Отчёт по использованию инструмента управления временем SingularityApp (на Windows)

В рамках эксперимента я решил опробовать **SingularityApp** — приложение, которое позиционируется как современный инструмент для организации времени и дел. Оно работает на разных устройствах, полностью на русском языке и предлагает функции создания проектов (папок), задач и подзадач, расстановки приоритетов, а также показа планов на сегодня в отдельном окне.

Честно говоря, мой обычный подход к управлению временем немного другой. Я предпочитаю делать всё заранее, чтобы потом не спешить и иметь свободные окна для чего-то внепланового. Когда у меня есть время, я не трачу его на планирование или размышления о том, в какой день выполнить задачу. Если есть возможность и настроение — я просто беру и делаю. Поэтому строгая система с календарём и расписанием мне, по сути, не нужна. Singularity, как мне показалось, больше рассчитан на тех, кто любит тщательно расписывать дела по дням и неделям. Я же считаю, что чем быстрее закрою задачу, тем лучше.

В этом плане приложение оказалось для меня не слишком полезным. Функция календаря доступна только по платной подписке (рис. 1), и это сильно ограничивает возможности. По сути **в бесплатной версии** остаётся только работа с проектами, вкладка «Сегодня» и раздел «Планы». При этом вкладка «Планы» оставила смешанные впечатления. Её интерфейс показался не самым удобным: например, почему слева от даты нельзя добавить кнопку «+» для создания задачи на день. Это выглядело бы гораздо логичнее и удобнее чем вести мышку в самый низ приложения для нажатия на кнопки там или использовать сочетания клавиш (Ctrl + N) . Кроме того, хотелось бы иметь возможность скрыть задачи определённого дня и открывать по необходимости, чтобы запланированные дела, которые пока не актуальны, не маячили постоянно перед глазами (рис. 2).

Тем не менее, у Singularity есть и свои плюсы. Мне понравилось, что приложение работает и на компьютере, и на телефоне. Это удобно: например, можно быстро записать дело в телефон на работе, в школе или университете, и оно автоматически появится на компьютере дома — не нужно как-то пересылать список задач или смотреть в телефон лишний раз.

Понравились функции управления временем задач (рис. 4) и **«Повторять»**, которая позволяет задачам автоматически переноситься на следующий день, неделю и т.д, пока они не будут выполнены. Это вроде бы полезно, есть задача которую ты не хочешь делать продолжительное время, но она постоянно всплывает перед глазами и ты хочешь от нее поскорее избавиться. Также есть специальные папка «Когда-нибудь» для дел без привязки ко времени что удобно.

Краткий список плюсов и минусов

Плюсы:

Синхронизация между устройствами.
Приятный дизайн с выбором нескольких темных и светлых тем (рис. 3).
Большое количество языков, в том числе русский.
Поиск по задачам.
Удобное изменение таймера задач (рис. 4).

Минусы:

Большинство функций находится в платной подписке, без которой приложение напоминает «Блокнот» или «Заметки»

Итог по использованию SingularityApp

В процессе работы с SingularityApp был создан проект «Практика», где для заданий 2.1-3 практики были оформлены отдельные задачи и подзадачи (рис. 5-7). Для контроля использовались вкладка «Сегодня» и раздел «Планы»

В целом приложение оставило смешанное впечатление. Оно предоставляет удобные инструменты для создания и структурирования задач, имеет современный интерфейс и синхронизацию между устройствами. Однако бесплатная версия ограничена: отсутствует календарь, добавление задач на конкретный день реализовано не самым удобным образом, а в разделе «Планы» невозможно скрывать неактуальные задачи.

Опыт использования SingularityApp можно оценить как нейтральный: приложение нельзя назвать неудобным, но и особых преимуществ по сравнению с простыми заметками в бесплатной версии оно не дало.

