Комитет по делам образования города Челябинска
Муниципальное автономное учреждение дополнительного образования
«Дворец пионеров и школьников им. Н.К. Крупской г. Челябинска»

УТВЕРЖДАЮ:
Директор МАУДО «ДПШ»
_____ Ю.В. Смирнова
Приказ МАУДО «ДПШ»
№269-од от 15.05.2025

Дополнительная общеобразовательная общеразвивающая модульная программа «Программирование в Minecraft»

Направленность программы: техническая
Возраст обучающихся: 10-13 лет
Срок реализации: 1 год
Количество часов по программе: 148 часов
Год разработки программы: 2025

Автор-составитель:
Николаева Елена Александровна,
педагог дополнительного образования

Челябинск, 2025 г.

Оглавление

Раздел 1. Пояснительная записка 3
Раздел 2. Содержание Программы 9
2.1 Учебный план дополнительной общеобразовательной общеразвивающей программы «Программирование в Minecraft»..... 9
2.2 Содержание учебного плана дополнительной общеобразовательной общеразвивающей программы «Программирование в Minecraft»..... 11
Раздел 3. Воспитательная деятельность 23
3.1. Цель, задачи, целевые ориентиры воспитания детей..... 23
3.2. Формы и методы воспитания..... 24
3.3. Условия осуществления воспитательного процесса, анализ результатов..... 24
3.4. Календарный план воспитательной работы Программе..... 25
Раздел 4. Формы аттестации и оценочные материалы 27
Раздел 5. Организационно-педагогические условия реализации Программы 31
5.1 Методические материалы..... 31
5.2 Материально-техническое обеспечение и оснащенность образовательного процесса по дополнительной общеобразовательной общеразвивающей программе «Программирование в Minecraft»..... 32
Приложение 1 34
Приложение 2 35
Приложение 3 38

Раздел 1. Пояснительная записка

Перечень нормативно-правовых актов:

1. Федеральный закон от 29.12.2012 №273-ФЗ «Об образовании в Российской Федерации» (ред. от 28.02.2025).
2. Федеральный закон от 24.07.1998 №124-ФЗ «Об основных гарантиях прав ребенка в Российской Федерации» (ред. от 30.11.2024).
3. Стратегия развития воспитания в Российской Федерации на период до 2025 года (распоряжение Правительства Российской Федерации от 29.05.2015 №996-р).
4. Постановление Главного государственного санитарного врача Российской Федерации от 28.09.2020 №28 «Об утверждении санитарных правил СП 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи».
5. Постановление Главного государственного санитарного врача Российской Федерации от 28.01.2021 №2 «Об утверждении санитарных правил и норм СанПиН 1.2.3685-21 «Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания» (рзд.VI. Гигиенические нормативы по устройству, содержанию и режиму работы организаций воспитания и обучения, отдыха и оздоровления детей и молодежи»).
6. Паспорт федерального проекта «Всё лучшее детям» (утвержден на заседании проектного комитета по национальному проекту «Молодёжь и дети»).
7. Постановление Правительства Российской Федерации от 26.12.2017 №1642 «Об утверждении государственной программы Российской Федерации «Развитие образования» (ред. от 21.02.2025).
8. Постановление Правительства Российской Федерации от 11.10.2023 №1678 «Об утверждении Правил применения организациями, осуществляющими образовательную деятельность, электронного обучения, дистанционных образовательных технологий при реализации образовательных программ».
9. Распоряжение Правительства Российской Федерации от 31.03.2022 №678-р «Об утверждении Концепции развития дополнительного образования детей до 2030 г. и плана мероприятий по ее реализации».
10. Приказ Министерства просвещения Российской Федерации от 27.07.2022 №629 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам».
11. Приказ Министерства просвещения Российской Федерации от 03.09.2019 №467 «Об утверждении Целевой модели развития региональных систем дополнительного образования детей» (с изм. и доп. от 21.04.2023).
12. Приказ Министерства просвещения Российской Федерации от 13.03.2019 №114 «Об утверждении показателей, характеризующих общие критерии оценки качества условий осуществления образовательной деятельности организациями, осуществляющими образовательную деятельность по основным общеобразовательным программам, образовательным программам среднего профессионального образования, основным программам профессионального обучения, дополнительным общеобразовательным программам».

13. Приказ Министерства труда и социальной защиты Российской Федерации от 22.09.2021 №652н «Об утверждении профессионального стандарта «Педагог дополнительного образования детей и взрослых».

14. Методические рекомендации по проектированию общеобразовательных программ (включая разноуровневые программы), разработанные Минобранауки России совместно с ГАОУ ВО «МГПУ», ФГАУ «ФИРО» и АНО дополнительного профессионального образования «Открытое образование» (письмо Минобрнауки России №09-3242 от 18.11.2015).

15. Закон Челябинской области от 30.08.2013 №515-ЗО «Об образовании в Челябинской области» (ред. от 29.01.2024).

16. Локальные акты МАУДО «ДПШ».

Направленность Программы «Программирование в Minecraft» (далее Программа): техническая.

Уровень освоения Программы: базовый.

Актуальность Программы:

В современном мире, где технологии развиваются с невероятной скоростью, умение программировать становится неотъемлемой частью образования. Одним из наиболее интересных и доступных способов введения детей в мир программирования является использование платформы Minecraft. Дополнительная общеобразовательная общеразвивающая программа «Программирование в Minecraft» становится актуальной не только благодаря популярности самой игры, но и благодаря множеству преимуществ, которые она предоставляет.

1. Интерактивность и вовлеченность: Minecraft — это игра, которая привлекает внимание миллионов детей по всему миру. Использование этой платформы в образовательных целях позволяет создать интерактивную среду, в которой обучающиеся могут не только изучать основы программирования, но и применять свои знания на практике. Игровой процесс делает обучение увлекательным и мотивирующим, что способствует более глубокому усвоению материала.

2. Развитие логического мышления и креативности: программирование в Minecraft требует от детей логического мышления и креативного подхода к решению задач. Создание различных механизмов, автоматизация процессов и разработка игр внутри Minecraft развивают аналитические способности и учат обучающихся мыслить нестандартно. Это, в свою очередь, помогает формировать навыки, которые будут полезны им в будущем, как в учебе, так и в профессиональной деятельности.

3. Командная работа и социальные навыки: работа в команде — важный аспект как в программировании, так и в жизни. Программа часто включает в себя групповые проекты, что способствует развитию социальных навыков у обучающихся. Они учатся делиться идеями, обсуждать решения и совместно достигать поставленных целей. Это не только улучшает их коммуникативные навыки, но и формирует чувство ответственности за общий результат.

4. Подготовка к будущему: в условиях быстро меняющегося мира, где цифровые технологии становятся основой большинства профессий, обучение

программированию с раннего возраста становится необходимостью. Программа не только обучает детей основам кодирования, но и подготавливает их к будущим вызовам, связанным с технологическим прогрессом. Умение работать с кодом, разрабатывать алгоритмы и решать сложные задачи станет важным преимуществом на рынке труда.

