

**Индивидуальный предприниматель
ХРАБРОВ РОМАН ИГОРЕВИЧ**



УТВЕРЖДАЮ

**Индивидуальный предприниматель
/Храбров Р.И./**

и ю л я 20 26 г.

**ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ ПРОГРАММА
ДОПОЛНИТЕЛЬНАЯ ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА
«ПРОГРАММИРОВАНИЕ ДЛЯ ДЕТЕЙ 13–16 ЛЕТ»**

Возраст обучающихся: 13–16 лет

Срок реализации: 1 учебный год

Объем программы: 144 академических часа

Направленность программы: техническая

Место реализации программы: г. Королёв, пр. Космонавтов, 20А, пом. 289

г. Королёв, 2026 г.

Оглавление

1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА	3
1.1. Общая характеристика программы	3
1.2. Актуальность программы	3
1.3. Адресат программы	4
1.4. Срок освоения, объем и режим занятий	4
1.5. Цель и задачи программы	4
1.6. Планируемые результаты обучения	5
2. УЧЕБНЫЙ ПЛАН ПРОГРАММЫ	6
2.1. УЧЕБНЫЙ ПЛАН	6
2.2. УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН	7
3. КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК	9
4. РАБОЧАЯ ПРОГРАММА	12
Раздел 1. Системы, виртуальные машины и ИИ-помощник — 20 часов	12
Раздел 2. Программирование на Python и ИИ-помощник — 32 часа	13
Раздел 3. Интернет вещей (IoT) — 16 часов	15
Раздел 4. Цифровой рисунок, Krita и нейросети — 24 часа	16
Раздел 5. Векторная графика, интерфейсы и ИИ — 20 часов	17
Раздел 6. 3D-моделирование в Blender и нейросети — 24 часа	18
Раздел 7. Комплексное портфолио и итоговая аттестация — 8 часов	19
5. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ	20
5.1. Контроль знаний, умений и навыков	20
5.2. Критерии оценивания освоения программы	21
6. ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ	21
6.1. Материально-техническое обеспечение	21
6.2. Кадровое обеспечение	22
7. УЧЕБНО-МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ	23
7.1. Информационные и учебно-методические материалы	23
7.2. Используемое программное обеспечение и цифровые ресурсы	23
7.3. Методы и технологии, использованные при составлении программы	23
7.4. Литература и источники, использованные при составлении программы	24
8. ПРИЛОЖЕНИЯ	25
Приложение 1. Примерные вопросы для контроля знаний	25
Приложение 2. Примерные практические задания	26

1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

1.1. Общая характеристика программы

Настоящая дополнительная общеобразовательная общеразвивающая программа «Программирование для детей 13–16 лет» представляет собой комплекс основных характеристик образования: объем, содержание, планируемые результаты, организационно-педагогические условия, формы контроля, учебный план, календарный учебный график, рабочую программу, оценочные и учебно-методические материалы.

Программа имеет техническую направленность и предназначена для практического знакомства подростков с несколькими профессиональными областями ИТ. Содержание объединяет инженерный трек — системы, виртуализацию, Python и интернет вещей — и дизайнерский трек — цифровой рисунок, интерфейсы и 3D-моделирование.

Программа разработана на основе параметров и тематических модулей годовой программы Senior Детской компьютерной академии КЛИК: 144 академических часа за учебный год, 16 академических часов в месяц, занятия один раз в неделю по 4 академических часа. Шесть модулей сайта составляют 136 академических часов; оставшиеся 8 часов отведены на сборку комплексного портфолио и итоговую защиту.

Образовательный процесс имеет практико-ориентированный, проектный и профориентационный характер. Обучающиеся настраивают учебную ИТ-лабораторию, пишут программы, создают IoT-устройство, цифровые иллюстрации, UI-макеты и 3D-сцены, документируют ход работы и представляют результаты.

Искусственный интеллект используется только как вспомогательный инструмент: для объяснения терминов и кода, поиска возможных причин ошибок, генерации идей и референсов. Все рекомендации и материалы должны быть проверены, поняты и адаптированы обучающимся. Особое внимание уделяется конфиденциальности, авторству и фиксации самостоятельного вклада.

Итогом обучения является комплексное цифровое портфолио. Обучающийся представляет работы по обоим трекам и защищает один основной проект, наиболее соответствующий его интересам и предполагаемой дальнейшей образовательной траектории.

Программа разработана на основе следующих нормативных документов:

Федеральный закон от 29 декабря 2012 г. № 273-ФЗ «Об образовании в Российской Федерации»;

Приказ Министерства просвещения Российской Федерации от 27 июля 2022 г. № 629 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»;

Постановление Главного государственного санитарного врача Российской Федерации от 28 сентября 2020 г. № 28 «Об утверждении санитарных правил СП 2.4.3648-20 “Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи”»;

Распоряжение Правительства Российской Федерации от 31 марта 2022 г. № 678-р «Об утверждении Концепции развития дополнительного образования детей до 2030 года»;

локальные нормативные акты организации, регламентирующие образовательную деятельность.

Направленность (профиль) программы: техническая.

1.2. Актуальность программы

Актуальность программы обусловлена необходимостью осознанной профессиональной ориентации подростков в условиях быстрого развития цифровых технологий. В возрасте 13–16 лет обучающемуся важно не только освоить отдельную программу, но и попробовать разные виды профессиональной деятельности: системное администрирование, программирование, работу с устройствами, цифровой рисунок, проектирование интерфейсов и трехмерную графику.

Комплексное содержание позволяет увидеть взаимосвязь аппаратного обеспечения, операционных систем, программного кода, сетей, пользовательского интерфейса и визуальной коммуникации. Это помогает сделать дальнейший выбор профильного направления более обоснованным.

Отличительной особенностью программы является сочетание инженерного и дизайнерского треков, реальных проектных задач и критического применения искусственного интеллекта. Обучающиеся не получают готовые решения, а учатся ставить задачу, проверять предложения ИИ, устранять ошибки и отвечать за качество конечного результата.

Программа развивает алгоритмическое и системное мышление, визуальную культуру, самостоятельность, тайм-менеджмент, коммуникацию, настойчивость и умение публично представлять результат — навыки, значимые как для дальнейшего ИТ-обучения, так и для общего образования.

1.3. Адресат программы

Программа предназначена для обучающихся в возрасте 13–16 лет, проявляющих интерес к программированию, устройству компьютеров, умным устройствам, цифровому рисунку, дизайну интерфейсов, трехмерному моделированию и созданию собственных проектов.

Специальная предварительная подготовка не требуется. Для успешного освоения программы желательно владение базовыми навыками работы с компьютером, файлами и браузером, а также готовность читать технические инструкции, работать с англоязычными названиями команд и самостоятельно исправлять ошибки.

Программа учитывает возрастные особенности подростков: стремление к самостоятельности и профессиональному самоопределению, способность к абстрактному мышлению, интерес к практическому результату, потребность в признании собственного вклада, критическом обсуждении решений и проектной работе.

1.4. Срок освоения, объем и режим занятий

Срок реализации программы: 1 учебный год.