Таким образом, **SingularityApp стоит рассматривать как инструмент для тех, кто ищет комплексное и гибкое планирование**, готов инвестировать в подписку и использовать календарь как основную основу организации времени. Для пользователей с более простым и «живым» подходом к задачам, приложение может оказаться избыточным и не принести ощутимой пользы.

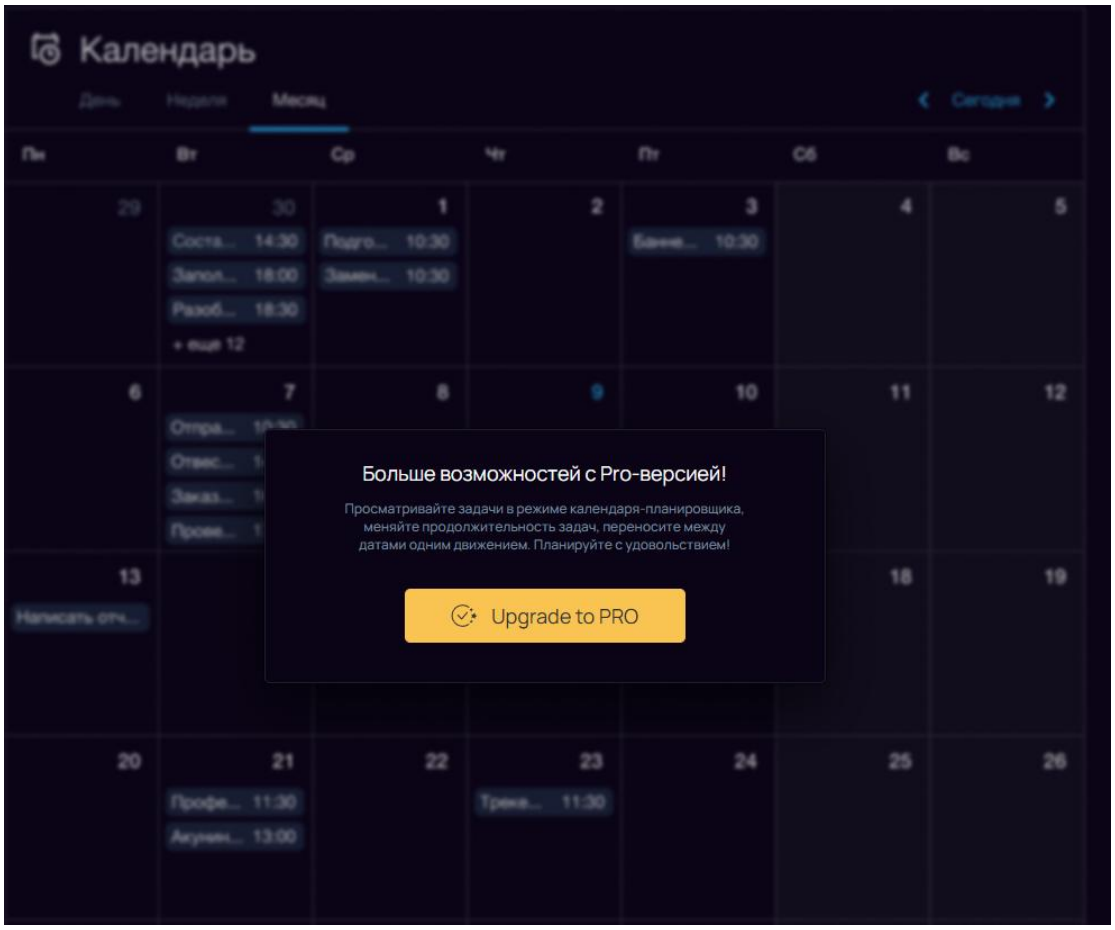
Календарь
День
Неделя
Месяц
Сегодня
Пн
Вт
Ср
Чт
Пт
Сб
Вс
Больше возможностей с Pro-версией!
Просматривайте задачи в режиме календаря-планировщика, меняйте продолжительность задач, переносите между датами одним движением. Планируйте с удовольствием!
Upgrade to PRO