Таким образом, Программа представляет собой актуальный и эффективный инструмент для обучения детей основам программирования. Она не только способствует развитию логического мышления и креативности, но и формирует важные социальные навыки. В условиях стремительных изменений в мире технологий такая Программа становится необходимостью, обеспечивая обучающимся возможность подготовиться к будущему и стать активными участниками цифровой экономики.

Отличительные особенности Программы заключаются в её междисциплинарности. Дети не просто учатся кодить — они проектируют автоматические фермы, создают мини-игры и умные дома, интегрируя знания по математике, физике и дизайну. Например, работа с редстоуном (игровым аналогом электричества) развивает логическое мышление, а программирование на Python через библиотеку mcpi знакомит с текстовым кодом. Важным элементом является геймификация: задания подаются как квесты, а финальные проекты (например, многопользовательские игры или AI-механизмы) позволяют сразу увидеть результат своих усилий. Программа также поддерживает командную работу, имитируя реальные процессы разработки, где каждый участник берёт на себя роль программиста, дизайнера или тестировщика.

Minecraft, будучи одной из самых популярных игр среди детей и подростков, служит идеальной платформой для вовлечения в обучение. Использование Minecraft Education Edition и инструментов вроде MakeCode и Python позволяет трансформировать игровую активность в практическое обучение, что повышает мотивацию и снижает барьер входа для новичков.

В отличие от стандартных курсов, здесь программирование изучается не абстрактно, а через взаимодействие с живым игровым миром. Например, обучающиеся могут создать нейросеть для распознавания блоков или настроить сервер с плагинами, что выходит за рамки базового обучения. Особый акцент сделан на проектной работе: обучающиеся не только пишут код, но и учатся презентовать свои идеи, анализировать ошибки и адаптировать решения под меняющиеся условия. Это формирует не только технические навыки, но и soft skills — критическое мышление, креативность и умение работать в команде.

Воспитательный потенциал Программы предполагает развитие технических способностей обучающихся, в том числе развитие логического и пространственного мышления, способствует раскрытию творческого потенциала личности, формированию усидчивости и трудолюбия, приобретению практических умений и навыков в области компьютерных технологий, способствует интеллектуальному развитию обучающегося.

В современном мире информационных технологий и цифровых инноваций особенно актуальным становится использование современных средств обучения для

формирования у детей навыков программирования и логического мышления. Одним из эффективных методов является внедрение образовательных программ, основанных на популярной среди молодого поколения видеоигре Minecraft.

Программа обладает значительным воспитательным потенциалом, способным развивать у обучающихся не только технические знания, но и важные личностные качества:

1. Формирование творческих и креативных способностей: Minecraft — это своего рода цифровая площадка для творчества. В рамках Программы дети учатся создавать собственные моды, автоматизировать процессы с помощью командных блоков и писать скрипты, что стимулирует развитие креативности. Возможность реализовать свои идеи на практике способствует формированию инициативности, самостоятельности и умения решать нестандартные задачи.

2. Развитие логического и аналитического мышления: процесс обучения программированию в Minecraft требует от обучающихся понимания логики, алгоритмических структур и последовательности действий. Этот опыт помогает формировать аналитический склад ума, развивать способность к системному и критическому мышлению — важнейшие качества современной личности.

3. Воспитание ответственности и командной работы: работа в учебных проектах и командных заданиях в Minecraft учит детей ответственности за результаты своих действий и умению взаимодействовать со сверстниками. Совместное создание проектов, обмен знаниями и помощь друг другу формируют коммуникативные навыки, терпимость и уважение к мнению других.

4. Воспитательный потенциал для формирования ценностей: обучение программированию в Minecraft способствует развитию таких нравственных ценностей, как ответственность, целеустремленность, терпение и настойчивость. Проектная деятельность и преодоление трудностей в процессе обучения учат детей ценить результат своей работы и стремиться к постоянному развитию.

Таким образом, Программа обладает мощным воспитательным потенциалом, так как способствует развитию технических навыков, креативности, ответственности и коммуникативных умений, что является залогом формирования гармонично развитой личности. Внедрение таких Программ в систему дополнительного образования помогает не только подготовить детей к будущему профессиональному миру, но и воспитать их как ответственных и инициативных граждан современного общества.

В содержание Программы включены темы: «Мой Дворец», «Мой выбор». Тема «Мой Дворец» предполагает знакомство с историей и традициями Дворца. Тема «Мой выбор» рассчитана на профессиональную ориентацию обучающихся.

Адресат Программы: 10-13 лет.

Младший школьный возраст - 8-11 лет. Ключевым, психолого-педагогическим аспектом данного возрастного периода является развитие психики детей на основе ведущей деятельности – учения. Младшие школьники отличаются остротой и свежестью восприятия, своего рода созерцательной любознательностью. Восприятие на этом уровне психического развития связано с практической деятельностью ребёнка.

Средний школьный возраст - 12-14 лет. Одним из ключевых факторов, характеризующих средний школьный возраст, является развитие мышления. Идеальная форма – то, что ребенок осваивает в этом возрасте, с чем он реально взаимодействует, - это область моральных норм, на основе которых строятся социальные взаимоотношения. Общение со своими сверстниками – ведущий тип деятельности в этом возрасте. В данном возрасте стабилизируются черты характера и основные формы межличностного поведения. Период характеризуется особенным вниманием ребёнка к собственным недостаткам.

Объем, структура, содержание, формы и методы обучения определены в соответствии с возрастными, физиологическими и психологическими особенностями обучающихся.

Данная Программа может быть освоена обучающимися с ОВЗ (по запросу родителей (законных представителей) обучающихся. В данном случае составляется индивидуальный учебный план освоения дополнительной общеобразовательной общеразвивающей программы.

Оптимальное количество обучающихся по Программе (в одной группе) – 15 человек.

Цель Программы: обучение обучающихся основам программирования с использованием MakeCode и Python в результате разработки игрового контента на базе Minecraft.

Задачи Программы:

Личностные:

- сформировать уверенность в решении технических задач;
- сформировать интерес к IT-сфере и технологиям;
- создать стимул у детей к созданию креативных проектов;

Метапредметные:

- развить умения планировать и осуществлять свою деятельность;
- развить навыки конструктивного взаимодействия внутри коллектива на основе принятых норм взаимоотношений и освоить различные социальные роли, умение работать на общий результат и нести ответственность за свои обязанности и поручения.

Предметные (образовательные):

- освоить базовые конструкции программирования: алгоритмы, циклы, условия, функции
- научиться работать с координатами и геометрией в Minecraft (построение фигур, лабиринтов)
- изучить инструменты визуального программирования (MakeCode) и текстового (Python) для управления игровым миром.

Планируемые образовательные результаты

Личностные:

- сформирована уверенность в решении технических задач;
- сформирован интерес к IT-сфере и технологиям;
- создан стимул у обучающихся к созданию креативных проектов.

Метапредметные:

- обучающийся умеет планировать и осуществлять свою деятельность;
- умеет взаимодействовать со сверстниками и педагогом.

