy/>

Пример измерения скорости используя Time -
<https://www.geeksforgeeks.org/how-to-check-the-execution-time-of-python-script/>

```

1  import tkinter as tk
2  from tkinter import filedialog, messagebox, simpledialog
3  from PIL import Image, ImageTk
4
5  class ImageUtilityApp:
6  def __init__(self, root):
7      self.root = root
8      self.root.title("Утилита работы с изображениями, практикант: Шевель Милена Александровна")
9
10     # Переменные для хранения изображения
11     self.image = None
12     self.image_path = None
13
14     # Кнопки
15     self.load_button = tk.Button(root, text="Загрузить изображение", command=self.load_image)
16     self.load_button.grid(row=0, column=0, padx=10, pady=10)
17
18     self.resize_button = tk.Button(root, text="Изменить размер", command=self.resize_image)
19     self.resize_button.grid(row=1, column=0, padx=10, pady=10)
20
21     # Блок информации
22     self.info_label = tk.Label(root, text="Информация об изображении:")
23     self.info_label.grid(row=2, column=0, padx=10, pady=10)
24
25     self.info_text = tk.Text(root, width=40, height=10)
26     self.info_text.grid(row=3, column=0, padx=20, pady=20)
27
28     # Область для отображения изображения
29     self.image_label = tk.Label(root)
30     self.image_label.grid(row=0, column=1, rowspan=4)
31
32  def load_image(self):
33     self.image_path = filedialog.askopenfilename(filetypes=[("Image files", "*.jpg;*.jpeg;*.png;*.gif")])
34
35     if not self.image_path:
36         return
37
38     self.image = Image.open(self.image_path)
39     self.display_image(self.image)
40     self.update_info()
41
42  def resize_image(self):
43     if self.image is None:
44         messagebox.showwarning("Warning", "Сначала загрузите изображение")
45         return
46
47     # Выбор новых размеров
48     new_width = simpledialog.askinteger("Изменить размер", "Новая ширина:")
49     new_height = simpledialog.askinteger("Изменить размер", "Новая высота:")
50
51     if new_width is not None and new_height is not None:
52         resized_image = self.image.resize((new_width, new_height), Image.LANCZOS)
53         self.display_image(resized_image)
54
55  def display_image(self, img):
56     img.thumbnail((400, 400)) # Уменьшается изображение, если оно слишком большое
57     self.tk_image = ImageTk.PhotoImage(img)
58     self.image_label.config(image=self.tk_image)
59     self.image_label.image = self.tk_image
60
61  def update_info(self):
62     self.info_text.delete(1.0, tk.END)
63     if self.image:
64         width, height = self.image.size
65         self.info_text.insert(tk.END, f"Файл: {self.image_path}\\n. ")
66         self.info_text.insert(tk.END, f"Размер: {width}x{height}")
67     else:
68         self.info_text.insert(tk.END, "Изображение не загружено")
69
70  if __name__ == "__main__":
71     root = tk.Tk()
72     app = ImageUtilityApp(root)
73     root.mainloop()

```

Рисунок 3 – программный код «Изменение размеров изображения»

```

1  import tkinter as tk
2  from tkinter import filedialog, messagebox, simpledialog
3  from ui import UI
4  from image_processor import ImageProcessor
5
6  class ImageUtilityApp:
7      def __init__(self, root):
8          self.root = root
9          self.root.title("Утилита работы с изображениями, практикант: Шевель Милена Александровна")
10
11         # Инициализация пользовательского интерфейса
12         self.ui = UI(root)
13         self.image_processor = ImageProcessor()
14
15         # Связывание кнопок с действиями
16         self.ui.load_button.config(command=self.load_image)
17         self.ui.resize_button.config(command=self.resize_image)
18
19     def load_image(self):
20         self.image_processor.load_image(filedialog.askopenfilename(filetypes=[("Image files", "*.jpg;*.jpeg;*.png;*.gif")]))
21         if not self.image_processor.image_path:
22             return
23         self.ui.display_image(self.image_processor.image)
24         self.ui.update_info(self.image_processor.image_path, self.image_processor.image.size)
25
26     def resize_image(self):
27         if self.image_processor.image is None:
28             messagebox.showwarning("Warning", "Сначала загрузите изображение")
29             return
30
31         new_width = simpledialog.askinteger("Изменить размер", "Новая ширина:")
32         new_height = simpledialog.askinteger("Изменить размер", "Новая высота:")
33
34         if new_width is not None and new_height is not None:
35             resized_image = self.image_processor.resize_image(new_width, new_height)
36             self.ui.display_image(resized_image)