рис.1

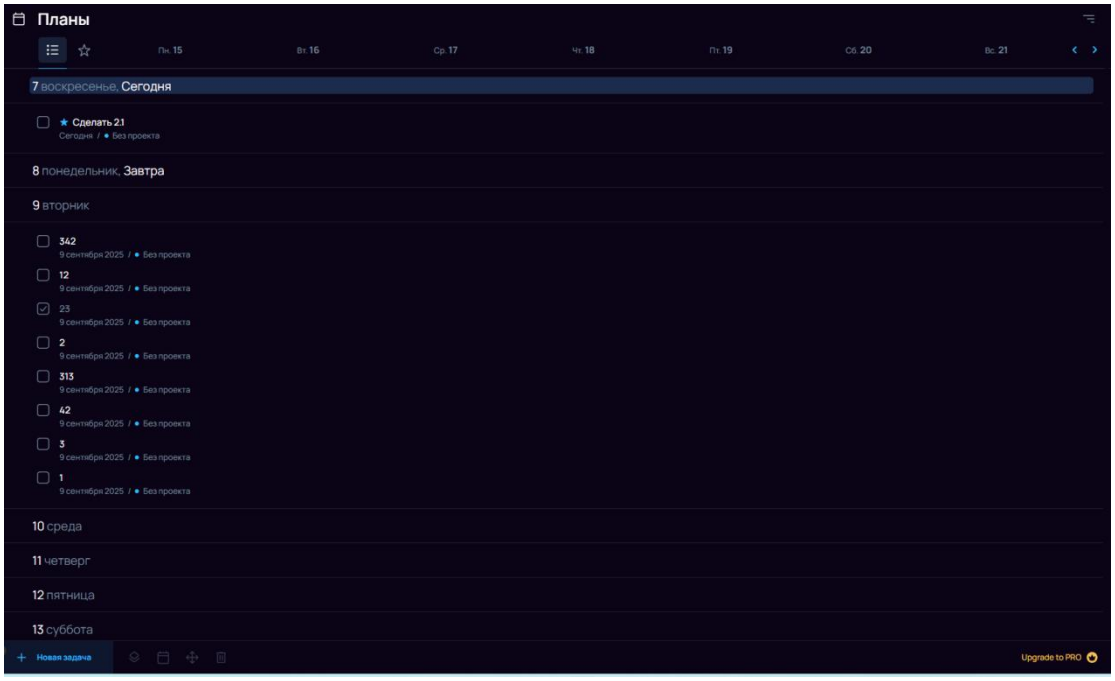
Планы
Пн. 15
Вт. 16
Ср. 17
Чт. 18
Пт. 19
Сб. 20
Вс. 21
7 воскресенье, Сегодня
Сделать 2.1
Сегодня / Без проекта
8 понедельник, Завтра
9 вторник
342
12
23
2
313
42
3
1
9 сентября 2025 / Без проекта
10 среда
11 четверг
12 пятница
13 суббота
Новая задача
Upgrade to PRO