Предметные (образовательные):

- освоены базовые конструкции программирования: алгоритмы, циклы, условия, функции
- научились работать с координатами и геометрией в Minecraft (построение фигур, лабиринтов)
- изучили инструменты визуального программирования (MakeCode) и текстового (Python) для управления игровым миром.

Объем Программы: 148 часов.

Формы обучения: очная. Программа может быть реализована с применением дистанционных образовательных технологий.

Виды занятий: практическое занятие, защита проекта, объяснение, беседа, визуализация.

Срок освоения Программы: 1 год.

Режим занятий

Занятия проводятся два раза в неделю по 2 академических часа, перерыв между занятиями 10 минут.

Формы контроля:

Текущий контроль: практическая работа для раздела «Введение в Minecraft и Make Code», практическая работа для раздела «Игровая механика и автоматизация», практическая работа для раздела «Продвинутый Python».

Итоговый контроль: защита проектов.

Раздел 2. Содержание Программы

2.1 Учебный план дополнительной общеобразовательной общеразвивающей программы «Программирование в Minecraft»

Предмет: «Программирование»

№ п/п	Наименование разделов и тем	Общее кол-во часов	Из них:		Формы аттестации/ контроля
			теория	практика	
1.	**Раздел 1. Введение в Minecraft и Make Code**	**18**	**7**	**11**	
1.1	Вводное занятие «Мой Дворец»: история Дворца пионеров. Инструктаж по технике безопасности. Знакомство с интерфейсом Minecraft (Creative/Survival)	2	1	1	
1.2	Основы командных блоков. Создание телепортов	2	1	1	
1.3	Введение в MakeCode: события и циклы. Проект: «Автоматическая дверь»	4	2	2	
1.4	Переменные в MakeCode	2	1	1	
1.5	Создание лабиринта с ловушками	4	1	3	
1.6	Ферма с автоматическим сбором урожая. Текущий контроль	4	1	3	практическая работа
2.	**Раздел 2. Основы Python для Minecraft**	**20**	**9**	**11**	
2.1	Установка и настройка Python + библиотека mcpi. Знакомство с синтаксисом	2	1	1	
2.2	Работа с координатами: построение геометрических фигур (пирамида, спираль)	2	1	1	
2.3	Условия if-else. Дождь из TNT при нажатии кнопки	2	1	1	
2.4	Циклы for и while. Спираль из алмазов	4	2	2	
2.5	Функции. Генератор случайных лабиринтов	4	2	2	
2.6	Работа с массивами	2	1	1	
2.7	Интеграция Python и MakeCode. Автоматический маяк	4	1	3	
3.	**Раздел 3. Игровая механика и автоматизация**	**20**	**6**	**14**	
3.1	Redstone: основы логических схем. Секретная дверь	2	1	1	
3.2	Создание мини-игры «Сапёр» на Python. Генерация мин, подсчёт очков	4	1	3	
3.3	Разработка квеста: «Поиск сокровищ» с использованием условий и циклов	4	1	3	

3.4	Автоматизация: «Ферма с самонаводящимися вагонетками»	2	1	1	
3.5	Многопользовательские игры: «Королевская битва» (зоны, бонусы)	4	1	3	
3.6	Создание умного дома с системой безопасности. Текущий контроль	4	1	3	практическая работа
4.	**Раздел 4. Новые функции и расширенные возможности**	**16**	**5**	**11**	
4.1	Датчики и события в Minecraft	2	1	1	
4.2	Графические интерфейсы (GUI)	4	1	3	
4.3	Многопользовательские взаимодействия	4	1	3	
4.4	Интеграция внешних данных	2	1	1	
4.5	Сложные механизмы на Redstone	4	1	3	
5.	**Раздел 5. Продвинутый Python**	**30**	**11**	**19**	
5.1	Классы и объекты. Создание NPC (персонаж с диалогами)	2	1	1	
5.2	Работа с файлами: сохранение прогресса игрока	2	1	1	
5.3	API Minecraft: взаимодействие с сервером. Онлайн-чат	2	1	1	
5.4	Игра-РПГ: инвентарь, квесты, система уровней.	4	1	3	
5.5	Интеграция с базами данных (SQLite). Рейтинг игроков	4	1	3	
5.6	Многопоточность. Параллельное выполнение задач	2	1	1	
5.7	Создание модов. Добавление новых предметов и блоков	2	1	1	
5.8	Искусственный интеллект. Простые алгоритмы для NPC (поиск пути)	2	1	1	
5.9	Сетевые запросы. Получение данных из внешних API (погода, время)	2	1	1	
5.10	Мини-игра с ИИ. «Охота на мобов» с умными противниками	4	1	3	
5.11	«Умный город» с API и ИИ. Текущий контроль	4	1	3	практическая работа
6.	**Раздел 6. Многопользовательские игры**	**20**	**6**	**14**	
6.1	Настройка локального сервера (Minecraft Java + Spigot)	2	1	1	
6.2	Создание мини-игр для сервера: «Королевская битва»	4	1	3	
6.3	Многопользовательский квест. Головоломки, требующие взаимодействия	4	1	3	
6.4	Плагины: добавление экономики (виртуальные деньги)	4	1	3	

6.5	Командная разработка: парк аттракционов с ролями (строитель, программист)	6	2	4	
7.	**Раздел 7. Креативное проектирование: 3D-моделирование и анимация**	**10**	**5**	**5**	
7.1	Основы 3D-моделирования в Minecraft	4	2	2	
7.2	Анимация и визуальные эффекты	4	2	2	
7.3	Интерактивное искусство: от пикселей к инсталляциям	2	1	1	
8.	**Раздел 8. От идеи до реализации**	**14**	**4**	**10**	
8.1	«Мой выбор»: профессия программист	2	2	-	
8.2	Разработка идеи: масштабный проект	4	1	3	
8.3	Реализация проекта. Консультации	6	1	5	
8.4	Итоговое занятие «Программирование в Minecraft». Презентация игр. Промежуточная аттестация	2	-	2	защита проекта
	Итого	**148**	**53**	**95**	

2.2 Содержание учебного плана дополнительной общеобразовательной общеразвивающей программы «Программирование в Minecraft»

Раздел 1. Введение в Minecraft и Make Code (18 часов).

Тема 1.1 Вводное занятие «Мой Дворец»: история Дворца пионеров. Инструктаж по технике безопасности. Знакомство с интерфейсом Minecraft (Creative/Survival) (2 часа).

Теория (1 час).

Что такое Дворец пионеров и школьников и его значение. Когда и почему был основан Дворец пионеров в Челябинске. Основные цели и задачи, которые ставились перед учреждением. Деятельность Дворца. Как менялась деятельность Дворца с течением времени. Текущие программы и проекты, осуществляемые в Дворце. Роль Дворца в жизни современных детей и подростков. Инструктаж по технике безопасности. Правила поведения в Minecraft: запрет на разрушение чужих построек, уважение к совместной работе.

Практика (1 час).

Интерфейс в Minecraft. Creative Mode: панель быстрого доступа, инвентарь, выбор блоков. Survival Mode: основы выживания (сбор ресурсов, крафт). Постройка небольшого здания (дом, мост) в Creative Mode.