```

Рисунок 4 — модуль AppModul.py к программному коду «Изменение размеров изображения»

```

1     import tkinter as tk
2     from PIL import ImageTk
3
4     class UI:
5     def __init__(self, root):
6         # Кнопки
7         self.load_button = tk.Button(root, text="Загрузить изображение")
8         self.load_button.grid(row=0, column=0, padx=10, pady=10)
9
10        self.resize_button = tk.Button(root, text="Изменить размер")
11        self.resize_button.grid(row=1, column=0, padx=10, pady=10)
12
13        # Блок информации
14        self.info_label = tk.Label(root, text="Информация об изображении:")
15        self.info_label.grid(row=2, column=0, padx=10, pady=10)
16
17        self.info_text = tk.Text(root, width=40, height=10)
18        self.info_text.grid(row=3, column=0, padx=20, pady=20)
19
20        # Область для отображения изображения
21        self.image_label = tk.Label(root)
22        self.image_label.grid(row=0, column=1, rowspan=4)
23
24    def display_image(self, img):
25        img.thumbnail((400, 400))
26        self.tk_image = ImageTk.PhotoImage(img)
27        self.image_label.config(image=self.tk_image)
28        self.image_label.image = self.tk_image
29
30    def update_info(self, image_path, size):
31        self.info_text.delete(1.0, tk.END)
32        if image_path:
33            width, height = size
34
35            self.info_text.insert(tk.END, f"Файл: {image_path}\\n")
36            self.info_text.insert(tk.END, f"Размер: {width}x{height}")
37        else:
38            self.info_text.insert(tk.END, "Изображение не загружено")

```

Рисунок 7 – модуль UiModul.py к программному коду «Изменение размеров изображения»