рис.2

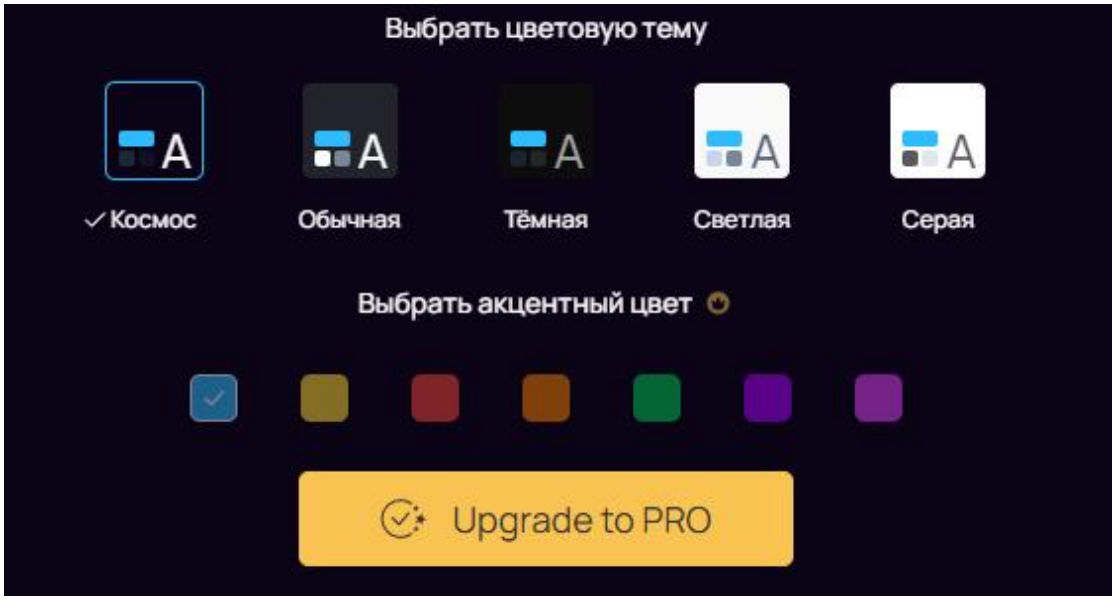
Выбрать цветовую тему
✓ Космос
Обычная
Тёмная
Светлая
Серая
Выбрать акцентный цвет
Upgrade to PRO

рис. 3

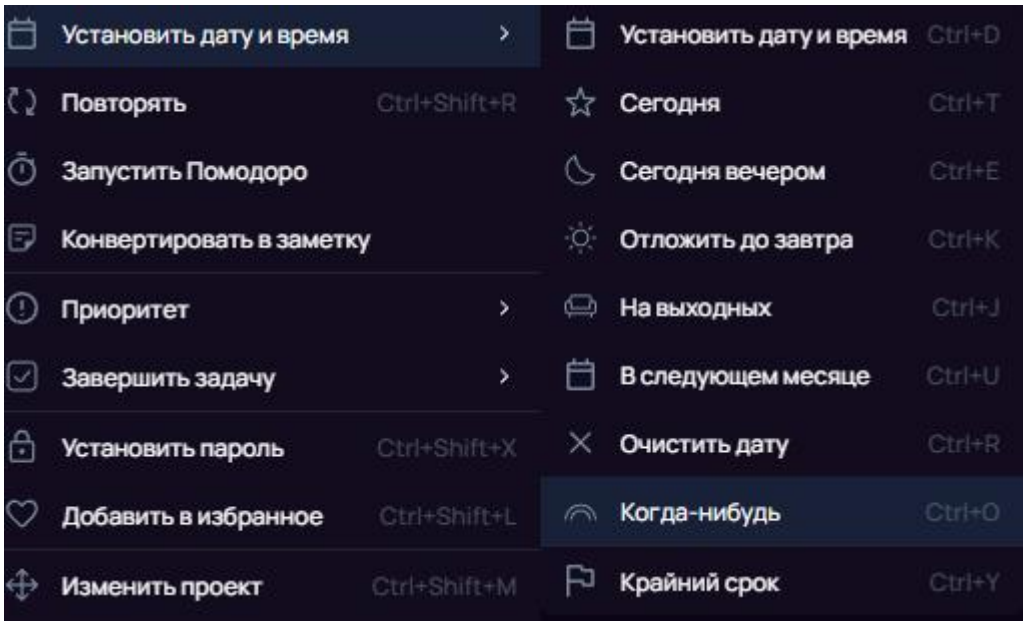
Установить дату и время
Повторять Ctrl+Shift+R
Запустить Помодоро
Конвертировать в заметку
Приоритет
Завершить задачу
Установить пароль Ctrl+Shift+X
Добавить в избранное Ctrl+Shift+L
Изменить проект Ctrl+Shift+M
Установить дату и время Ctrl+D
Сегодня Ctrl+T
Сегодня вечером Ctrl+E
Отложить до завтра Ctrl+K
На выходных Ctrl+J
В следующем месяце Ctrl+U
Очистить дату Ctrl+R
Когда-нибудь Ctrl+O
Крайний срок Ctrl+Y

