

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ

по выполнению внеаудиторной работы при реализации программы дополнительного
общеобразовательного образования

**«Подготовка к аттестации за 11 класс для дистанционного семейного
обучения»**

г. Карабулак,

2026

ВВЕДЕНИЕ

Настоящие методические рекомендации разработаны для обучающихся 11-х классов, завершающих программу информатики. Этот год является кульминацией обучения, где акцент делается на глубоком понимании фундаментальных теоретических концепций, таких как теория систем, графы, деревья, а также на освоении продвинутых прикладных технологий, необходимых для успешной сдачи ЕГЭ, поступления в ВУЗы и начала профессиональной деятельности в IT-сфере. Программа охватывает углубленное программирование на Python (с рекурсией, обработкой файлов, словарями и множествами), продвинутые методы работы с искусственным интеллектом (машинное обучение, генеративный ИИ, Fine-Tuning), Big Data и язык SQL, профессиональное 3D-моделирование в Blender, а также критически важные аспекты кибербезопасности (криптография, OSINT, государственные электронные сервисы). Документ призван помочь обучающемуся успешно освоить эти сложные и актуальные темы, но и развить компетенции, необходимые для успешной сдачи ЕГЭ и профессиональной деятельности в IT-сфере.

Материал структурирован таким образом, чтобы обучающийся мог пройти путь от понимания принципов системного анализа и моделирования до создания сложных программных продуктов, анализа больших данных, разработки ИИ-решений, обеспечения кибербезопасности и создания профессиональных дизайн-проектов.

Цель документа

Основная цель рекомендаций — обеспечить глубокое понимание профильных теоретических основ информатики, сформировать устойчивые практические навыки в области продвинутого программирования на Python, работы с Big Data и SQL, кибербезопасности, продвинутого ИИ, 3D-моделирования, а также развить компетенции в UI/UX дизайне и подготовке к ЕГЭ:

1. Разъяснить принципы системного анализа, моделирования, теории графов и деревьев.
2. Сформировать глубокое понимание алгоритмов сжатия информации.
3. Обучить продвинутым аспектам программирования на Python, включая рекурсию, работу с файлами, словарями и множествами.

4. Развить навыки работы с Big Data и языком SQL для управления базами данных.
5. Обеспечить освоение принципов работы с искусственным интеллектом: машинное обучение, генеративный ИИ, Fine-Tuning, компьютерное зрение, промпт-инжиниринг.
6. Повысить уровень владения инструментами 3D-моделирования в Blender, включая скульптинг, системы частиц, экспорт/импорт моделей.
7. Сформировать компетенции в области кибербезопасности: информационная безопасность, цифровой след (OSINT), криптография, государственные электронные сервисы.
8. Развить навыки профессионального графического дизайна: композиция, типографика, дизайн-системы, айдентика бренда.
9. Подготовить к успешной реализации итоговых проектов, демонстрирующих комплексное применение полученных знаний и навыков, а также к успешной сдаче ЕГЭ по информатике.

Структура документа

Рекомендации разделены на ключевые блоки, соответствующие основным видам учебной деятельности в 11 классе:

1. Контроль знаний (тестирование).
2. Алгоритмический блок: Продвинутое программирование на Python: рекурсия, работа с файлами, словари, множества, алгоритмы и системы счисления.
3. Искусственный интеллект: машинное обучение, генеративный ИИ, промпт-инжиниринг, Fine-Tuning, компьютерное зрение, ИИ в робототехнике и ГИС.
4. Кибербезопасность: информационная безопасность, техногенные угрозы, цифровой след (OSINT), криптография, государственные электронные сервисы.
5. Графический дизайн и Веб-разработка.

Каждый раздел содержит:

1. Пошаговые алгоритмы действий («чек-листы»).
2. Типичные сценарии выполнения заданий с разбором нюансов.

3. Систему критериев для самопроверки.