Тема 1.2 Основы командных блоков. Создание телепортов (2 часа).

Теория (1 час).

Как получить командный блок. Типы командных блоков: импульсный, цепной, повторяющийся. Команда телепортации, синтаксис.

Практика (1 час).

Создание сети телепортов. Разместить командные блоки в разных локациях (дом, ферма, шахта). Настроить активацию через рычаги или кнопки. Добавить эффекты: частицы (/particle) или звуки (/playsound).

Тема 1.3 Введение в MakeCode: события и циклы (4 часа).
Теория (2 часа).
События в программировании (on chat command, on button pressed). Циклы (repeat, while).
Практика (2 часа).
Создание проекта «умная дверь» с использованием события и цикла.

Тема 1.4 Переменные в MakeCode (2 часа).
Теория (1 час).
Переменные. Хранение данных: числа, строки, логические значения. Использование переменных в играх.
Практика (1 час).
Создание счетчика очков с выводом сообщения в чат.

Тема 1.5 Создание лабиринта с ловушками (4 часа).
Теория (1 час).
Проектирование лабиринта (тупики, петли, секретные комнаты). Ловушки: TNT, лава, стреляющие dispensers. Активация ловушек: использование командных блоков или MakeCode.
Практика (3 часа).
Постройка лабиринта, использование блоков (камень, кирпич) и декораций (факелы, паутина). Настройка ловушек, добавить секретную комнату с сундуком.

Тема 1.6 Ферма с автоматическим сбором урожая (4 часа).
Теория (1 час).
Принципы автоматизации (программируемый NPC для выполнения задач, редстоун: передача сигналов для активации механизмов). Этапы работы фермы.
Практика (3 часа).
Создание фермы пшеницы, добавить редстоун-механизм для автоматического полива (ведро с водой). Цикл сбора урожая каждые 10 минут. Текущий контроль: обучающиеся должны создать автоматическую ферму пшеницы, добавить редстоун-механизм для автоматического полива (ведро с водой). Цикл сбора урожая каждые 10 минут.

Раздел 2. Основы Python для Minecraft (20 часов).
Тема 2.1 Установка и настройка Python + библиотека mcpi. Знакомство с синтаксисом (2 часа).
Теория (1 час).
Обзор Python как языка программирования, его применения. Установка интерпретатора Python (на примере Windows). Библиотека mcpi. Назначение

библиотеки: взаимодействие с миром Minecraft. Подключение к Minecraft. Запуск локального сервера или использование Minecraft Education Edition.
Практика (1 час).
Написание скрипта, который выводит координаты игрока в чат каждые 5 секунд.

Тема 2.2 Работа с координатами: построение геометрических фигур (пирамида, спираль) (2 часа).
Теория (1 час).
Система координат в Minecraft. Как использовать координаты для построения объектов. Примеры фигур куб (циклы для заполнения пространства блоками) и спираль (математические формулы для расчета позиций).
Практика (1 час).
Построение круга из алмазных блоков. Создание трёхмерной звёзды (использование вложенных циклов для объёмных фигур).

Тема 2.3 Условия if-else. Дождь из TNT при нажатии кнопки (2 часа).
Теория (1 час).
Логические операторы, сравнение значений (==, !=, >, <.). Логические И (and), ИЛИ (or). Синтаксис if-else.
Практика (1 час).
Активация дождя TNT при нажатии кнопки (используется редстоун-сигнал).

Тема 2.4 Циклы for и while (4 часа).
Теория (2 часа).
Цикл for, вложенные циклы для 3D-построек. Цикл while. Вложенные циклы, оптимизация циклов.
Практика (2 часа).
Создание автоматической башни с этажами, создание лабиринта с движущимися стенами. Рандомизированный лес.

Тема 2.5 Функции. Генератор случайных лабиринтов (4 часа).
Теория (2 часа).
Функции в Python. Объявление функции, возвращение значения. Значения функций в коде. Алгоритм «Случайное блуждание», создание сетки блоков и «прорывание» стен в случайных направлениях. Использование модуля random. Оптимизация.
Практика (2 часа).
Создание функции для генерации стен. Генератор лабиринта 10x10. Добавление сундука с наградой в конце лабиринта.

Тема 2.6 Работа с массивами (2 часа).
Теория (1 час).
Массивы (списки) в Python. Создание, индексация, методы массивов. Применение в Minecraft.
Практика (1 час).

Создание инвентаря игрока. Случайный спавн мобов.

Тема 2.7 Интеграция Python и MakeCode. Автоматический маяк (4 часа).
Теория (1 час).
Совместное использование Python и MakeCode. MakeCode: Для быстрого прототипирования (например, реакция на кнопку). Python: Для сложной логики (работа с API, базы данных).
Практика (3 часа).
Создание маяка, который включается ночью. Написать код в MakeCode. Написать код на Python.

Раздел 3. Игровая механика и автоматизация (20 часов).
Тема 3.1 Redstone: основы логических схем. Секретная дверь (2 часа).
Теория (1 час).
Основы редстоуна: провода, повторители, компараторы: их функции и взаимодействие. Логические элементы: AND (И), OR (ИЛИ), NOT (НЕ) — принципы работы и применение. Секретные механизмы: создание дверей, активируемых комбинацией сигналов (например, нажатие рычага + наличие ключевого предмета). Использование поршней и люков для скрытых проходов.
Практика (1 час).
Создание секретной двери. Схема: рычаг → редстоун-цепь → поршень, открывающий проход. Усложнить при помощи добавления кодового замка с использованием 3 рычагов (логическое И).

Тема 3.2 Создание мини-игры «Сапёр» на Python. Генерация мин, подсчёт очков (4 часа).
Теория (1 час).
Алгоритм игры, генерация мин с помощью модуля random: случайное размещение TNT в сетке 10x10. Подсчёт мин вокруг клетки: проверка соседних блоков. Использование функций для открытия клеток и проверки условий победы/поражения.
Практика (3 часа).
Создание мини-игры «Сапёр». Добавление подсчёта очков за разминированные клетки.

Тема 3.3 Разработка квеста: «Поиск сокровищ» с использованием условий и циклов (4 часа).
Теория (1 час).
Создание цепочки заданий: найти карту → расшифровать подсказки → открыть сундук. Использование NPC для выдачи заданий и диалогов (библиотека mcpi). Триггеры: активация событий при нахождении предметов (например, ключ открывает портал).
Практика (3 часа).

Разработать квест. Создать NPC с диалогами. Настроить триггеры. Добавление головоломок: перемещение блоков по схеме, решение арифметических задач для открытия дверей.

Тема 3.4 Автоматизация: «Ферма с самонаводящимися вагонетками» (2 часа).
Теория (1 час).
Использование редстоуна для управления вагонетками: активация рельс, воронки для сбора ресурсов. Оптимизация маршрутов: петли, остановки, сортировка предметов.
Практика (1 час).
Постройка фермы. Создание рельсового пути от фермы к складу. Настройка воронок для автоматической загрузки урожая в вагонетки.