рис. 4

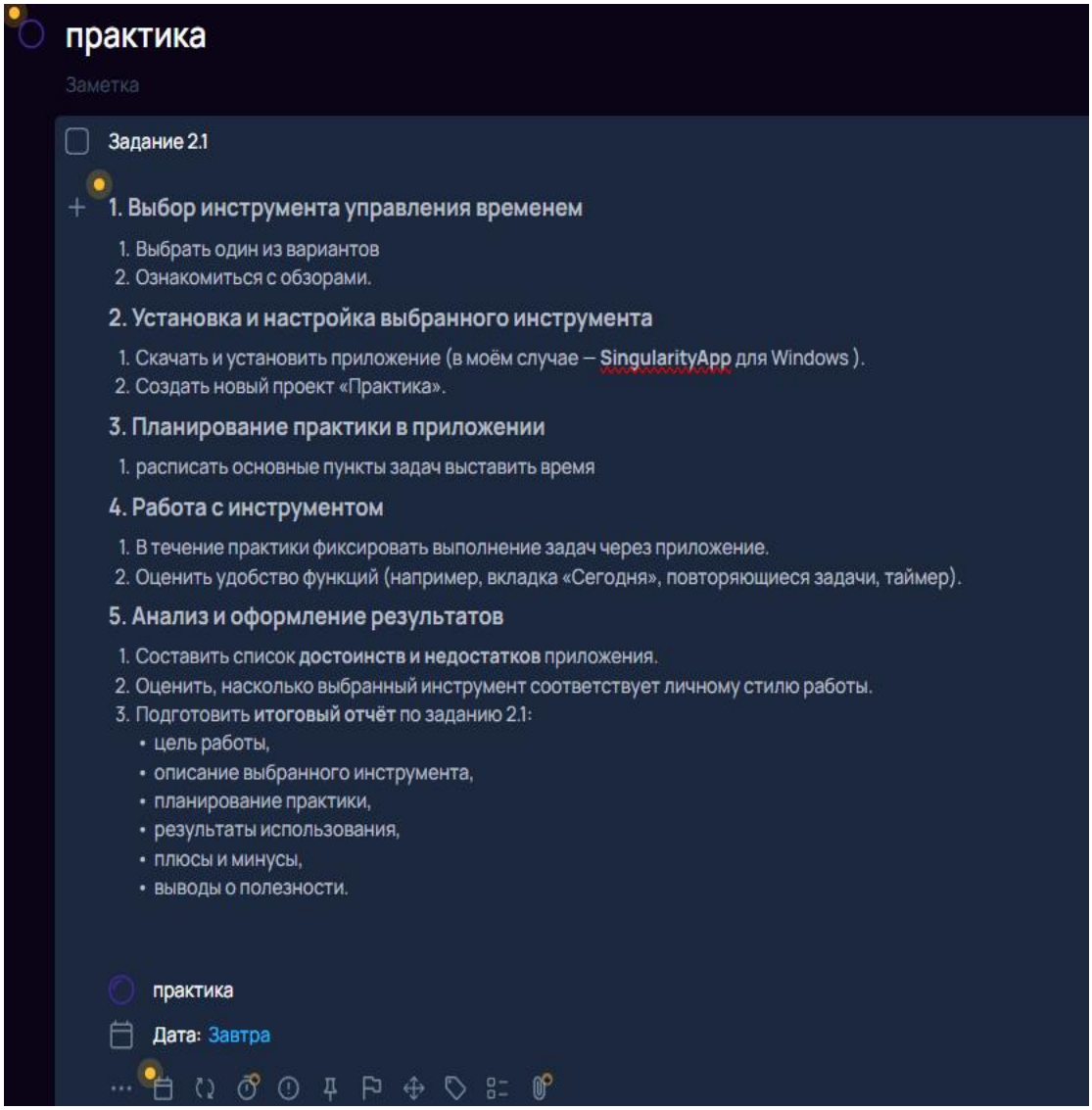
практика
Заметка
Задание 2.1
1. Выбор инструмента управления временем
1. Выбрать один из вариантов
2. Ознакомиться с обзорами.
2. Установка и настройка выбранного инструмента
1. Скачать и установить приложение (в моём случае – SingularityApp для Windows).
2. Создать новый проект «Практика».
3. Планирование практики в приложении
1. расписать основные пункты задач выставить время
4. Работа с инструментом
1. В течение практики фиксировать выполнение задач через приложение.
2. Оценить удобство функций (например, вкладка «Сегодня», повторяющиеся задачи, таймер).
5. Анализ и оформление результатов
1. Составить список достоинств и недостатков приложения.
2. Оценить, насколько выбранный инструмент соответствует личному стилю работы.
3. Подготовить итоговый отчёт по заданию 2.1:
• цель работы,
• описание выбранного инструмента,
• планирование практики,
• результаты использования,
• плюсы и минусы,
• выводы о полезности.
практика
Дата: Завтра