Методологические основы

Рекомендации базируются на следующих принципах:

1. Профильно-ориентированный подход — углубленное изучение тем, необходимых для ЕГЭ и поступления в ВУЗы IT-направления.
2. Практико-ориентированный подход — обучение через создание реальных, комплексных цифровых продуктов и решение прикладных задач.
3. Междисциплинарность — интеграция знаний из математики (логика, системы счисления), программирования, дизайна, анализа данных и менеджмента проектов.
4. Актуальность технологий — использование современных инструментов и платформ (Python, HTML/CSS/JS, LLM, Blender, Git, облачные сервисы).
5. Развитие критического мышления — формирование навыков анализа информации, оценки эффективности ИИ-решений, выбора оптимальных технологий.
6. Самостоятельность и инициатива — поощрение экспериментов, поиска решений и творческого подхода к задачам, а также навыков самообучения

Целевая аудитория

Документ ориентирован на обучающихся 11 класса (для контроля процесса обучения) и педагогов, реализующих программы профильного обучения и подготовки к ЕГЭ по информатике.

Ожидаемые результаты реализации

Использование данных рекомендаций позволит:

1. Уверенно ориентироваться в теории систем, моделях и моделировании, графах и деревьях, алгоритмах сжатия информации;
2. Глубоко понимать и применять методы машинного обучения, генеративного ИИ, промпт-инжиниринга, Fine-Tuning и компьютерного зрения;
3. Создавать сложные программы на Python, эффективно используя рекурсию, работу с файлами, словари и множества;

4. Эффективно работать с Big Data и языком SQL для управления базами данных;
5. Получить продвинутые навыки 3D-моделирования в Blender, включая скульптинг, работу с частицами, экспорт/импорт моделей;
6. Развивать компетенции в области кибербезопасности: информационная безопасность, цифровой след (OSINT), криптография, государственные электронные сервисы;
7. Создавать профессиональные дизайн-проекты, включая айдентику бренда, дизайн-системы, вайрфреймы;
8. Разрабатывать веб-сайты с использованием HTML, CSS, JS, Tilda и генерации с ИИ;
9. Успешно реализовать итоговые, демонстрирующие комплексное применение всех полученных знаний и навыков;
10. Развивать самостоятельность в обучении;
11. Формировать ответственное отношение к учебной деятельности;
12. Повышать уровень подготовки к текущему и итоговому контролю.

Особенности применения

Рекомендации следует использовать:

1. Как пошаговое руководство при выполнении практических работ по программированию на Python, работе с Big Data и SQL, 3D-моделированию, кибербезопасности, веб-разработке и дизайну;
2. Как справочник при подготовке к контрольным срезам и тестам, включая теоретические вопросы по архитектуре ПК, сетям, БД, кодированию, системам счисления, алгебре логики;
3. Во время выполнения домашних заданий;
4. Для устранения пробелов в знаниях при самостоятельном изучении пропущенных тем.

Структура работы с документом

Для наиболее эффективного использования рекомендаций рекомендуется придерживаться следующего порядка работы:

1. Ознакомиться с теоретическим материалом по изучаемой теме.
2. Открыть соответствующий раздел данных рекомендаций.
3. Выполнить подготовительные действия (установка ПО, подготовка медиафайлов, создание аккаунта в онлайн-сервисе).
4. Следовать алгоритму выполнения задания, обращая внимание на советы по оформлению и технические нюансы.
5. Провести самоаудит работы по критериям оценивания.
6. В случае возникновения ошибок — использовать методы отладки.

1. ТЕСТИРОВАНИЕ

Цели проведения тестирования

Тестирование в информатике 11 класса направлено на оценку глубокого понимания фундаментальных теоретических основ, а также способности применять знания в контексте современных технологий и подготовки к ЕГЭ:

- Проверка понимания теории систем, моделей и моделирования, графов и деревьев.
- Оценка знаний по алгоритмам сжатия информации, Big Data, SQL, кибербезопасности (криптография, OSINT).
- Контроль владения терминологией (Машинное обучение, Генеративный ИИ, Fine-Tuning, OSINT, Криптография, SQL).
- Проверка умения анализировать информацию, включая критическую оценку ИИ-решений, эффективности веб-технологий и киберугроз.
- Формирование у обучающихся навыков самоконтроля и ответственности за результат, а также готовности к ЕГЭ.