Тема 3.5 Многопользовательские игры: «Королевская битва» (зоны, бонусы) (4 часа).
Теория (1 час).
Механика игры. Сужение игровой зоны: уменьшение безопасной области каждые 5 минут. Случайные бонусы: спавн брони, оружия, зелий. Распределение ресурсов для предотвращения доминирования одного игрока.
Практика (3 часа).
Создание арены. Постройка границ из непроходимых блоков (bedrock). Генерация лут-боксов с бонусами, написать скрипт для сужения зоны.

Тема 3.6 Создание умного дома с системой безопасности. Текущий контроль (4 часа).
Теория (1 час).
Компоненты системы, датчики движения: активация света и ловушек, автоматические двери: открытие по распознаванию игрока. Использование скриптов для управления системами через события (например, время суток).
Практика (3 часа).
Создать умный дом. Настроить освещение, включающееся при приближении, ловушки: ямы с лавой, активируемые при вторжении, видеонаблюдение: камеры (созданные из стекла и табличек с текстом). Текущий контроль: обучающиеся должны создать умный дом. Настроить освещение, включающееся при приближении, ловушки: ямы с лавой, активируемые при вторжении, видеонаблюдение: камеры (созданные из стекла и табличек с текстом).

Раздел 4. Новые функции и расширенные возможности (16 часов).
Тема 4.1 Датчики и события в Minecraft (2 часа).
Теория (1 час).
Типы датчиков (датчики движения, датчики давления, световые сенсоры). Как обрабатывать события (нажатие кнопки, открытие сундука) через код. Использование триггеров для запуска действий.
Практика (1 час).

Создание «умной сигнализации». Настройка датчика движения для активации лампы и звукового сигнала при приближении мобов.

Тема 4.2 Графические интерфейсы (GUI) (4 часа).
Теория (1 час).
Основы GUI в Minecraft. Использование модов (например, CustomNPCs) для создания кнопок, меню и диалогов. Форматы данных: JSON для настройки интерфейсов. Принципы UX.
Практика (3 часа).
Создание меню настроек. Добавление кнопок для управления игрой (пауза, смена сложности). Интеграция с Python для сохранения выбранных опций.

Тема 4.3 Многопользовательские взаимодействия (4 часа).
Теория (1 час).
Особенности многопользовательских игр. Синхронизация данных между игроками через сервер. Проверка действий на стороне сервера. Сетевые механики (Общий инвентарь, командные квесты, PvP-режимы).
Практика (3 часа).
Создание игры «Захват флага». Создание двух баз с флагами и зоной для сражений. Скрипт для подсчёта очков (Python).

Тема 4.4 Интеграция внешних данных (2 часа).
Теория (1 час).
Работа с API. Получение данных из внешних источников (погода, курс валют). Использование библиотек requests и json в Python. Сохранение прогресса игроков в SQLite или облачных хранилищах.
Практика (1 час).
Создание погодной системы. Загрузка реальных данных о погоде через API OpenWeatherMap. Адаптация игрового мира (дождь в Minecraft при дожде в реальности).

Тема 4.5 Сложные механизмы на Redstone (4 часа).
Теория (1 час).
Логические вентили (XOR, NAND), компараторы, повторители с задержкой. Создание таймеров и цепных реакций. Системы сортировки предметов, автоматические фермы мобов.
Практика (3 часа).
Автоматическая ферма скелетов. Использование поршней и воды для направления мобов в ловушку. Редстоун-схема для активации дробителя. Интеграция с Python для подсчёта добычи.

Раздел 5. Продвинутый Python (30 часов).
Тема 5.1 Классы и объекты. Создание NPC (персонаж с диалогами) (2 часа).
Теория (1 час).

Классы как шаблоны для объектов (пример: NPC — неигровой персонаж). Поля (имя, здоровье) и методы (диалог, движение). Создание класса. Наследование: создание специализированных NPC (торговец, стражник).
Практика (1 час).
Реализация NPC с диалогами. Интеграция с Minecraft: NPC, реагирующий на приближение игрока.

Тема 5.2 Работа с файлами: сохранение прогресса игрока (2 часа).
Теория (1 час).
Сохранение в JSON: структуры данных, сериализация. Безопасность: проверка целостности файлов. Чтение/запись файлов: open(), json.load(), json.dump().
Практика (1 час).
Написать скрипт сохранения инвентаря. Загрузка данных: восстановление прогресса после перезапуска игры.

Тема 5.3 API Minecraft: взаимодействие с сервером. Онлайн-чат (2 часа).
Теория (1 час).
Взаимодействие с сервером. Использование библиотеки mcpi для отправки сообщений. Создание REST-API для управления игрой (например, через Flask).
Практика (1 час).
Онлайн-чат между игроками. Добавление команд: /msg, /broadcast.

Тема 5.4 Игра-РПГ: инвентарь, квесты, система уровней (4 часа).
Теория (1 час).
Элементы RPG. Инвентарь (слоты, вес), квесты (цепочки заданий), уровни (опыт, навыки). Баланс: настройка сложности врагов и наград.
Практика (3 часа).
Создание игры в жанре RPG, настроить систему уровней и квесты. Квест «Убийство босса»: спавн моба и выдача награды.

Тема 5.5 Интеграция с базами данных (SQLite). Рейтинг игроков (4 часа).
Теория (1 час).
Основы SQL. Создание таблиц: CREATE TABLE players (name TEXT, score INT). Запросы: SELECT, INSERT, UPDATE. Подключение к базе, выполнение запросов.
Практика (3 часа).
Разработка рейтинга игроков. Отображение топа игроков в чате Minecraft.

Тема 5.6 Многопоточность. Параллельное выполнение задач (2 часа).
Теория (1 час).
Что такое поток, зачем нужна многопоточность, какие бывают проблемы без потоков. Главные потоки и дочерние потоки.
Практика (1 часа).

Создать костёр, который мигает и выпускает искры в отдельном потоке. Создать систему, где игрок может копать блоки, а в это время в фоне генерируется новый участок карты.

Тема 5.7 Создание модов. Добавление новых предметов и блоков (2 часа).
Теория (1 час).
Основы моддинга. Использование MCreator или прямого редактирования .jar-файлов. Добавление текстур и моделей блоков.
Практика (1 час).
Новый блок «Радиоактивная руда». Настройка эффектов при добыче (взрыв, ядовитый туман).

Тема 5.8 Искусственный интеллект. Простые алгоритмы для NPC (поиск пути) (2 часа).
Теория (1 час).
Примеры алгоритмов: NPC убегает при низком здоровье, NPC ищет укрытие во время дождя или прячется ночью. Реакция на изменения в окружении: время суток, погода, наличие угроз (игрок с оружием). Использование условий (if-else) и вероятностных решений (random).
Практика (1 час).
NPC-стражник с интеллектуальным поведением. Настроить поведение. День: Патрулирует территорию по заданному маршруту. Ночь: Возвращается на базу и включает факелы. Угроза: При обнаружении врага (игрок в броне) поднимает тревогу (звук и сигнальные огни).