Продолжительность реализации: 9 месяцев.

Объем программы: 144 академических часа.

Режим занятий: 1 раз в неделю по 4 академических часа.

Объем занятий в месяц: 16 академических часов.

Продолжительность академического часа составляет 40 минут. Допускается продолжительность 35–40 минут с учетом расписания и санитарных требований. Между академическими часами предусматриваются перерывы.

Занятия проводятся в групповой форме. Практические задания могут выполняться индивидуально, в парах или малых проектных группах. Каждый обучающийся сохраняет личные рабочие файлы, исходный код, промежуточные версии и материалы для портфолио.

Занятия начинаются не ранее 9:00 и заканчиваются не позднее 20:00. При организации образовательного процесса соблюдаются санитарно-эпидемиологические требования, правила электрической безопасности, требования безопасной работы с компьютерной техникой, микроконтроллерами и сетевыми сервисами.

Адрес места осуществления образовательной деятельности: г. Королёв, пр. Космонавтов, 20А, пом. 289.

1.5. Цель и задачи программы

Цель программы — формирование у обучающихся 13–16 лет комплекса инженерных, программных, дизайнерских и проектных компетенций, необходимых для осознанного выбора дальнейшей траектории в ИТ-сфере и создания первых работ профессионального портфолио.

Задачи программы:

1. Сформировать представление об устройстве компьютера, операционных системах, виртуализации и базовых сетевых технологиях.
2. Научить создавать и безопасно использовать учебные виртуальные машины с Windows и Linux.
3. Сформировать навыки работы с командной строкой Linux и базовой диагностики систем и сети.
4. Дать системные основы программирования на Python: переменные, типы данных, условия, циклы, коллекции и функции.
5. Научить проектировать, тестировать, отлаживать и защищать собственное приложение на Python.
6. Познакомить с микроконтроллерами, сенсорами, MicroPython, Wi-Fi и передачей данных на сервер.
7. Научить создавать цифровые скетчи, персонажей, фоны и концепт-арт в Krita.
8. Сформировать понимание цвета, света, композиции, визуального стиля и подготовки работ для портфолио.
9. Дать основы векторной графики и проектирования пользовательских интерфейсов в Figma.
10. Научить создавать компоненты, экраны и кликабельные прототипы сайтов или приложений.
11. Сформировать навыки полигонального 3D-моделирования, работы с материалами, светом, камерой и рендером в Blender.
12. Научить критически и безопасно применять ИИ для объяснения, поиска ошибок, идей и референсов с обязательной проверкой результата.
13. Развить системное, алгоритмическое, инженерное, пространственное и творческое мышление.
14. Сформировать навыки планирования, документирования, командного обсуждения, презентации и защиты проекта.

1.6. Планируемые результаты обучения

По итогам освоения программы обучающиеся должны:

Знать:

- назначение основных компонентов компьютера и факторы, влияющие на производительность;
- назначение операционной системы, файловой системы, учетных записей и прав доступа;
- принципы виртуализации, сетевого взаимодействия и безопасной установки программ;
- основы синтаксиса Python, типы данных, условия, циклы, списки и функции;
- основные этапы разработки, тестирования, отладки и рефакторинга программы;
- назначение микроконтроллера, сенсора, MicroPython, Wi-Fi и клиент-серверной передачи данных;
- основы цифрового рисунка: слои, кисти, цвет, свет, композиция и концепт-арт;
- различия растровой и векторной графики, принципы UI-дизайна, сеток и компонентов;
- основы трехмерной сцены, полигонального моделирования, материалов, света, камеры и рендера;
- возможности, ограничения и риски применения генеративного ИИ в технических и дизайнерских задачах;
- правила авторства, конфиденциальности и фиксации самостоятельного вклада;
- структуру портфолио и требования к публичной защите проекта.

Уметь:

- определять назначение компонентов компьютера и выполнять базовую настройку системы;

- создавать виртуальную машину, устанавливать учебную операционную систему и сохранять снимок состояния;
- выполнять базовые команды Linux и диагностировать простые проблемы подключения;
- писать программы на Python с вводом, вычислениями, условиями, циклами, списками и функциями;
- составлять тест-кейсы, находить и исправлять синтаксические и логические ошибки;
- подключать датчик к микроконтроллеру, считывать данные и передавать их по Wi-Fi;
- создавать в Krita скетчи, персонажей, фоны и завершённые цифровые иллюстрации;
- создавать векторные элементы, компоненты, экраны и интерактивный прототип в Figma;
- моделировать объекты и сцены в Blender, назначать материалы, настраивать свет и выполнять рендер;
- формулировать точный запрос к ИИ, проверять ответ, объяснять и адаптировать предложенное решение;
- документировать процесс, сохранять промежуточные версии и оформлять проект для портфолио;
- представлять результат, обосновывать решения и отвечать на вопросы.

Владеть навыками:

- системного и алгоритмического анализа технической задачи;
- работы с Windows, Linux, виртуальной машиной, Python, MicroPython, Krita, Figma и Blender;
- создания программного приложения, IoT-прототипа, цифровой иллюстрации, UI-макета и 3D-сцены;
- безопасной работы с оборудованием, сетевыми ресурсами и искусственным интеллектом;
- самостоятельного поиска, проверки и исправления ошибок;
- проектного планирования, коммуникации, презентации и профессиональной рефлексии.

2. УЧЕБНЫЙ ПЛАН ПРОГРАММЫ

2.1. УЧЕБНЫЙ ПЛАН

В процессе преподавания используются объяснение, демонстрация, лабораторная и практическая работа, программирование, моделирование, проектная деятельность, анализ кода и визуальных решений, тестирование, взаимопроверка, работа в парах и малых группах, консультация с ИИ-помощником под контролем преподавателя, обсуждение результатов и публичная защита.

№	Наименование разделов и уроков	Всего часов	Теория	Практика	Форма аттестации / контроля
1	Раздел 1. Системы, виртуальные машины и ИИ-помощник	20	5	15	Практические задания, защита виртуальной IT-лаборатории
2	Раздел 2. Программирование на Python и ИИ-помощник	32	8	24	Практические задания, защита программы на Python
3	Раздел 3. Интернет вещей (IoT)	16	4	12	Практические задания, демонстрация умной метеостанции
4	Раздел 4. Цифровой рисунок, Krita и нейросети	24	6	18	Практические задания, презентация серии цифровых работ
5	Раздел 5. Векторная графика, интерфейсы и ИИ	20	5	15	Практические задания, защита UI-макета
6	Раздел 6. 3D-моделирование в Blender и нейросети	24	6	18	Практические задания, защита 3D-проекта

№	Наименование разделов и уроков	Всего часов	Теория	Практика	Форма аттестации / контроля
7	Раздел 7. Комплексное портфолио и итоговая аттестация	8	2	6	Подготовка портфолио, итоговая защита проекта
	Итого часов:	144	36	108	