Планируемые результаты

Обучающийся должен уметь:

- Разбираться в теории систем, компонентах и их взаимодействии, моделях и моделировании.
- Понимать принципы работы графов (виды, применение) и деревьев (бинарные, деревья поиска).
- Знать алгоритмы сжатия информации и их применение.
- Применять методы машинного обучения, генеративного ИИ, промпт-инжиниринга, Fine-Tuning и компьютерного зрения.
- Использовать Python для решения задач, включая рекурсию, работу с файлами, словари и множества.
- Работать с Big Data и языком SQL для управления базами данных.
- Понимать основы информационной безопасности, цифрового следа (OSINT) и криптографии.

- Понимать основные понятия ИИ: LLM, промптинг, Fine-Tuning, компьютерное зрение.
- Понимать базовые принципы работы 3D-моделирования в Blender.

Методические рекомендации

Тесты по информатике 11 класса требуют не только знания определений, но и умения применять теоретические знания для решения комплексных задач, часто с элементами анализа алгоритмов, логических выражений, SQL-запросов или расчетов. При выполнении следует:

- Внимательно читать вопросы: особенно те, что касаются кодирования, систем счисления, алгебры логики и анализа кода, где важна точность каждого шага;
- Анализировать фрагменты кода: если в вопросе есть фрагмент Python-кода или HTML/CSS/JS, попробуйте мысленно «выполнить» его, чтобы понять результат. Обращайте внимание на ООП-конструкции;
- Критически оценивать варианты: в вопросах по ИИ, сетевым технологиям или правовому регулированию часто встречаются правдоподобные, но неверные утверждения;
- Использовать логику и математику: для вопросов по алгебре логики, системам счисления, комбинаторике применяйте системное мышление и расчеты.

Все задания основаны исключительно на изученном учебном материале, поэтому перед выполнением тестовой работы рекомендуется повторить основные понятия, определения, формулы и правила по соответствующей теме.

Временной регламент

Стандартный тест из 10-15 вопросов рассчитан на 25–30 минут. Практические задания внутри теста могут занимать больше времени.

Пример структуры теста

Общее количество заданий — 10, разделенных на три тематических модуля. Ниже представлен детальный разбор каждого блока с примерами вопросов.

Блок 1. Теоретические основы информатики (3 задания):

- Теория систем и моделирование: Системы, компоненты и их взаимодействие, модели и моделирование (например, «Объясните, чем отличается компьютерная модель от математической»).
- Графы и деревья: Виды графов, деревья, бинарные деревья, деревья поиска (например, «Назовите основные свойства дерева как структуры данных»).
- Алгоритмы сжатия: Принципы работы алгоритмов сжатия информации (например, «Объясните принцип работы алгоритма сжатия Хаффмана»).

Блок 2. Искусственный интеллект и Big Data (2 задания):

- Искусственный интеллект: Большие языковые модели, нейросети, машинное обучение, компьютерное зрение, генеративный ИИ, промпт-инжиниринг, Fine-Tuning моделей, ИИ в робототехнике и ГИС (например, «В чем заключается разница между машинным обучением с педагогом и без педагога?»).
- Big Data и SQL: Понятие Big Data, СУБД, язык SQL (создание таблиц, выборка, фильтрация, сортировка данных) (например, «Напишите SQL-запрос для выборки всех пользователей из таблицы users, у которых возраст больше 18 лет»).

Блок 3. Программирование на Python (3-4 задания):

- Рекурсия: Понимание и применение рекурсивных функций (например, «Напишите рекурсивную функцию на Python для вычисления факториала числа»).
- Файлы и обработка данных: Чтение и запись данных в файлы, парсинг данных из файлов (например, «Напишите Python-код для подсчета количества слов в текстовом файле?»)
- Словари и множества: Их использование для эффективного хранения и обработки данных (например, «Объясните, когда целесообразно использовать множество вместо списка в Python»)
- Алгоритмы и системы счисления: Применение Python для реализации алгоритмов перевода чисел между системами счисления, решения задач по комбинаторике (например, «Напишите Python-функцию для перевода числа из десятичной системы в двоичную»).

Блок 4. Кибербезопасность и Современные технологии (2 задания):

– Информационная безопасность: Техногенные и экономические угрозы, государственные электронные сервисы (например, «Назовите основные принципы обеспечения информационной безопасности»).