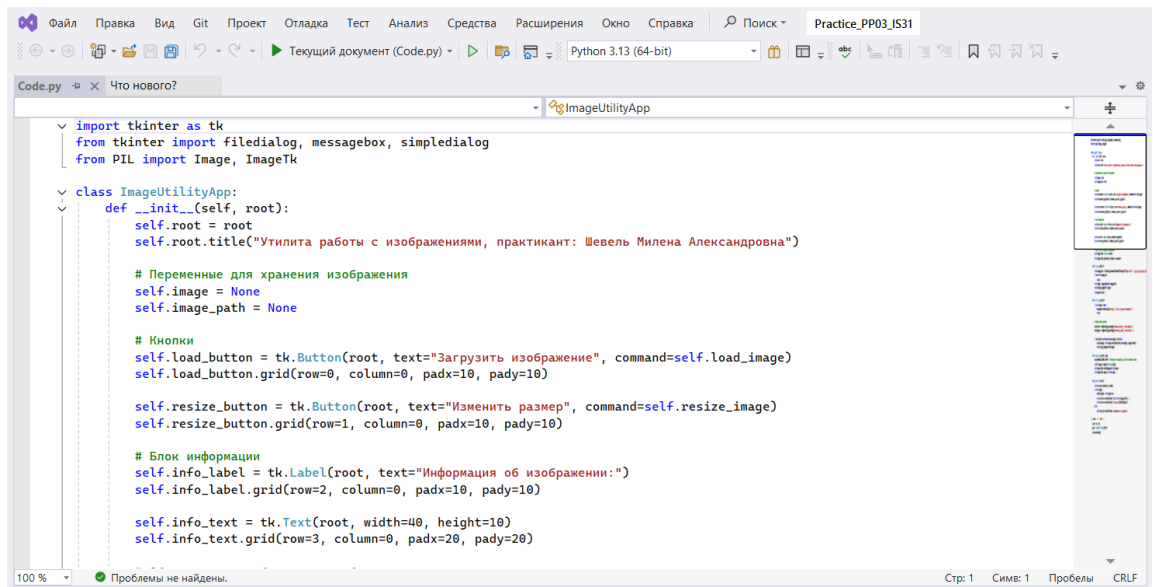


Рисунок 12 – вид приложения Visual Studio

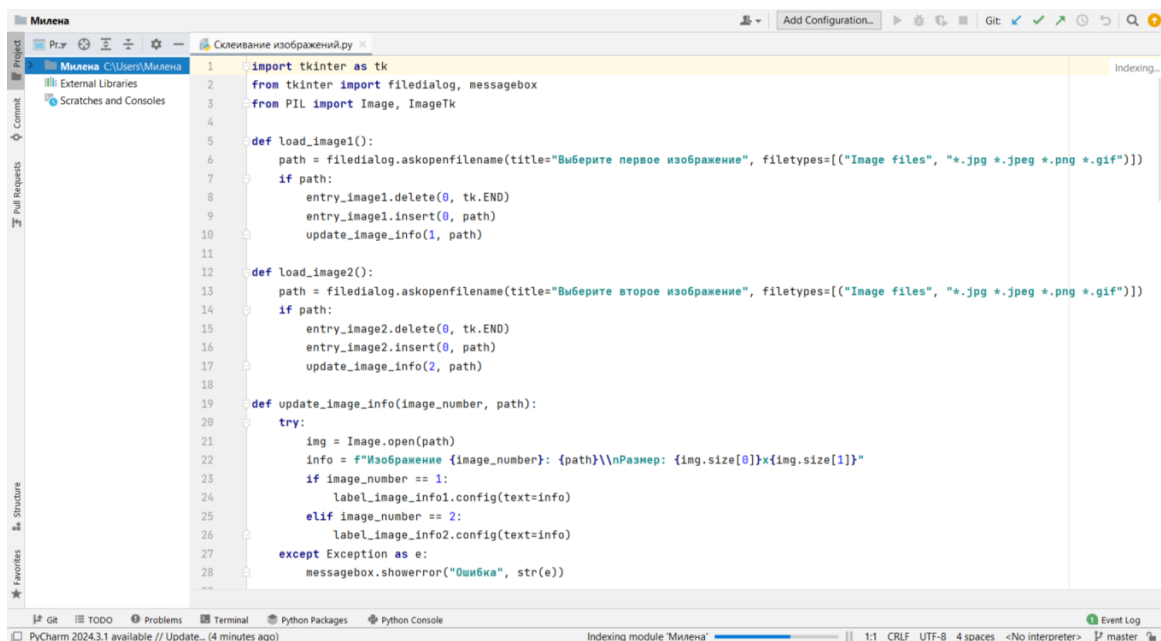


Рисунок 13 – вид приложения PyCharm

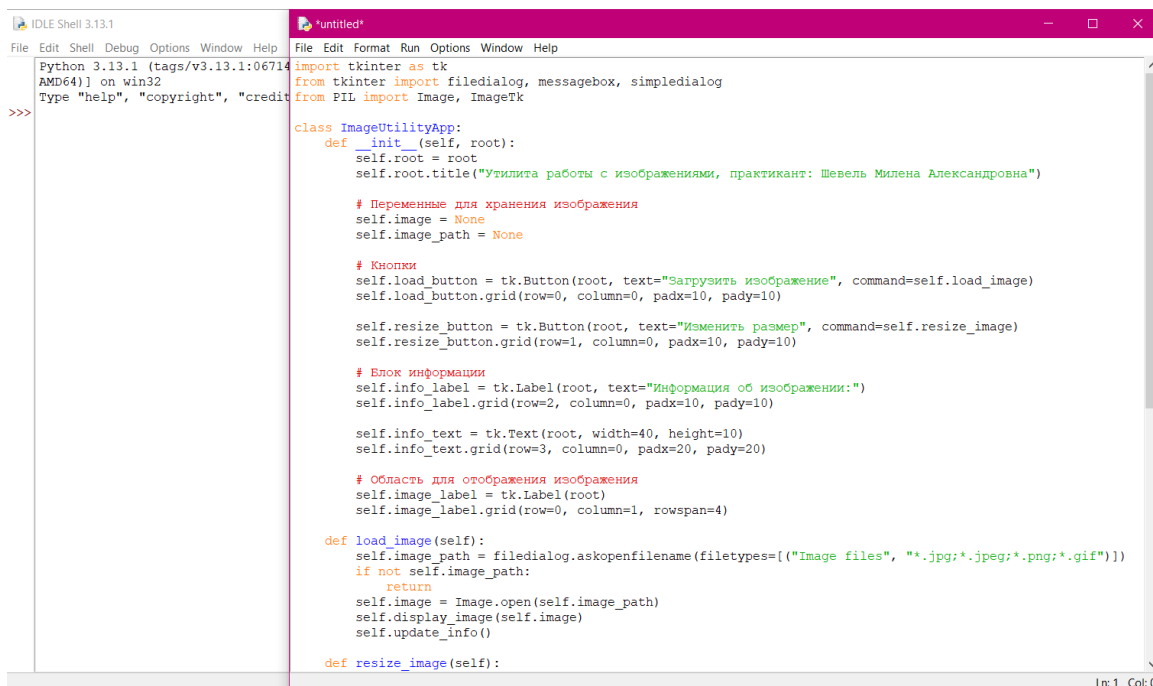


Рисунок 14 – вид приложения IDLE (Integrated Development and Learning)