2.2. УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН

№	Наименование разделов и уроков	Всего часов	Теория	Практика	Форма аттестации / контроля
1	Раздел 1. Системы, виртуальные машины и ИИ-помощник	20	5	15	Практические задания, защита виртуальной IT-лаборатории
1.1	Урок 1. Устройство компьютера и системное мышление	4	1	3	Практическое задание
1.2	Урок 2. Операционные системы Windows и Linux	4	1	3	Практическое задание
1.3	Урок 3. Виртуальные машины и учебная IT-лаборатория	4	1	3	Практическое задание
1.4	Урок 4. Командная строка Linux и системные операции	4	1	3	Практическое задание
1.5	Урок 5. Компьютерные сети, диагностика и защита лаборатории	4	1	3	Практическое задание
2	Раздел 2. Программирование на Python и ИИ-помощник	32	8	24	Практические задания, защита программы на Python
2.1	Урок 6. Python: среда разработки, синтаксис и первая программа	4	1	3	Практическое задание
2.2	Урок 7. Переменные, типы данных и пользовательский ввод	4	1	3	Практическое задание
2.3	Урок 8. Условия и логические выражения	4	1	3	Практическое задание
2.4	Урок 9. Циклы и алгоритмы повторения	4	1	3	Практическое задание
2.5	Урок 10. Строки, списки и обработка данных	4	1	3	Практическое задание
2.6	Урок 11. Функции и декомпозиция программы	4	1	3	Практическое задание
2.7	Урок 12. Проектное приложение и ИИ-помощник программиста	4	1	3	Практическое задание
2.8	Урок 13. Тестирование, рефакторинг и защита программы	4	1	3	Практическое задание
3	Раздел 3. Интернет вещей (IoT)	16	4	12	Практические задания, демонстрация умной метеостанции
3.1	Урок 14. Интернет вещей, микроконтроллеры и электрическая безопасность	4	1	3	Практическое задание
3.2	Урок 15. MicroPython и работа с датчиками	4	1	3	Практическое задание
3.3	Урок 16. Wi-Fi и отправка данных на сервер	4	1	3	Практическое задание
3.4	Урок 17. Сборка, тестирование и защита умной метеостанции	4	1	3	Практическое задание
4	Раздел 4. Цифровой рисунок, Krita и нейросети	24	6	18	Практические задания, презентация серии цифровых работ
4.1	Урок 18. Krita: рабочее пространство, кисти и цифровой скетч	4	1	3	Практическое задание
4.2	Урок 19. Слои, выделения, маски и чистовой рисунок	4	1	3	Практическое задание
4.3	Урок 20. Цвет, свет, тень и объем	4	1	3	Практическое задание
4.4	Урок 21. Композиция, персонаж и визуальный стиль	4	1	3	Практическое задание
4.5	Урок 22. Фон и концепт-арт сцены	4	1	3	Практическое задание
4.6	Урок 23. Нейросети в работе художника и портфолио	4	1	3	Практическое задание
5	Раздел 5. Векторная графика, интерфейсы и ИИ	20	5	15	Практические задания, защита UI-макета
5.1	Урок 24. Векторная графика и компоненты интерфейса	4	1	3	Практическое задание
5.2	Урок 25. Figma: фреймы, сетки, стили и компоненты	4	1	3	Практическое задание
5.3	Урок 26. Проектирование экранов и пользовательских сценариев	4	1	3	Практическое задание
5.4	Урок 27. UI-макет и проверка удобства	4	1	3	Практическое задание
5.5	Урок 28. ИИ для идей, стиля и промо-визуалов	4	1	3	Практическое задание
6	Раздел 6. 3D-моделирование в Blender и нейросети	24	6	18	Практические задания, защита 3D-проекта
6.1	Урок 29. Blender: интерфейс, навигация и трансформации	4	1	3	Практическое задание
6.2	Урок 30. Полигональное моделирование и форма объекта	4	1	3	Практическое задание
6.3	Урок 31. Материалы, текстуры и визуальные свойства	4	1	3	Практическое задание
6.4	Урок 32. Свет, камера и композиция 3D-сцены	4	1	3	Практическое задание

№	Наименование разделов и уроков	Всего часов	Теория	Практика	Форма аттестации / контроля
6.5	Урок 33. Рендер, оптимизация и сборка финальной сцены	4	1	3	Практическое задание
6.6	Урок 34. Нейросети для концептов и защита 3D-проекта	4	1	3	Практическое задание
7	Раздел 7. Комплексное портфолио и итоговая аттестация	8	2	6	Подготовка портфолио, итоговая защита проекта
7.1	Урок 35. Сборка комплексного IT-портфолио	4	1	3	Подготовка портфолио
7.2	Урок 36. Итоговая защита проекта и портфолио	4	1	3	Итоговая защита проекта
	Итого часов:	144	36	108	

3. КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК

Количество учебных часов: 144 часа. Занятия по программе проводятся в течение 9 месяцев.

Учебная нагрузка распределяется равномерно: 16 академических часов в месяц, четыре занятия по 4 академических часа.

№ п/п	Уроки	1 месяц	2 месяц	3 месяц	4 месяц	5 месяц	6 месяц	7 месяц	8 месяц	9 месяц
1	Раздел 1. Урок 1. Устройство компьютера и системное мышление	4								
2	Раздел 1. Урок 2. Операционные системы Windows и Linux	4								
3	Раздел 1. Урок 3. Виртуальные машины и учебная IT-лаборатория	4								
4	Раздел 1. Урок 4. Командная строка Linux и системные операции	4								
5	Раздел 1. Урок 5. Компьютерные сети, диагностика и защита лабораторий		4							
6	Раздел 2. Урок 6. Python: среда разработки, синтаксис и первая программа		4							
7	Раздел 2. Урок 7. Переменные, типы данных и пользовательский ввод		4							
8	Раздел 2. Урок 8. Условия и логические выражения		4							
9	Раздел 2. Урок 9. Циклы и алгоритмы повторения			4						
10	Раздел 2. Урок 10. Строки, списки и обработка данных			4						
11	Раздел 2. Урок 11. Функции и декомпозиция программы			4						
12	Раздел 2. Урок 12. Проектное приложение и ИИ-помощник программиста			4						
13	Раздел 2. Урок 13. Тестирование, рефакторинг и защита программы				4					
14	Раздел 3. Урок 14. Интернет вещей, микроконтроллеры и электрическая безопасность				4					
15	Раздел 3. Урок 15. MicroPython и работа с датчиками				4					
16	Раздел 3. Урок 16. Wi-Fi и отправка данных на сервер				4					
17	Раздел 3. Урок 17. Сборка, тестирование и защита умной метеостанции					4				
18	Раздел 4. Урок 18. Krita: рабочее пространство, кисти и цифровой скетч					4				
19	Раздел 4. Урок 19. Слои, выделения, маски и чистовой рисунок					4				
20	Раздел 4. Урок 20. Цвет, свет, тень и объем					4				
21	Раздел 4. Урок 21. Композиция, персонаж и визуальный стиль						4			
22	Раздел 4. Урок 22. Фон и концепт-арт сцены						4			
23	Раздел 4. Урок 23. Нейросети в работе художника и портфолио						4			
24	Раздел 5. Урок 24. Векторная графика и компоненты интерфейса						4			
25	Раздел 5. Урок 25. Figma: фреймы, сетки, стили и компоненты							4		
26	Раздел 5. Урок 26. Проектирование экранов и пользовательских сценариев							4		
27	Раздел 5. Урок 27. UI-макет и проверка удобства							4		
28	Раздел 5. Урок 28. ИИ для идей, стиля и промо-визуалов							4		
29	Раздел 6. Урок 29. Blender: интерфейс, навигация и трансформации								4	
30	Раздел 6. Урок 30. Полигональное моделирование и форма объекта								4	
31	Раздел 6. Урок 31. Материалы, текстуры и визуальные свойства								4	
32	Раздел 6. Урок 32. Свет, камера и композиция 3D-сцены								4	