– Цифровой след (OSINT): Понятие цифрового следа, методы его анализа (OSINT) и конфиденциальность (например, «Что такое OSINT и как он связан с цифровым следом?»).

Уровень	Описание	Пример задания
Базовый (30%)	Знание терминов, простых определений, базовых синтаксических конструкций Python, основ SQL, понятий ИИ и кибербезопасности.	Что такое граф?
Повышенный (50%)	Применение знаний в знакомой ситуации, анализ фрагментов кода Python/SQL, решение задач по графам, деревьям, алгоритмам сжатия, ИИ, кибербезопасности.	Напишите SQL-запрос для добавления новой записи в таблицу products.
Высокий (20%)	Анализ, синтез, решение проблем, критическая оценка, понимание сложных концепций и их применение, например, в контексте рекурсивных алгоритмов, Big Data, криптографии или продвинутых ИИ-решений	Разработайте рекурсивную функцию на Python для обхода бинарного дерева в глубину.

2. АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ (ПРОФИЛЬНЫЙ PYTHON)

Цели практических работ

В 11 классе мы углубляем наши знания Python, переходя к профильным концепциям, необходимым для создания сложных приложений, работы с данными и подготовки к ЕГЭ. Особое внимание уделяется эффективному использованию структур данных, объектно-ориентированному программированию и работе с системой контроля версий Git.

Планируемые результаты

- Продвинутый синтаксис: углубленное изучение переменных, чисел, операций, операторов сравнения и логических связок;
- Структуры данных: освоение работы со списками, кортежами, словарями и множествами, их методов и эффективного применения;
- Функции: создание и использование функций с различными типами аргументов, возвращаемыми значениями, рекурсия;
- Объектно-ориентированное программирование (ООП): понимание классов, объектов, наследования, полиморфизма и инкапсуляции;
- Система контроля версий Git: освоение базовых команд Git для командной разработки и управления версиями кода.

Основные концепции Python

Для успешного освоения профильного Python вы должны уверенно ориентироваться в следующих концепциях:

- Ввод-вывод данных: Эффективное использование `input()` и `print()`. Форматирование вывода (f-строки, метод `format()`);
- Переменные и типы данных: Понимание изменяемости и неизменяемости типов данных. Преобразование типов. Работа с Adele Code IDE;
- Базовые операторы и математика: Арифметические, логические, побитовые операторы. Приоритет операций;

- Условные конструкции (if/elif/else): Вложенные условия, тернарный оператор;
- Функции: Определение функций с параметрами по умолчанию, именованными аргументами (*args, **kwargs). Область видимости переменных (локальные, глобальные). Рекурсия.
- Классы и объекты: Определение классов, создание объектов, атрибуты и методы.
- Наследование: Создание дочерних классов, переопределение методов.
- Инкапсуляция: Скрытие внутренней реализации, использование геттеров и сеттеров.
- Полиморфизм: Использование одного интерфейса для разных типов данных.

Рекомендации по рефакторингу и отладке кода на Python

Рефакторинг — это процесс изменения внутренней структуры программы без изменения её внешнего поведения. Цель — улучшить читаемость, поддерживаемость и расширяемость кода.

Главные аспекты рефакторинга:

- Читаемость: Используйте осмысленные имена переменных и функций (например, `calculate_total_price` вместо `ctr`). Добавляйте комментарии для сложных участков кода;
- Функции: Разделяйте большие блоки кода на небольшие, специализированные функции. Каждая функция должна выполнять одну конкретную задачу;
- Устранение дублирования: Если один и тот же фрагмент кода встречается несколько раз, вынесите его в отдельную функцию.
- Оптимизация: Ищите более эффективные алгоритмы или встроенные функции Python для решения задач

Главные аспекты отладки кода:

- Сообщения об ошибках (Traceback): Внимательно читайте Traceback. Он указывает на тип ошибки (например, `SyntaxError`, `NameError`, `TypeError`) и строку, где она произошла.;
- `print()` для отладки: Вставляйте `print()` в разных местах кода, чтобы выводить значения переменных и отслеживать ход выполнения программы. Это самый простой и эффективный способ.;
- Пошаговое выполнение (отладчик): для более сложных случаев используйте встроенный отладчик Python (`pdb`) или отладочные инструменты в IDE (например, PyCharm, VS Code). Они позволяют выполнять код пошагово, просматривать значения переменных и устанавливать точки останова.
- Тестирование: Пишите небольшие тесты для отдельных функций, чтобы убедиться в их корректной работе

<i>Критерии оценивания работ</i>	
Работоспособность	программа запускается и выполняет заявленный функционал без ошибок.
Корректность кода	Python соблюден, нет логических ошибок.
Эффективность	код решает задачу оптимальным способом, без избыточных вычислений
Читаемость кода	код хорошо структурирован, используются осмысленные имена, есть комментарии, применен рефакторинг
Использование продвинутых конструкций	продемонстрировано умение применять функции, списки, строки, циклы с <code>break/continue</code> .?
Отладка	продемонстрировано умение находить и исправлять ошибки в коде

3. ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ: МЕТОДЫ И ПРИМЕНЕНИЕ

В 11 классе мы углубляем наше понимание искусственного интеллекта, фокусируясь на его практическом применении в различных областях, таких как машинное обучение, генеративный ИИ, компьютерное зрение, робототехника и ГИС. Особое внимание уделяется продвинутым техникам промпт-инжиниринга и Fine-Tuning моделей.

Цели и задачи раздела:

- Большие языковые модели (LLM) и Нейросети: углубленное понимание архитектуры, принципов работы и возможностей современных LLM и нейросетей;
- Машинное обучение: изучение основных парадигм (с педагогом, без педагога, с подкреплением), алгоритмов (линейная регрессия, классификация, кластеризация) и их применения;
- Компьютерное зрение: продвинутое изучение задач (распознавание объектов, лиц, анализ изображений, сегментация) и применение в реальных проектах;
- Искусственный генеративный интеллект: понимание принципов работы генеративных моделей (GAN, Diffusion Models) для создания изображений и текста.
- Промпт-инжиниринг: освоение продвинутых техник составления эффективных промптов для LLM, включая Chain-of-Thought, Few-shot Learning, Persona-based prompting, а также понимание структуры промпта.
- Fine-Tuning моделей ИИ: понимание концепции адаптации предобученных моделей под специфические задачи и данные, практическое применение.
- ГИС с помощью ИИ: использование ИИ для анализа геопространственных данных, прогнозирования, создания карт.

Рекомендации по выполнению работ

- Понимание LLM и Нейросетей: Изучите, как работают нейронные сети, их слои, активационные функции. Поймите, как LLM обрабатывают естественный язык и генерируют ответы.

- Компьютерное зрение: углубите знания по задачам компьютерного зрения. Изучите библиотеки для работы с изображениями (например, OpenCV) и их применение для распознавания объектов, лиц, анализа видеопотока.
- Генеративный ИИ: Поймите, как генеративные модели создают новый контент. Экспериментируйте с генерацией изображений, текста, музыки, используя доступные онлайн-инструменты.
- Fine-Tuning моделей ИИ: Поймите, как дообучать предобученные модели на своих данных для достижения лучших результатов в специфических задачах. Изучите доступные платформы и инструменты для Fine-Tuning.
- ИИ в Робототехнике: Изучите, как ИИ используется для автономной навигации роботов, распознавания объектов, принятия решений в реальном времени. Рассмотрите примеры промышленных и сервисных роботов.
- ГИС с помощью ИИ: Поймите, как ИИ анализирует геопространственные данные для прогнозирования изменений климата, анализа городского планирования, оптимизации логистики).

<i>Критерии оценивания работ</i>	
Эффективность промптинга	Насколько качественно и детально были составлены промпты для ИИ, продемонстрировано ли владение продвинутыми техниками?
Интеграция ИИ	Насколько глубоко и осмысленно ИИ был интегрирован в проект (анализ данных, генерация кода, дизайна, текста, логики, компьютерное зрение).
Доработка и критический анализ	Насколько качественно вы доработали сгенерированный ИИ контент и критически оценили его результаты?
Этика ИИ	Соблюдены ли этические нормы при использовании ИИ, указано ли его применение, осознаны ли потенциальные риски и ограничения?

