### \*\*Доклад по учебной практике: Разработка программного обеспечения «Менеджер задач»\*\*

\*\*Введение\*\*

В рамках учебной практики по модулю ПМ.05 «Проектирование и разработка информационных систем» передо мной стояла задача разработать приложение «Менеджер задач». Практика проходила на базе Колледжа Вятского государственного университета с 13 января по 21 июня 2025 года. Основная цель заключалась в освоении полного цикла создания прикладного программного обеспечения — от анализа требований до реализации и тестирования.

\*\*Актуальность проекта\*\*

Современный ритм жизни требует эффективного управления временем и задачами. Существующие аналоги, такие как To-Do List или Trello, обладают своими преимуществами, но часто либо слишком сложны, либо ограничены в функционале. Мой «Менеджер задач» призван решить эту проблему, предлагая простой, но мощный инструмент для планирования дел с возможностью настройки дедлайнов, статусов и экспорта данных.

\*\*Анализ предметной области\*\*

Перед началом разработки я изучил существующие решения. Например, приложение «To-Do-List» от Wynand1004 отличается простотой, но не поддерживает автосохранение, что может привести к потере данных. Другой аналог, «ToDoList-Python», работает через командную строку, что неудобно для рядового пользователя. Третий вариант, «ToDo», имеет удобный интерфейс, но устаревший дизайн и отсутствие редактирования задач.

На основе этого анализа я сформулировал требования к своему приложению:

- \*\*Простота интерфейса\*\* — интуитивно понятное управление даже для новичков.

- \*\*Автосохранение\*\* — все данные хранятся в JSON-файле, что исключает потерю информации.

- \*\*Гибкость\*\* — поддержка дедлайнов, статусов (например, «В процессе», «Выполнено»), а также импорт/экспорт в формат iCalendar (.ics).

\*\*Реализация проекта\*\*

Для разработки был выбран язык Python 3.10 благодаря его простоте и богатой экосистеме библиотек. Графический интерфейс создан с использованием PySide6 — это официальная библиотека для работы с фреймворком Qt, которая позволяет разрабатывать кроссплатформенные приложения.

Ключевые этапы работы:

1. \*\*Настройка окружения\*\*

- Установка Python, Git, Visual Studio Code.

- Создание репозитория на GitHub для контроля версий.

2. \*\*Проектирование архитектуры\*\*

- Разработка UML-диаграмм для визуализации структуры приложения.

- Определение формата хранения данных (JSON для локального использования и .ics для совместимости с календарями).

3. \*\*Программная реализация\*\*

- Главное окно содержит список задач, кнопки управления (добавление, удаление, редактирование) и элементы для фильтрации.

- Диалоговое окно редактирования задачи включает поля для названия, описания, статуса и дедлайна.

- Для обработки дат использована библиотека `datetime`, а для работы с .ics-файлами — `icalendar`.

4. \*\*Тестирование\*\*

Проверялись все сценарии использования:

- Корректность сохранения и загрузки данных.

- Валидация ввода (например, запрет на пустые названия задач или даты в прошлом).

- Работа функций импорта/экспорта.

\*\*Примеры работы программы\*\*

- При запуске приложения пользователь видит список задач, отсортированных по статусу.

- Добавление новой задачи занимает несколько секунд: достаточно ввести название, описание и выбрать дедлайн.

- Если срок выполнения задачи истек, её статус автоматически меняется на «Просрочена».

- Задачи можно экспортировать в .ics и открывать в Google Calendar или Outlook.

\*\*Заключение\*\*

В результате практики я не только разработал рабочее приложение, но и освоил ключевые этапы создания ПО:

- Анализ требований и проектирование.

- Написание кода с использованием современных инструментов.

- Тестирование и документирование.

«Менеджер задач» уже готов к использованию, но в будущем его можно улучшить, добавив:

- Синхронизацию с облачными сервисами.

- Напоминания о дедлайнах.

- Поддержку совместной работы над задачами.

\*\*Вывод\*\*

Эта практика стала для меня важным шагом в профессиональном развитии. Я научился не только писать код, но и мыслить как инженер, учитывая требования пользователей и потенциальные риски. Уверен, что полученный опыт поможет мне в дальнейшей работе над более сложными проектами.

\*\*Благодарю за внимание!\*\* Если у вас есть вопросы по реализации или тестированию, буду рад на них ответить.