№ п/п	Уроки	1 месяц	2 месяц	3 месяц	4 месяц	5 месяц	6 месяц	7 месяц	8 месяц	9 месяц
33	Раздел 6. Урок 33. Рендер, оптимизация и сборка финальной сцены									4
34	Раздел 6. Урок 34. Нейросети для концептов и защита 3D-проекта									4
35	Раздел 7. Урок 35. Сборка комплексного IT-портфолио									4
36	Раздел 7. Урок 36. Итоговая защита проекта и портфолио									4
	ИТОГО	16	16	16	16	16	16	16	16	16

4. РАБОЧАЯ ПРОГРАММА

Раздел 1. Системы, виртуальные машины и ИИ-помощник — 20 часов

Урок 1. Устройство компьютера и системное мышление

Содержание:

- архитектура персонального компьютера и взаимодействие компонентов;
- процессор, оперативная память, накопители, видеосистема и периферия;
- параметры оборудования и их влияние на производительность;
- правила безопасной работы с компьютерной техникой;
- использование ИИ для объяснения технических терминов с проверкой ответа.

Практическое задание: Составить схему компонентов компьютера, определить назначение каждого элемента и подготовить краткую спецификацию учебного рабочего места.

Урок 2. Операционные системы Windows и Linux

Содержание:

- назначение операционной системы;
- файловые системы, пользователи, права доступа и учетные записи;
- сходства и различия Windows и Linux;
- установка и удаление прикладных программ;
- обновления, резервное копирование и восстановление.

Практическое задание: Выполнить базовую настройку учебной системы, создать пользователей и подготовить сравнительную таблицу Windows и Linux.

Урок 3. Виртуальные машины и учебная IT-лаборатория

Содержание:

- понятие виртуализации и назначение гипервизора;
- виртуальная машина, образ системы, снимок состояния и изоляция;
- выделение оперативной памяти, процессоров и дискового пространства;
- установка гостевой операционной системы;
- правила безопасной работы с виртуальной средой.

Практическое задание: Создать виртуальную машину, установить учебную операционную систему и сохранить снимок настроенного состояния.

Урок 4. Командная строка Linux и системные операции

Содержание:

- терминал и структура команды;
- навигация по каталогам;
- создание, копирование, перемещение и удаление файлов;
- просмотр содержимого и получение справки;
- права доступа и осторожность при выполнении системных команд;
- ИИ как помощник при разборе команд и сообщений об ошибках.

Практическое задание: Выполнить набор команд в терминале, оформить журнал команд и объяснить назначение каждой операции.

Урок 5. Компьютерные сети, диагностика и защита лаборатории

Содержание:

- локальная и глобальная сеть;
- IP-адрес, маршрутизатор, DNS и клиент-серверное взаимодействие;
- подключение виртуальной машины к сети;
- базовые средства диагностики соединения;
- безопасная установка программ и проверка источников;
- проверка советов ИИ перед применением;
- подготовка отчета и защита виртуальной ИТ-лаборатории.

Практическое задание: Настроить сетевое подключение виртуальной машины, провести базовую диагностику и защитить отчет о созданной ИТ-лаборатории.

Раздел 2. Программирование на Python и ИИ-помощник — 32 часов

Урок 6. Python: среда разработки, синтаксис и первая программа

Содержание:

- назначение языка Python и области применения;
- интерпретатор и среда разработки;
- структура программы, инструкции и комментарии;
- вывод данных и правила оформления кода;
- запуск программы и чтение сообщений об ошибках.

Практическое задание: Настроить среду разработки и создать программу, которая выводит структурированную информацию о выбранном ИТ-проекте.

Урок 7. Переменные, типы данных и пользовательский ввод

Содержание:

- переменные и понятные имена;
- целые и вещественные числа, строки и логические значения;
- преобразование типов;
- функции input и print;
- форматирование строк;
- проверка корректности входных данных.

Практическое задание: Создать интерактивную программу-анкету с вычислениями и форматированным выводом результата.

Урок 8. Условия и логические выражения

Содержание:

- операторы сравнения;
- логические операции and, or, not;
- конструкции if, elif, else;
- вложенные условия;
- проектирование ветвящейся логики;
- тестирование граничных случаев.

Практическое задание: Разработать программу принятия решения с несколькими ветвями и набором тестовых примеров.

Урок 9. Циклы и алгоритмы повторения

Содержание:

- циклы for и while;
- функция range;
- счетчики и накопители;
- условия завершения цикла;
- break и continue;
- ошибки бесконечного цикла и способы их предотвращения.

Практическое задание: Создать программу с меню, повторяющимися действиями и контролируемым завершением.

Урок 10. Строки, списки и обработка данных

Содержание:

- индексы и срезы строк;
- основные методы строк;
- списки, добавление, удаление и перебор элементов;
- поиск, фильтрация и подсчет;
- связь структуры данных с задачей программы.

Практическое задание: Создать приложение для обработки списка данных: добавление, поиск, изменение, удаление и вывод статистики.

Урок 11. Функции и декомпозиция программы

Содержание:

- назначение функций;
- параметры и возвращаемое значение;
- локальные и глобальные переменные;
- декомпозиция задачи на подзадачи;
- повторное использование кода;
- документирование функций.

Практическое задание: Разделить ранее созданную программу на функции и подготовить краткое описание их назначения.

Урок 12. Проектное приложение и ИИ-помощник программиста

Содержание:

- выбор идеи и требований к приложению;
- структура проекта и последовательность разработки;
- формулирование точных запросов к ИИ;
- объяснение сложного кода, поиск ошибки и предложение варианта решения;
- обязательная проверка результата, тестирование и самостоятельная доработка;
- фиксация использованных подсказок и собственного вклада.

Практическое задание: Создать рабочую версию приложения на Python и документировать, как использовался и проверялся ИИ-помощник.

Урок 13. Тестирование, рефакторинг и защита программы

Содержание:

- виды ошибок: синтаксические, логические и ошибки выполнения;
- тест-кейсы и проверка граничных значений;
- отладочный вывод и пошаговый анализ;
- рефакторинг и читаемость кода;
- подготовка демонстрации и объяснение логики;
- защита программы и ответы на вопросы.