4. КИБЕРБЕЗОПАСНОСТЬ

В 11 классе кибербезопасность становится критически важным разделом, фокусируясь на понимании угроз, методов защиты, цифрового следа и криптографии. Особое внимание уделяется государственным электронным сервисам и конфиденциальности данных.

Цели и задачи раздела

- Информационная безопасность: углубленное понимание принципов, угроз и методов обеспечения информационной безопасности;
- Техногенные и экономические угрозы: изучение современных угроз, связанных с технологиями (фишинг, вредоносное ПО, DDoS-атаки, социальная инженерия);
- Государственные электронные сервисы: изучение принципов работы и безопасности государственных электронных сервисов (Госуслуги, налоговые порталы);
- Цифровой след (OSINT) и конфиденциальность: понимание понятия цифрового следа, методов его анализа (OSINT) и способов защиты конфиденциальности в сети;
- Криптография: изучение основ криптографии, симметричных и асимметричных алгоритмов шифрования, хеширования, цифровой подписи.

Рекомендации по кибербезопасности

- Изучите основные принципы информационной безопасности: конфиденциальность, целостность, доступность. Поймите, как они реализуются на практике.
- Разберитесь в различных видах угроз: фишинг, вредоносное ПО (вирусы, трояны, шифровальщики), DDoS-атаки, социальная инженерия. Изучите методы защиты от них.

Цифровой след (OSINT)

- Изучите, как методы OSINT используются для сбора и анализа информации из открытых источников. Изучите инструменты OSINT.

– Освойте методы защиты своей конфиденциальности в сети: настройка приватности в соцсетях и браузеров).

Криптография

– Изучите основные понятия криптографии: шифрование, дешифрование, ключ, хеширование, цифровая подпись.

– Разберитесь в принципах работы симметричных (AES) и асимметричных (RSA) алгоритмов шифрования. Поймите, для чего используется хеширование (MD5, SHA-256) и цифровая подпись.

<i>Критерии оценки работ по кибербезопасности</i>	
Понимание угроз	Насколько глубоко понимаются современные киберугрозы и методы защиты от них?
Знание принципов ИБ	Понимание конфиденциальности, целостности, доступности.
OSINT и конфиденциальность	Умение анализировать цифровой след и применять методы защиты конфиденциальности.
Криптография	Понимание основных криптографических алгоритмов и их применения
Практическое применение	Умение применять полученные знания для обеспечения личной и корпоративной кибербезопасности.

5. ГРАФИЧЕСКИЙ ДИЗАЙН И ВЕБ-РАЗРАБОТКА

В 11 классе графический дизайн выходит на профессиональный уровень, фокусируясь на создании айдентики бренда, дизайн-систем, а также на продвинутом веб- и мобильном дизайне с использованием ИИ. Особое внимание уделяется композиции, типографике и проектированию.

Цели и задачи раздела

- Графический дизайн. Основы: углубленное изучение принципов дизайна, его роли в современном мире.
- Основы композиции и типографика: продвинутое изучение правил композиции (ритм, баланс, контраст), типографики (шрифтовые пары, иерархия, читаемость).
- Проектирование. Вайрфреймы: освоение методов проектирования интерфейсов, создание вайрфреймов и прототипов.
- Генерация дизайна с помощью ИИ: использование ИИ для создания макетов, цветовых схем, элементов интерфейса.
- Дизайн-система: понимание концепции дизайн-систем, их компонентов и преимуществ для масштабируемой разработки.
- Дизайн-презентаций: создание профессиональных, интерактивных презентаций, соответствующих айдентике бренда.
- Практикум: Дизайн «айдентики» бренда, дизайн сайта и приложения с помощью ИИ. Вёрстка сайта и рефакторинг работы.

Рекомендации по графическому дизайну

- Основы работы: углубите знания по работе с фигурами, изображениями, слоями. Освойте маски, режимы наложения, эффекты.
- Цвет и Типографика: Изучите теорию цвета (цветовые модели, гармонии, контрасты) и типографику (шрифтовые пары, иерархия, читаемость) для создания профессионального дизайна.