Тема 5.9 Сетевые запросы. Получение данных из внешних API (погода, время) (2 часа).
Теория (1 час).
Типы API и их особенности. REST vs SOAP: Различия в структуре, форматах данных (JSON/XML), методах запросов (GET, POST). Публичные и защищённые API: Использование API-ключей, OAuth 2.0 для аутентификации. Rate Limits: Ограничения на количество запросов и способы их обхода (кэширование, паузы).
Асинхронные запросы: Использование библиотеки aiohttp для параллельной обработки. Обработка ошибок: Исключения при таймаутах, неверных статус-кодах (404, 500), парсинге данных. Динамическое изменение игрового мира на основе внешних данных (например, дождь в Minecraft при дожде в реальности). Использование API времени для синхронизации игровых суток с реальными.
Практика (1 час).
Сравнение данных из двух погодных API. Запрос к OpenWeatherMap и WeatherAPI. Анализ расхождений и выбор наиболее точного прогноза. Сохранение результатов запросов в локальный файл для уменьшения нагрузки на API. Настроить влияние погоды на NPC. Обработка ошибок и асинхронность. Использование try-except для перехвата исключений. Асинхронный запрос к API времени для обновления игрового времени без задержек.

Тема 5.10 Мини-игра с ИИ. «Охота на мобов» с умными противниками (4 часа).
Теория (1 час).
ИИ для мобов: поведенческие деревья: атака, побег, патрулирование. Адаптация под действия игрока. Древовидные решения: Выбор действий на основе условий (например: «Если здоровье < 20% → убегать»). Использование пулов объектов для управления множеством мобов без лагов. Баланс между сложностью и производительностью. Влияние погоды (дождь снижает видимость, мобы чаще прячутся). Время суток (ночью мобы агрессивнее, днём — осторожнее).
Практика (3 часа).
Создание мини-игры «Охота на мобов». Создание классов мобов с уникальным поведением (сапер, танк, снайпер, лидер). Реализация алгоритма поиска пути (обход препятствий, использование сетки Minecraft для построения карты). Настроить уровень сложности и взаимодействие с окружением (мобы активируют поршни, чтобы заблокировать игрока, сапёры закладывают TNT в узких проходах), создать кооперативные атаки.

Тема 5.11 «Умный город» с API и ИИ. Текущий контроль (4 часа).
Теория (1 час).
Концепция умного города. Автоматизация инфраструктуры: управление освещением, транспортом, энергоснабжением через код. Интеграция внешних данных: использование API для получения реальных данных (погода, время, трафик). ИИ-оптимизация: алгоритмы для распределения ресурсов, предотвращения пробок, адаптации к игровым событиям. Добавление ключевых моментов (умное освещение, транспорт, система безопасности, экологический мониторинг). API для данных. ИИ-алгоритмы.
Практика (3 часа).
Практическая работа. Создать «Умный город» с управляемым освещением (уличные фонари включаются автоматически в реальном времени, синхронизированном с вашим часовым поясом), автоматизированным транспортом (вагонетки перемещаются между станциями, избегая столкновений и выбирая оптимальный маршрут). Добавить дашборд в Minecraft, отображающий данные через таблички: время суток, погода, активность мобов. Настроить энергопотребление (солнечные панели активны днём, ветрогенераторы — при сильном ветре (данные из API). Текущий контроль: обучающиеся должны создать «Умный город» с управляемым освещением (уличные фонари включаются автоматически в реальном времени, синхронизированном с вашим часовым поясом), автоматизированным транспортом (вагонетки перемещаются между станциями, избегая столкновений и выбирая оптимальный маршрут). Добавить дашборд в Minecraft, отображающий данные через таблички: время суток, погода, активность мобов. Настроить энергопотребление (солнечные панели активны днём, ветрогенераторы — при сильном ветре (данные из API).

Раздел 6. Многопользовательские игры (20 часов).
Тема 6.1 Настройка локального сервера (Minecraft Java + Spigot) (2 часа).

Теория (1 час).
Типы серверов: Vanilla, Spigot, Bukkit, PaperMC. Основные понятия: порты, IP-адреса, файл конфигурации server.properties. Инструменты и требования. Безопасность сервера.
Практика (1 час).
Запуск BuildTools для сборки сервера. Настройка server.properties (режим выживания, включение сетевого доступа). Запуск сервера, проверка доступности через локальную сеть. Подключение игроков, проверка базовых команд (/op, /gamemode). Установка EssentialsX для управления сервером. Настройка разрешений через LuckPerms.

Тема 6.2 Создание мини-игр для сервера: «Королевская битва» (4 часа).
Теория (1 час).
Концепция Battle Royale в Minecraft. Использование плагинов: WorldEdit (создание арены), Multiverse (отдельные миры). Примеры плагинов: BattleRoyale, MCBattleRoyale.
Практика (3 часа).
Создание карты. Постройка арены с помощью WorldEdit (горы, укрытия). Генерация лута в сундуках через редактор. Установка BattleRoyale, настройка зоны смерти (команда /br zone). Анализ баланса: время сокращения зоны, количество лута.

Тема 6.3 Многопользовательский квест. Головоломки, требующие взаимодействия (4 часа).
Теория (1 час).
Основы проектирования квестов. Сюжетные линии, диалоги с NPC (плагин Citizens). Головоломки: активация рычагов, сбор артефактов, координация игроков. Использование Redstone и командных блоков.
Практика (3 часа).
Создание квеста. Настройка NPC через Citizens (диалоги, квестовые задания). Разработка головоломки с Redstone (дверь, открывающаяся при одновременном нажатии кнопок игроками). Выдача наград через CrateReloaded (сундуки с призами). Логирование прогресса игроков через Quests. Тестирование.

Тема 6.4 Плагины: добавление экономики (виртуальные деньги) (4 часа).
Теория (1 час).
Экономика в Minecraft. Популярные плагины: Vault (ядро), EssentialsX (базовые команды), Shopkeepers (торговцы). Балансировка: цены на ресурсы, античит-механизмы.
Практика (3 часа).
Установка Vault и EssentialsX. Настройка валюты (название, курс обмена). Создание магазина через ChestShop (ценники, условия продажи). Зарплата за квесты через Quests. Платные возможности (например, телепортация за деньги).

Тема 6.5 Командная разработка: парк аттракционов с ролями (строитель, программист) (6 часов).
Теория (2 часа).
Организация командной работы. Распределение ролей: строители (создание структур), программисты (настройка плагинов). Использование GitHub для совместной работы над мирами.
Практика (4 часа).
Планирование. Создание макета парка в Creative-режиме. Строители: возведение аттракционов (использование WorldEdit). Программисты: добавление механик (например, запуск горки по команде /start). Демонстрация.

Раздел 7. Креативное проектирование: 3D-моделирование и анимация (10 часов).
Тема 7.1 Основы 3D-моделирования в Minecraft (4 часа).
Теория (2 часа).
Инструменты для создания 3D-объектов: Tinkercad, Blockbench, Minecraft Structure Generator. Принципы моделирования: текстуры, геометрия, оптимизация для игровых миров. Импорт/экспорт моделей в Minecraft с помощью модов (например, MrCrayfish's Furniture Mod).
Практика (2 часа).
Создание собственного 3D-объекта (например, мебель, транспорт) в Tinkercad и интеграция его в Minecraft.