Практическое задание: Протестировать, доработать и защитить собственную программу на Python с вводом, условиями, циклами и функциями.

Раздел 3. Интернет вещей (IoT) — 16 часов

Урок 14. Интернет вещей, микроконтроллеры и электрическая безопасность

Содержание:

- понятие IoT и примеры умных устройств;
- микроконтроллер, цифровые и аналоговые сигналы;
- пины питания, земли и ввода-вывода;
- макетная плата и безопасное подключение компонентов;
- структура простого устройства: датчик, обработка, сеть и сервер.

Практическое задание: Собрать безопасную учебную схему с микроконтроллером и проверить базовое управление встроенным светодиодом.

Урок 15. MicroPython и работа с датчиками

Содержание:

- назначение MicroPython;
- подключение среды разработки к микроконтроллеру;
- импорт модулей и управление пинами;
- считывание данных датчика температуры;
- периодический опрос и вывод показаний;
- проверка реалистичности данных.

Практическое задание: Подключить датчик, написать программу считывания температуры и вывести серию измерений.

Урок 16. Wi-Fi и отправка данных на сервер

Содержание:

- подключение устройства к беспроводной сети;
- клиент-серверная модель;
- HTTP-запрос и формат передачи данных;
- защита учетных данных и сетевых параметров;
- обработка ошибок соединения;
- отправка показаний в учебный сервер или облачный сервис.

Практическое задание: Подключить микроконтроллер к Wi-Fi и передать тестовые показания температуры на учебный сервер.

Урок 17. Сборка, тестирование и защита умной метеостанции

Содержание:

- объединение датчика, программы и сетевой передачи;
- интервал измерений и стабильность работы;
- журналирование и анализ ошибок;
- визуализация или просмотр отправленных данных;
- доработка корпуса или схемы подключения;
- подготовка демонстрации проекта.

Практическое задание: Собрать прототип умной метеостанции, проверить передачу данных и представить работу устройства.

Раздел 4. Цифровой рисунок, Krita и нейросети — 24 часов

Урок 18. Krita: рабочее пространство, кисти и цифровой скетч

Содержание:

- назначение растрового редактора Krita;
- холст, разрешение и цветовой профиль;
- кисти, стабилизация линии и графический планшет;
- набросок, силуэт и поиск формы;
- организация рабочего файла.

Практическое задание: Создать серию быстрых цифровых скетчей и выбрать один вариант для дальнейшей разработки.

Урок 19. Слои, выделения, маски и чистовой рисунок

Содержание:

- слои и группы;
- прозрачность и режимы наложения;
- выделения и трансформации;
- маски и неразрушающее редактирование;
- переход от чернового наброска к чистовой линии.

Практическое задание: Подготовить чистовой многослойный рисунок персонажа или объекта.

Урок 20. Цвет, свет, тень и объем

Содержание:

- цветовой круг и гармонии;
- локальный цвет и цвет среды;
- источник света и направление тени;
- тональный контраст и объем;
- материалы и отражающие свойства.

Практическое задание: Выполнить цветовую разработку и светотеневую проработку выбранного рисунка.

Урок 21. Композиция, персонаж и визуальный стиль

Содержание:

- композиционный центр, баланс и ритм;
- пропорции и характер персонажа;
- силуэт и читаемость;
- стилизация и единая система деталей;
- подбор референсов без прямого копирования.

Практическое задание: Разработать заверченный концепт персонажа с листом вариантов или характерных деталей.

Урок 22. Фон и концепт-арт сцены

Содержание:

- перспектива и планы изображения;
- масштаб, глубина и атмосферная перспектива;
- связь персонажа и окружения;
- цветовой сценарий сцены;
- подготовка презентационного концепт-арта.

Практическое задание: Создать фон или цельную сцену, объединяющую персонажа, окружение и освещение.

Урок 23. Нейросети в работе художника и портфолио

Содержание:

- ИИ для поиска идей, стилистических вариантов и референсов;
- доработка собственного наброска без подмены авторской работы;
- сравнение вариантов и критический отбор;
- авторское право, маркировка AI-элементов и сохранение исходников;
- финальная обработка и подготовка серии работ к презентации.

Практическое задание: Собрать серию: скетч, персонаж, фон и финальный арт; показать исходники, AI-варианты и собственную доработку.

Раздел 5. Векторная графика, интерфейсы и ИИ — 20 часов

Урок 24. Векторная графика и компоненты интерфейса

Содержание:

- отличия растровой и векторной графики;
- геометрические формы, контуры, заливки и эффекты;
- масштабирование без потери качества;
- иконки, пиктограммы и визуальная система;
- экспорт векторных материалов.

Практическое задание: Создать набор согласованных иконок и графических элементов для сайта или приложения.

Урок 25. Figma: фреймы, сетки, стили и компоненты

Содержание:

- интерфейс Figma и организация файла;
- фреймы и направляющие;
- цветовые и текстовые стили;
- компоненты и варианты;
- автоматическая компоновка;
- правила именования слоев.

Практическое задание: Создать библиотеку базовых компонентов: кнопки, поля, карточки и навигационные элементы.

Урок 26. Проектирование экранов и пользовательских сценариев

Содержание:

- цель пользователя и сценарий взаимодействия;
- информационная архитектура;
- иерархия, читаемость и доступность;
- состояния интерфейсных элементов;
- переходы между экранами и прототипирование.

Практическое задание: Разработать пользовательский сценарий и не менее трех связанных экранов сайта или приложения.

Урок 27. UI-макет и проверка удобства

Содержание:

- единый визуальный стиль;
- адаптация макета под разные размеры экрана;
- проверка контраста и размеров элементов;
- простое пользовательское тестирование;
- сбор обратной связи и доработка интерфейса.

Практическое задание: Собрать кликабельный прототип, провести тестирование и внести улучшения по результатам обратной связи.

Урок 28. ИИ для идей, стиля и промо-визуалов

Содержание:

- формулирование задачи и ограничений для ИИ;
- поиск идей интерфейса и визуального направления;
- генерация промо-визуалов и оценка соответствия задаче;
- редактирование и интеграция результата в макет;
- прозрачное указание роли ИИ;
- подготовка UI-макета к защите и портфолио.

Практическое задание: Доработать интерфейс с использованием проверенных AI-материалов и защитить макет сайта или приложения.

Раздел 6. 3D-моделирование в Blender и нейросети — 24 часов

Урок 29. Blender: интерфейс, навигация и трансформации

Содержание:

- рабочие области Blender;
- навигация в трехмерном пространстве;
- объектный режим;
- перемещение, вращение и масштабирование;
- координаты, привязки и организация сцены.

Практическое задание: Создать композицию из примитивов, точно расположить объекты и организовать коллекции сцены.

Урок 30. Полигональное моделирование и форма объекта

Содержание:

- режим редактирования;
- вершины, ребра и грани;
- экструзия, вставка, фаска и разрез;
- модификаторы и симметрия;
- топология простой модели.

Практическое задание: Смоделировать предмет или стилизованный объект с использованием базовых операций и модификаторов.