рис. 5

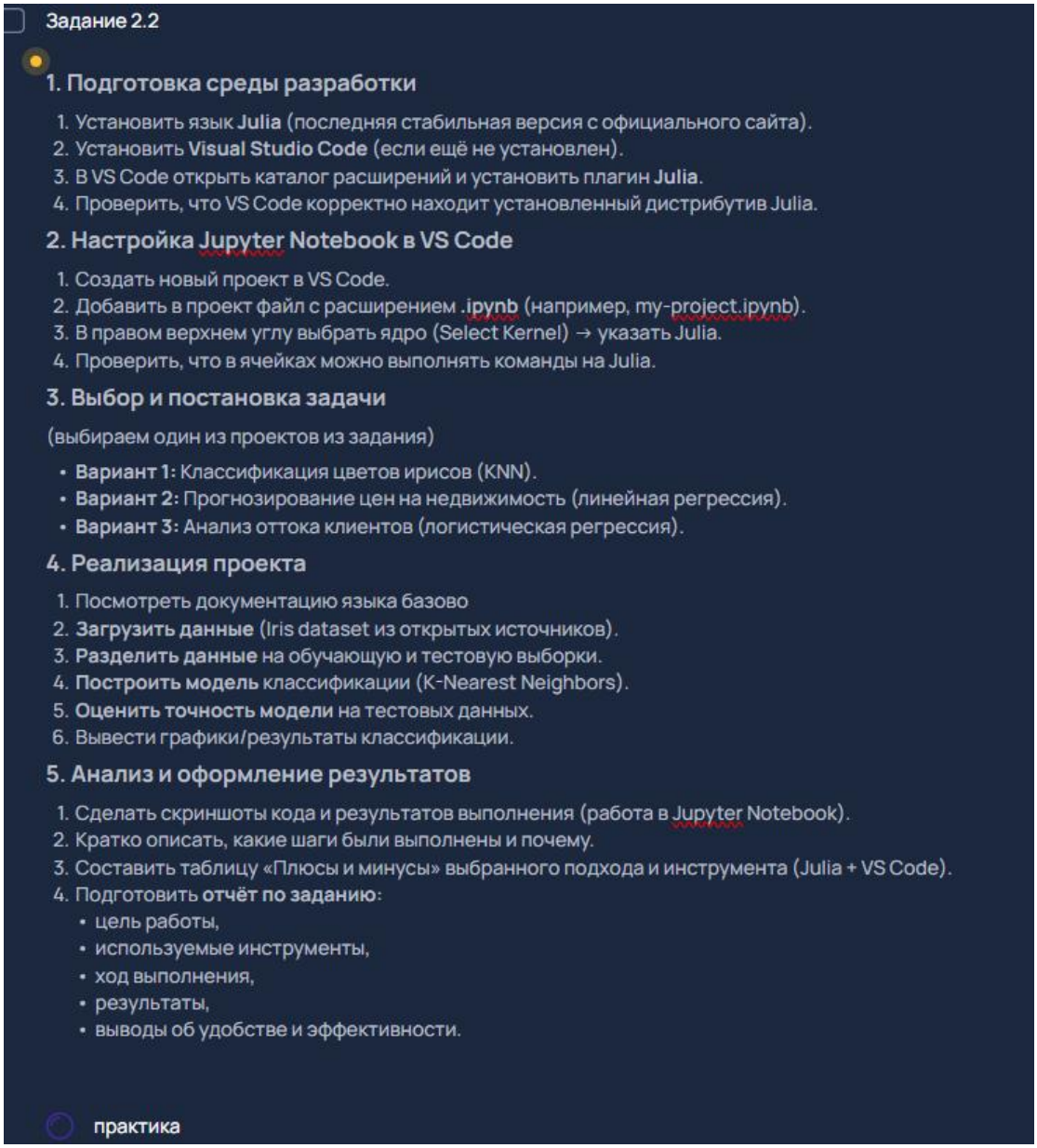
Задание 2.2
1. Подготовка среды разработки
1. Установить язык Julia (последняя стабильная версия с официального сайта).
2. Установить Visual Studio Code (если ещё не установлен).
3. В VS Code открыть каталог расширений и установить плагин Julia.
4. Проверить, что VS Code корректно находит установленный дистрибутив Julia.
2. Настройка Jupyter Notebook в VS Code
1. Создать новый проект в VS Code.
2. Добавить в проект файл с расширением .ipynb (например, my-project.ipynb).
3. В правом верхнем углу выбрать ядро (Select Kernel) → указать Julia.
4. Проверить, что в ячейках можно выполнять команды на Julia.
3. Выбор и постановка задачи
(выбираем один из проектов из задания)
• Вариант 1: Классификация цветов ирисов (KNN).
• Вариант 2: Прогнозирование цен на недвижимость (линейная регрессия).
• Вариант 3: Анализ оттока клиентов (логистическая регрессия).
4. Реализация проекта
1. Посмотреть документацию языка базово
2. Загрузить данные (Iris dataset из открытых источников).
3. Разделить данные на обучающую и тестовую выборки.
4. Построить модель классификации (K-Nearest Neighbors).
5. Оценить точность модели на тестовых данных.
6. Вывести графики/результаты классификации.
5. Анализ и оформление результатов
1. Сделать скриншоты кода и результатов выполнения (работа в Jupyter Notebook).
2. Кратко описать, какие шаги были выполнены и почему.
3. Составить таблицу «Плюсы и минусы» выбранного подхода и инструмента (Julia + VS Code).
4. Подготовить отчёт по заданию:
• цель работы,
• используемые инструменты,
• ход выполнения,
• результаты,
• выводы об удобстве и эффективности.
практика

рис. 6


Задание 2.3
План выполнения задания 2.3 (Git через терминал)
1. Подготовка среды: установка Git и настройка глобальных параметров пользователя.
2. Создание нового репозитория или клонирование существующего.
3. Добавление файлов в индекс (staging area).
4. Создание коммитов с описанием изменений.
5. Отправка изменений на удалённый репозиторий (push).
6. Получение изменений из удалённого репозитория (pull).
7. Работа с ветками: создание, переключение и слияние.
8. Просмотр состояния репозитория и истории коммитов.
9. Демонстрация выполненных действий через скриншоты терминала или текстовый отчёт.
10. Анализ особенностей работы через терминал, оценка удобства и эффективности.

рис. 7