Тема 7.2 Анимация и визуальные эффекты (4 часа).
Теория (2 часа).
Основы анимации: ключевые кадры, циклы, плавность движений. Использование частиц и эффектов (дым, огонь, свет) через командные блоки и Python. Интеграция анимации в игровые механики (например, движущиеся платформы, "оживающие" NPC).
Практика (2 часа).
Создание анимированного моба (например, дракон с движущимися крыльями) с помощью Blockbench. Добавление визуальных эффектов к квесту: взрывы при завершении, сияние артефактов.

Тема 7.3 Интерактивное искусство: от пикселей к инсталляциям (2 часа).
Теория (1 час).
Minecraft как платформа для цифрового искусства: примеры миров-инсталляций, pixel art. Связь программирования и творчества: генеративные паттерны, алгоритмическое искусство.
Практика (1 час).
Создание анимированного арт-объекта с управлением через Python. Разработка 3D-модели в Blockbench. Использование мода Custom Model Data или ресурса-пака для интеграции модели в игру. Написание скрипта на Python, чтобы анимация запускалась при приближении игрока или нажатии кнопки.

Раздел 8. От идеи до реализации (14 часов).

Тема 8.1 «Мой выбор»: профессия программист (2 часа).

Теория (2 часа).

Определение программирования. Зачем нужен программист? Как стать программистом? Роль программиста в современном мире. Программисты в разных сферах. Современные профессии в IT-сфере и IT-профессий, которые могут появиться в ближайшем будущем.

Тема 8.2 Разработка идеи: масштабный проект (4 часа).

Теория (1 час).

Этапы создания масштабного проекта. Генерация идей. Планирование. Составление технического задания: цели проекта, используемые технологии (Python, MakeCode, Redstone), сроки. Создание MVP (минимально жизнеспособного продукта): базовая функциональность без детализации. Инструменты для организации работы. Примеры проектов.

Практика (3 часа).

Создание концепции проекта, составление технического задания, прототипирование, распледение ролей в команде (программист, дизайнер, тестировщик).

Тема 8.3 Реализация проекта. Консультации (6 часов).

Теория (1 час).

Методологии разработки, советы по эффективной реализации, инструменты.

Практика (5 часов).

Разработка основных систем, интеграция компонентов, тестирование и отладка, демо-версия и обратная связь. Педагог проводит индивидуальные консультации с обучающимися, которые требуют дополнительной помощи. Обсуждение возможных трудностей и путей их решения. Обучающиеся возвращаются к своим проектам и вносят изменения на основе полученной обратной связи. Педагог помогает по мере необходимости, отвечает на вопросы.

Тема 8.4 Итоговое занятие «Программирование в Minecraft». Презентация игр. Промежуточная аттестация (2 часа).

Практика (2 часа).

Итоговое занятие «Программирование в Minecraft». Промежуточная аттестация. Обучающиеся презентуют получившуюся игру. Отвечают на вопросы.

Раздел 3. Воспитательная деятельность

3.1. Цель, задачи, целевые ориентиры воспитания детей

Целью воспитания является развитие личности, самоопределение и социализация детей на основе социокультурных, духовно-нравственных ценностей и принятых в российском обществе правил и норм поведения в интересах человека, семьи, общества и государства, формирование чувства патриотизма, гражданственности, уважения к памяти защитников Отечества и подвигам Героев Отечества, закону и право-порядку, человеку труда и старшему поколению, взаимного уважения, бережного отношения к культурному наследию и традициям многонационального народа Российской Федерации, природе и окружающей среде (Федеральный закон от 29.12.2012 № 273-ФЗ «Об образованиив Российской Федерации», ст. 2, п. 2).

Задачами воспитания по ДООП «Программирование в Minecraft» являются:

- содействие развитию коммуникативных навыков, навыков сотрудничества и распределения обязанностей при работе в группе и обмене идеями по проектам в области программирования;
- воспитание интереса к современным технологиям, развитие информационной грамотности;
- формирование самодисциплины и ответственности, так как работа над программами и проектами требует самодисциплины, пунктуальности и ответственности за результат.

Основные целевые ориентиры определяются на основе российских базовых конституционных ценностей с учётом целевых ориентиров результатов воспитания обучающихся в общеобразовательных организациях, что обеспечивает единство содержания воспитания, воспитательной деятельности, воспитательного пространства во всех образовательных организациях, в которых обучаются дети.

Основные целевые ориентиры воспитания детей ДООП «Программирование в Minecraft»:

- развитие у обучающихся любознательности, терпения, настойчивости;
- формирование навыков коллективной работы и креативного мышления через совместные проекты и задания;
- формирование у обучающихся воли, дисциплинированности; уверенности в своих силах и способностях, а также развитие креативности и фантазии;
- формирование ценностей технической безопасности и контроля;
- развитие трудолюбия, ответственности и самостоятельности;
- ориентация на осознанный выбор сферы профессиональных интересов, профессиональной деятельности в российском обществе с учётом личных жизненных планов, потребностей семьи, общества.

3.2. Формы и методы воспитания

Дополнительное образование имеет практико-ориентированный характер и ориентировано на свободный выбор педагогом таких видов и форм воспитательной деятельности, которые способствуют формированию и развитию у детей индивидуальных способностей и способов деятельности, объективных представлений о мире, окружающей действительности, внутренней мотивации к творческой деятельности, познанию, нравственному поведению.

Основными формами воспитания по методу воздействия в ДООП «Программирование в Minecraft» являются:

- получение информации об истории развития языков программирования; этапах развития программирования; изучение биографий выдающихся программистов и их вклада в развитие программирования, как источник формирования у детей сферы интересов, этических установок, личностных позиций и норм поведения (важно, чтобы дети не только получали эти сведения от педагога, но и сами осуществляли работу с информацией: поиск, сбор, обработку, обмен и т. д.):
- практические занятия обучающихся - развитие коммуникативных навыков при работе в группе и обмене идеями при выполнении проектов в рамках образовательной программы;
- участие в воспитательных мероприятиях коллектива/объединения, где проявляются и развиваются личностные качества: эмоциональность, активность, нацеленность на успех, готовность к командной деятельности и взаимопомощи;
- проведение практических работ и заданий - поддержка самостоятельности и инициативности обучающихся, развитие умения принимать ответственные решения и доводить задачи до конца;
- участие в конкурсах способствует закреплению ситуации успеха, развитию рефлексивных и коммуникативных умений, ответственности, благоприятно воздействуют на эмоциональную сферу детей.

В воспитательной деятельности с обучающимися по ДООП «Программирование в Minecraft» используются следующие методы воспитания:

- метод убеждения (рассказ, разъяснение, внушение);
- метод положительного примера (педагога и других взрослых, детей);
- метод упражнений (приучения);
- методы стимулирования, поощрения (индивидуального и публичного);
- методы самовоспитания, развития самоконтроля и самооценки;
- методы воспитания воздействием группы, в коллективе.