Урок 31. Материалы, текстуры и визуальные свойства

Содержание:

- материал и шейдер;
- цвет, шероховатость и отражение;
- назначение текстур и координат;
- базовая UV-развертка;
- проверка материалов в разных режимах просмотра.

Практическое задание: Создать и настроить несколько материалов, применить их к модели и подготовить простой текстурированный вариант.

Урок 32. Свет, камера и композиция 3D-сцены

Содержание:

- типы источников света;
- ключевой, заполняющий и контровой свет;
- настройка камеры и фокусного расстояния;

- композиция и глубина сцены;
- управление атмосферой и акцентом.

Практическое задание: Построить презентационную сцену с камерой и осмысленной схемой освещения.

Урок 33. Рендер, оптимизация и сборка финальной сцены

Содержание:

- движки рендера и основные настройки;
- разрешение, качество и время расчета;
- шум, тени и оптимизация;
- проверка сцены и исправление артефактов;
- экспорт изображения и сохранение проекта.

Практическое задание: Собрать финальную 3D-сцену, выполнить тестовые рендеры и подготовить итоговое изображение.

Урок 34. Нейросети для концептов и защита 3D-проекта

Содержание:

- генерация референсов и 3D-концептов по тексту или изображению;
- анализ пропорций, материалов и света в AI-референсе;
- адаптация идеи под собственную модель;
- недопустимость выдачи готового AI-результата за собственное моделирование;
- оформление этапов работы и презентационного листа;
- защита модели или сцены.

Практическое задание: Представить финальную 3D-модель или сцену с материалами, светом, рендером, AI-концептом и описанием собственной доработки.

Раздел 7. Комплексное портфолио и итоговая аттестация — 8 часов

Урок 35. Сборка комплексного IT-портфолио

Содержание:

- отбор лучших технических и дизайнерских работ;
- структура цифрового портфолио;
- описание задачи, процесса, инструментов и результата;
- оформление скриншотов, схем, исходников и ссылок;
- фиксация самостоятельного вклада и роли ИИ;
- планирование итогового выступления.

Практическое задание: Собрать портфолио из проектов по системам, Python, IoT, цифровому рисунку, UI и 3D и подготовить презентацию.

Урок 36. Итоговая защита проекта и портфолио

Содержание:

- проверка работоспособности и доступности материалов;
- структура публичного выступления;
- демонстрация выбранного основного проекта;
- объяснение технических или дизайнерских решений;
- ответы на вопросы и самооценка результата;
- обсуждение дальнейшей образовательной траектории.

Практическое задание: Защитить портфолио и один основной проект: программу на Python, IoT-метеостанцию, цифровой арт, UI-макет или 3D-сцену.

5. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

5.1. Контроль знаний, умений и навыков

Формами подведения итогов реализации программы являются текущий контроль, промежуточный контроль по завершении модулей и итоговая аттестация в форме защиты портфолио и выбранного основного проекта.

Текущий контроль проводится на каждом занятии и направлен на оценку понимания понятий, правильности и безопасности практических действий, работоспособности результата, самостоятельности и способности объяснить принятое решение.

Формы текущего контроля:

- наблюдение за выполнением лабораторных и практических заданий;
- устные ответы и объяснение последовательности действий;
- проверка виртуальных машин, команд, исходного кода, схем, цифровых работ, макетов и 3D-сцен;
- тестирование программ и устройств, чтение сообщений об ошибках;
- ведение журнала команд, тест-кейсов, измерений или этапов творческой работы;
- обсуждение решений, взаимопроверка и внесение исправлений.

Промежуточный контроль проводится по окончании каждого профильного модуля.

Формы промежуточного контроля:

- защита виртуальной IT-лаборатории;
- демонстрация и объяснение программы на Python;
- презентация прототипа умной метеостанции;
- презентация серии цифровых работ;
- защита UI-макета;
- защита 3D-модели или сцены.

Итоговая аттестация проводится в форме публичной защиты комплексного портфолио и одного основного проекта. Обучающийся показывает исходные и промежуточные материалы, демонстрирует работоспособность результата, объясняет технические или визуальные решения, отвечает на вопросы и раскрывает, каким образом применял ИИ и что было выполнено самостоятельно.

По завершении обучения обучающимся выдается сертификат о прохождении обучения при условии освоения программы и успешного прохождения итоговой аттестации.

5.2. Критерии оценивания освоения программы

Оценивание проводится по системе «зачтено / не зачтено».

Практические задания

«Зачтено» ставится, если обучающийся:

- выполнил основную часть задания и получил проверяемый результат;
- понимает назначение используемых инструментов, команд, конструкций или визуальных приемов;
- соблюдает требования безопасности и организации файлов;
- может провести тестирование, обнаружить проблему и внести исправление;
- использует ИИ как вспомогательный инструмент, проверяет ответ и способен объяснить итоговое решение.

«Не зачтено» ставится, если обучающийся:

- не приступил к выполнению задания без уважительной причины;
- не выполнил ключевые действия даже с помощью преподавателя;

- не может объяснить назначение основных элементов работы;
- игнорирует требования электрической, цифровой или информационной безопасности;
- представил скопированный или сгенерированный результат без понимания, проверки и самостоятельной доработки.

Итоговое портфолио и основной проект

Критерий оценки	Баллы
Полнота, структура и качество оформления портфолио	0–4
Работоспособность и техническое либо визуальное качество основного проекта	0–4
Обоснованность решений, тестирование и исправление ошибок	0–4
Самостоятельность и ответственное применение ИИ	0–4
Презентация проекта и ответы на вопросы	0–4
Итого	0–20

Итоговая оценка: 0–11 баллов — «не зачтено»; 12–20 баллов — «зачтено».

При оценивании учитываются личный прогресс, сложность выбранного проекта, самостоятельность, способность проверять результат и аргументировать решения. Использование сложных инструментов или ИИ не компенсирует отсутствие понимания и собственной работы.

6. ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ

6.1. Материально-техническое обеспечение

Для реализации программы используется учебная аудитория, расположенная по адресу: г. Королёв, пр. Космонавтов, 20А, пом. 289.

Материально-техническое обеспечение программы включает:

- парта двухместная — 8 шт.;
- стул ученический — 16 шт.;
- стол преподавателя — 1 шт.;
- стул преподавателя — 1 шт.;
- монитор — 1 шт.;
- компьютер — 17 шт.;
- клавиатура — 17 шт.;
- мышь — 17 шт.;
- проектор или большой экран для демонстрации — 1 шт.;
- компьютеры с аппаратной поддержкой виртуализации и достаточным объемом оперативной памяти;
- локальное или облачное хранилище учебных проектов и резервных копий;
- доступ в сеть Интернет и защищенную беспроводную сеть;
- операционные системы Windows и дистрибутив Linux;
- программное обеспечение для создания и запуска виртуальных машин;
- Python и среда разработки;
- микроконтроллеры с поддержкой MicroPython — по количеству рабочих групп;
- датчики температуры, соединительные провода, макетные платы и кабели;
- при необходимости — мультиметр и безопасные низковольтные источники питания;
- графические планшеты — по количеству рабочих мест или на группу;
- графический редактор Krita;
- редактор интерфейсов Figma либо функционально сопоставимый инструмент;

- программное обеспечение Blender;
- современный браузер;
- разрешенные организацией нейросетевые сервисы для работы с текстом, кодом, изображениями и референсами;
- антивирусная защита и средства ограничения доступа к нежелательному контенту.