- Веб-дизайн: Создавайте макеты веб-сайтов, учитывая пользовательский опыт, навигацию, структуру контента.
- Мобильный дизайн: Разрабатывайте дизайн, который будет хорошо выглядеть и удобно использоваться на мобильных устройствах. Учитывайте особенности сенсорного ввода и минимализм.
- Вайрфреймы и прототипы: Создавайте детальные вайрфреймы и интерактивные прототипы, которые имитируют работу реального приложения или сайта. Это позволяет тестировать пользовательский опыт и собирать обратную связь.
- Генерация дизайна с помощью ИИ: Используйте ИИ-инструменты для быстрого создания идей дизайна, цветовых палитр, шрифтовых пар или даже целых макетов мобильных приложений и сайтов. Дорабатывайте их вручную, применяя свои знания UI/UX.

Создание изображений с помощью ИИ

- Продвинутый промптинг: Применяйте техники промпт-инжиниринга для генерации высококачественных изображений, указывая стиль, детали, композицию, освещение.
- Редактирование с ИИ: Используйте ИИ для редактирования изображений (удаление объектов, изменение фона, стилизация, масштабирование).

Рекомендации по веб-разработке

- Используйте семантические теги HTML5 для создания логичной и доступной структуры страницы. Особое внимание уделите атрибутам aria- для улучшения доступности.
- Flexbox и Grid Layout: уверенно используйте эти инструменты для создания сложных, адаптивных макетов. Комбинируйте их для достижения максимальной гибкости.
- Анимации и Переходы: Создавайте сложные CSS-анимации и переходы для улучшения пользовательского опыта и привлечения внимания.

<i>Критерии оценки работ по графическому дизайну и веб-разработке</i>	
Комплексность	Насколько полно и разнообразно использованы изученные технологии (Python, веб-разработка, ИИ, дизайн, 3D-моделирование, БД/SQL, кибербезопасность, Git)?
Оригинальность	Насколько уникальна и интересна задумка проекта?
Техническая реализация	Качество кода, дизайна, функциональность, отсутствие ошибок. Соблюдение стандартов (PEP 8, семантика HTML, SQL-стандарты).
UI/UX	Удобство использования, эстетика, логичность интерфейса, адаптивность. Соответствие айдентике бренда.
Использование ИИ	Эффективность применения ИИ для генерации или доработки контента.

ЗАКЛЮЧЕНИЕ

- 1. Постоянное обучение:** Мир IT меняется очень быстро. Регулярно изучайте новые технологии, языки программирования, инструменты и тренды в веб-разработке, ИИ и мультимедиа.
- 2. Практика — это ключ к успеху:** Теория важна, но без постоянной практики в кодировании, дизайне, видеомонтаже и промптинге вы не сможете освоить материал. Создавайте собственные проекты!
- 3. Дебаггинг ваш лучший друг:** Ошибки в коде или дизайне это нормально. Умение находить, анализировать и исправлять их (дебаггинг) — один из самых ценных навыков программиста и разработчика.
- 4. Сначала логика, потом дизайн:** сначала добейтесь того, чтобы программа или таблица работала правильно, и только потом приступайте к украшению, подбору шрифтов и картинок.
- 5. Чистота и структура кода:** Пишите аккуратный, читаемый Python код. Используйте комментарии, осмысленные имена переменных и функций. Это облегчит работу вам и другим разработчикам.
- 6. Здоровье прежде всего:** Правильная осанка за компьютером — залог отсутствия усталости.
- 7. Умение искать ответы:** если программа выдает ошибку или вы не знаете, как сделать действие, используйте поиск в интернете или встроенную справку. Умение находить решение — это главный навык айтишника.
- 8. Эффективный промптинг:** научитесь четко и точно формулировать свои мысли для ИИ. Это навык будущего, который позволит вам максимально использовать потенциал искусственного интеллекта.
- 9. Этика общения:** помните о правилах сетевого этикета (нетикета) при переписке в почте или работе в общих документах. Уважайте чужой труд и личное пространство.
- 10. Оптимизация ресурсов:** при создании веб-сайтов и видеороликов думайте об оптимизации изображений, видео и кода для быстрой загрузки и эффективной работы.