Программное обеспечение, оборудование и сетевые сервисы используются в соответствии с инструкциями производителей, лицензионными условиями, возрастными ограничениями, правилами обработки персональных данных и локальными актами организации. Работа с микроконтроллерами осуществляется только на безопасном низком напряжении под контролем преподавателя.

6.2. Кадровое обеспечение

Реализация программы обеспечивается педагогическими работниками, имеющими профильное образование и/или практический опыт в сфере информационных технологий, программирования, системного администрирования, электроники, цифрового дизайна, 3D-графики или дополнительного образования детей.

Для реализации инженерного и дизайнерского треков допускается привлечение нескольких педагогов, обладающих взаимодополняющими компетенциями.

Педагогический состав должен обладать следующими компетенциями:

- знание возрастных особенностей и профориентационных потребностей подростков 13–16 лет;
- понимание устройства компьютера, Windows, Linux, виртуализации и базовых сетевых технологий;
- владение Python и основами разработки, тестирования и отладки приложений;
- опыт работы с MicroPython, микроконтроллерами, сенсорами и безопасными низковольтными схемами;
- владение Krita и основами цифрового рисунка;
- владение Figma и основами проектирования пользовательских интерфейсов;
- владение Blender на уровне создания, материалов, освещения и рендера учебных 3D-проектов;
- знание принципов кибербезопасности, защиты персональных данных и авторского права;
- умение безопасно и критически применять генеративный ИИ в образовательном процессе;
- умение организовывать лабораторную, проектную, индивидуальную и командную работу;
- умение проводить критериальное оценивание, давать развивающую обратную связь и сопровождать подготовку портфолио.

7. УЧЕБНО-МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ

7.1. Информационные и учебно-методические материалы

Для реализации программы используется учебно-методический комплект, включающий:

- методические планы и сценарии всех 36 занятий;
- презентации, схемы и демонстрационные материалы преподавателя;
- инструкции по созданию виртуальной машины, установке систем и безопасной работе в терминале;
- карточки с командами Linux и заданиями по сетевой диагностике;
- заготовки, примеры и тестовые наборы для программ на Python;
- схемы подключения микроконтроллеров и датчиков, шаблон журнала измерений;
- задания и референсы по Krita, Figma и Blender;
- шаблоны пользовательских сценариев, компонентов интерфейса и презентационных листов;

- инструкции по тестированию, отладке, резервному копированию и ведению истории версий;
- памятки по составлению запросов к ИИ, проверке ответа, конфиденциальности и авторству;
- шаблон структуры цифрового портфолио и критерии итоговой защиты.

7.2. Используемое программное обеспечение и цифровые ресурсы

- операционная система Windows 10/11;
- дистрибутив Linux, выбранный организацией для учебной работы;
- Oracle VM VirtualBox или функционально сопоставимое программное обеспечение виртуализации;
- Python 3 и среда разработки, выбранная организацией;
- MicroPython и программное обеспечение для загрузки кода на микроконтроллер;
- Krita;
- Figma или функционально сопоставимый редактор интерфейсов;
- Blender;
- современный браузер и сетевые диагностические средства;
- локальный или учебный облачный сервис для приема данных IoT;
- нейросетевые сервисы для объяснения кода, поиска идей и создания референсов, разрешенные организацией и используемые под контролем преподавателя.

7.3. Методы и технологии, использованные при составлении программы

При составлении программы использовались методы и технологии методического проектирования, обеспечивающие соответствие содержания возрасту обучающихся, технической направленности, заявленному объему 144 академических часа и структуре программы Senior.

К использованным методам и технологиям относятся:

- анализ нормативных правовых актов и методических рекомендаций в сфере дополнительного образования детей;
- анализ структуры, продолжительности, содержания, проектов и заявленных результатов программы Senior;
- компетентностный подход: формулирование результатов через знания, умения и практические навыки;
- двухтрековое модульное построение: инженерные и дизайнерские дисциплины с итоговой профориентационной рефлексией;
- практико-ориентированный и лабораторный подходы с обязательным созданием работающего или визуально завершенного результата;
- проектный метод: последовательная постановка задачи, разработка, тестирование, документирование и защита;
- проблемное обучение и отладка через анализ ошибок, логов, измерений и альтернативных решений;
- метод постепенного усложнения: от базовой настройки и примитивов к целостному приложению, устройству, интерфейсу или сцене;
- критическое применение ИИ: запрос, получение варианта, проверка, объяснение, адаптация и фиксация самостоятельного вклада;
- портфолио-технология и профориентационная оценка результатов;
- формирующее и критериальное оценивание;
- технологии безопасной цифровой среды, электрической безопасности и защиты персональных данных.

7.4. Литература и источники, использованные при составлении программы

Ниже приведены источники, использованные при разработке структуры, содержания и учебно-методического обеспечения программы. Раздел не является перечнем обязательной литературы для обучающихся.

Нормативные и методические документы:

1. Федеральный закон от 29 декабря 2012 г. № 273-ФЗ «Об образовании в Российской Федерации».
2. Приказ Министерства просвещения Российской Федерации от 27 июля 2022 г. № 629 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам».
3. Постановление Главного государственного санитарного врача Российской Федерации от 28 сентября 2020 г. № 28 «Об утверждении санитарных правил СП 2.4.3648-20 “Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи”».
4. Распоряжение Правительства Российской Федерации от 31 марта 2022 г. № 678-р «Об утверждении Концепции развития дополнительного образования детей до 2030 года».
5. Письмо Минобрнауки России от 18 ноября 2015 г. № 09-3242 «О направлении информации» (вместе с Методическими рекомендациями по проектированию дополнительных общеразвивающих программ, включая разноуровневые программы).
6. Локальные нормативные акты организации, регламентирующие образовательную деятельность.

Учебно-методические материалы и литература:

1. Matthes E. Python Crash Course: A Hands-On, Project-Based Introduction to Programming. No Starch Press.
2. Sweigart A. Automate the Boring Stuff with Python. No Starch Press.
3. Downey A. Think Python: How to Think Like a Computer Scientist. O'Reilly Media.
4. Bell C. MicroPython for the Internet of Things. Apress.
5. Monk S. Programming the Raspberry Pi Pico in MicroPython. McGraw Hill.
6. Shneiderman B., Plaisant C., Cohen M., Jacobs S., Elmqvist N., Diakopoulos N. Designing the User Interface. Pearson.
7. Tidwell J., Brewer C., Valencia A. Designing Interfaces. O'Reilly Media.
8. Williams R. The Non-Designer's Design Book. Peachpit Press.
9. van Gumster J. Blender For Dummies. Wiley.
10. Krita Foundation. Krita User Manual and Reference Manual.
11. Blender Foundation. Blender Manual.
12. Python Software Foundation. Python Documentation.
13. MicroPython Project. MicroPython Documentation.
14. Методические материалы по кибербезопасности, авторскому праву и ответственному применению генеративного искусственного интеллекта в образовании.

Интернет-ресурсы, использованные при составлении и реализации программы:

<https://xn--b1afoeoan.xn----7sbaqfsbkfbji2w.xn--p1ai/senior/> — описание программы Senior для детей 13–16 лет; дата обращения: 30.07.2026.

<https://www.virtualbox.org/manual/> — официальное руководство Oracle VM VirtualBox; дата обращения: 30.07.2026.

<https://docs.python.org/3/> — официальная документация Python; дата обращения: 30.07.2026.

<https://docs.python.org/3/tutorial/> — официальный учебник Python; дата обращения: 30.07.2026.

<https://docs.micropython.org/> — официальная документация MicroPython; дата обращения: 30.07.2026.

<https://docs.micropython.org/en/latest/esp32/quickref.html> — справочные материалы MicroPython для микроконтроллеров ESP32; дата обращения: 30.07.2026.

https://docs.krita.org/en/user_manual.html — руководство пользователя Krita; дата обращения: 30.07.2026.

<https://help.figma.com/hc/en-us/categories/360002042553-Figma-design> — справочные материалы Figma; дата обращения: 30.07.2026.

<https://docs.blender.org/manual/en/latest/> — официальное руководство Blender; дата обращения: 30.07.2026.

<https://www.saferunet.ru/> — материалы по безопасному интернету; дата обращения: 30.07.2026.

<https://beinternetawesome.withgoogle.com/> — материалы по цифровой безопасности и ответственному поведению; дата обращения: 30.07.2026.

8. ПРИЛОЖЕНИЯ

Приложение 1. Примерные вопросы для контроля знаний

1. Какие основные компоненты входят в состав компьютера?
2. Как оперативная память отличается от накопителя?
3. Для чего нужна операционная система?
4. Что такое учетная запись и права доступа?
5. Что такое виртуальная машина?
6. Для чего нужен снимок состояния виртуальной машины?
7. Какие операции можно выполнять в командной строке Linux?
8. Почему системную команду, предложенную ИИ, необходимо проверять?
9. Что такое IP-адрес?
10. Какова роль DNS?
11. Что такое переменная в Python?
12. Какие типы данных используются в Python?
13. Чем условие отличается от цикла?
14. Для чего нужны функции?
15. Что такое локальная переменная?
16. Какие виды ошибок встречаются в программе?
17. Что такое тест-кейс?
18. Что означает рефакторинг?
19. Что такое микроконтроллер?
20. Чем цифровой сигнал отличается от аналогового?
21. Для чего используется MicroPython?
22. Как датчик передает измерение программе?
23. Что означает клиент-серверная модель?
24. Почему нельзя публиковать пароль Wi-Fi в исходном коде?
25. Что такое слой в Krita?
26. Чем маска отличается от удаления части изображения?
27. Как свет и тень создают объем?
28. Что такое композиционный центр?
29. Что такое концепт-арт?

30. Как безопасно использовать AI-референс?
31. Чем векторная графика отличается от растровой?
32. Что такое компонент интерфейса?
33. Для чего используется сетка?
34. Что такое пользовательский сценарий?
35. Какие признаки делают интерфейс удобным?
36. Для чего нужен кликабельный прототип?
37. Что такое вершина, ребро и грань в Blender?
38. Для чего используется экструзия?
39. Что такое материал и текстура?
40. Как камера влияет на восприятие 3D-сцены?
41. Какие источники света используются в сцене?
42. Что такое рендер?
43. Как оптимизация влияет на время рендера?
44. Почему AI-концепт не является готовой собственной 3D-моделью?
45. Какие разделы должны быть в цифровом портфолио?
46. Как показать самостоятельный вклад?
47. Как подготовить убедительную защиту проекта?

Приложение 2. Примерные практические задания

Задание 1. Компоненты компьютера

Составьте схему персонального компьютера и объясните, как компоненты взаимодействуют при запуске программы.

Задание 2. Виртуальная машина

Создайте виртуальную машину, установите учебную систему и сохраните снимок настроенного состояния.

Задание 3. Командная строка Linux

Выполните создание, копирование, перемещение и удаление учебных файлов и оформите журнал команд.

Задание 4. Сетевая диагностика

Определите параметры сетевого подключения и выполните безопасную проверку доступности учебного ресурса.

Задание 5. Интерактивная программа Python

Создайте приложение с пользовательским вводом, вычислениями и форматированным выводом.

Задание 6. Условия и циклы

Разработайте программу с несколькими ветвями и повторяющимся меню действий.

Задание 7. Списки и функции

Создайте программу учета данных, разделите ее на функции и добавьте поиск или статистику.

Задание 8. Тестирование Python

Составьте не менее шести тест-кейсов, найдите ошибки и зафиксируйте исправления.

Задание 9. Микроконтроллер

Подключите светодиод или иной безопасный компонент и управляйте им с помощью MicroPython.

Задание 10. Датчик температуры

Считайте серию измерений, проверьте реалистичность и сохраните результаты.

Задание 11. Передача данных

Подключите устройство к учебной Wi-Fi-сети и отправьте тестовое значение на сервер.

Задание 12. Умная метеостанция

Объедините датчик, MicroPython и сетевую передачу в работающий прототип.

Задание 13. Цифровой скетч

Создайте серию набросков в Krita и выберите вариант для дальнейшей разработки.

Задание 14. Персонаж и цвет

Подготовьте чистовой рисунок персонажа, цветовую палитру, свет и тень.

Задание 15. Концепт-арт

Создайте сцену с персонажем и окружением, покажите исходные этапы и референсы.

Задание 16. Набор иконок

Разработайте единый набор векторных иконок для сайта или приложения.

Задание 17. UI-компоненты

Создайте в Figma кнопки, поля, карточки и их состояния.

Задание 18. Кликабельный прототип

Свяжите несколько экранов, проведите пользовательское тестирование и улучшите макет.

Задание 19. Полигональная модель

Создайте в Blender стилизованный объект с помощью экструзии и модификаторов.

Задание 20. Материалы и свет

Назначьте материалы, настройте камеру и осветите объект для презентации.

Задание 21. Финальная 3D-сцена

Соберите сцену, выполните тестовые рендеры, исправьте артефакты и экспортируйте итоговое изображение.

Задание 22. Проверка ответа ИИ

Получите от ИИ объяснение команды, кода или визуальной идеи, проверьте его по документации и опишите внесенные исправления.

Задание 23. Портфолио

Отберите лучшие работы, добавьте описание задачи, этапов, инструментов, результата и самостоятельного вклада.

Задание 24. Итоговая защита

Подготовьте демонстрацию портфолио и одного основного проекта, объясните решения и ответьте на вопросы.