3.3. Условия осуществления воспитательного процесса, анализ результатов

Воспитательный процесс осуществляется в условиях организации деятельности детского коллектива на основной учебной базе реализации программы в организации дополнительного образования детей в

соответствии с нормами и правилами работы организации, а также на выездных базах, площадках, мероприятиях в других организациях с учётом установленных правил и норм деятельности на этих площадках.

Анализ результатов воспитания проводится в процессе педагогического наблюдения за поведением детей, их общением, отношениями детей друг с другом, в коллективе, их отношением к педагогам, к выполнению своих заданий по программе. Косвенная оценка результатов воспитания, достижения целевых ориентиров воспитания по программе проводится путём опросов родителей в процессе реализации программы (отзывы родителей, интервью с ними) и после её завершения (итоговые исследования результатов реализации программы за учебный период, учебный год).

Анализ результатов воспитания по программе предусматривает не определение персонифицированного уровня воспитанности, развития качеств личности конкретного обучающегося, а получение общего представления о воспитательных результатах реализации программы, продвижении в достижении определённых в программе целевых ориентиров воспитания, о влиянии воспитательных мероприятий в рамках реализации программы на коллектив обучающихся.

3.4. Календарный план воспитательной работы Программе

№ п/п	Название мероприятия, события	Цель мероприятия	Сроки	Практический результат и информационный продукт, иллюстрирующий успешное достижение цели события
1.	Игра: «Давайте познакомимся»	знакомство обучающихся с традициями коллектива; создание благоприятной психологической атмосферы для дальнейшего обучения в объединении	сентябрь	размещение фото- и видеоматериалов, постов с проведённого мероприятия на странице объединения ВКонтакте; в официальной группе ВКонтакте филиала МАУДО «ДПШ»
2.	«История Minecraft» беседа о развитии Minecraft и его влиянии на игровую индустрию	воспитание интереса к программированию и творчеству	ноябрь	размещение фото- и видеоматериалов, постов с проведённого мероприятия на странице объединения ВКонтакте; в официальной группе ВКонтакте филиала МАУДО «ДПШ»
3.	Фестиваль «Pro Технологии». Конкурс компьютерной графики	воспитание ответственности за порученное дело, аккуратности, взаимовыручки, коллективизма и «здорового духа» соперничества; популяризация занятий компьютерной графикой	декабрь, апрель	размещение фото- и видеоматериалов, постов с проведённого мероприятия на странице объединения ВКонтакте; в официальной группе ВКонтакте филиала МАУДО «ДПШ»

4.	День открытых дверей	воспитание чувства уважения к традициям ДПШ и чувства сопричастности к успехам коллектива	май	размещение фото- и видеоматериалов, постов с проведённого мероприятия на странице объединения ВКонтакте; в официальной группе ВКонтакте филиала МАУДО «ДПШ»

Раздел 4. Формы аттестации и оценочные материалы

Текущий контроль: практическая работа.

В рамках текущего контроля по разделу **«Введение в Minecraft и Make Code»** обучающиеся должны создать автоматическую ферму пшеницы, добавить редстоун-механизм для автоматического полива (ведро с водой). Цикл сбора урожая каждые 10 минут.

Форма контроля	Уровень освоения материала	Зачётные требования
Практическая работа. Раздел «Введение в Minecraft и Make Code»	низкий	обучающийся демонстрирует низкий уровень заинтересованности в освоении разделов Программы: ферма требует ручного вмешательства (например, сбор урожая вручную), автоматизация частичная или не работает, механизм полива работает нестабильно (например, вода не убирается, заливает ферму), вода покрывает только часть фермы, растения не получают достаточно влаги, цикл сбора не соответствует 10 минутам (отклонение ±3+ минут), скрипты отсутствуют или не интегрированы с фермой, ферма занимает много места, провода и механизмы не упорядочены. Обучающийся не может объяснить принцип работы фермы или указать на ошибки
	средний	обучающийся демонстрирует достаточную заинтересованность в освоении разделов Программы: урожай собирается автоматически, но цикл нестабилен (например, задержки больше/меньше 10 минут), вода активируется по таймеру, но требует ручного вмешательства для перезапуска, вода распределяется равномерно, но есть избыточное использование ресурсов (например, много лишних блоков), цикл близок к 10 минутам (±1-2 минуты), но нет визуальной индикации (например, редстоун-часы), простые скрипты для активации механизмов, но без синхронизации циклов, конструкция функциональна, но есть лишние элементы или нелогичное расположение блоков. Обучающийся может объяснить основные элементы (например, как активируется вода), но без деталей о синхронизации
	высокий	обучающийся демонстрирует высокую заинтересованность в освоении Программы: полностью автоматизированный цикл сбора каждые 10 минут, включая повторную посадку семян, полив полностью автоматизирован с использованием редстоун-схем (например, поршни, распределители воды), цикл синхронизирован со сбором урожая, оптимальное использование воды (например, каналы или вертикальное распространение), вся ферма поливается без перерасхода, соблюден точный цикл 10 минут, реализованный через редстоун-повторители, командные блоки или таймеры в MakeCode, программирование логики таймера, автоматического полива и сбора через MakeCode с обработкой событий работает, компактный дизайн, эстетичное использование декоративных блоков. Обучающийся четко описывает все компоненты

В рамках текущего контроля по разделу **«Игровая механика и автоматизация»** обучающиеся создать умный дом. Настроить освещение, включающееся при

приближении, ловушки: ямы с лавой, активируемые при вторжении, видеонаблюдение: камеры (созданные из стекла и табличек с текстом).

Форма контроля	Уровень освоения материала	Зачётные требования
Практическая работа. Раздел «Игровая механика и автоматизация»	низкий	обучающийся демонстрирует низкий уровень заинтересованности в освоении разделов Программы: освещение включается/выключается вручную (например, рычаг или кнопка), статичные ямы с лавой без механизма активации, камеры созданы из стекла и табличек с текстом, но не имеют функциональности (просто декорация), компоненты умного дома работают изолированно (например, свет не связан с ловушками), ловушки могут навредить владельцу (например, лава вытекает за пределы ямы), дом построен без детализации, механизмы выглядят неаккуратно (например, провода торчат наружу). Студент не может продемонстрировать работу систем или объяснить их устройство
	средний	обучающийся демонстрирует достаточную заинтересованность в освоении разделов Программы: датчики приближения (например, нажимные пластины, триггеры из редстоуна) активируют свет, но есть задержки или нестабильность, ловушки активируются при приближении (например, поршни открывают яму), но нет защиты от ложных срабатываний; «камеры» имитируют обзор, есть связь с ловушками через редстоун; частичная интеграция систем, функциональный дизайн, но минимальная эстетика, ловушки безопасны для игрока, но иногда дают сбои. Объяснены основные механизмы, но без технических детале
	высокий	обучающийся демонстрирует высокую заинтересованность в освоении Программы: плавное включение света при приближении игрока, интеграция с другими системами, реализованы сложные механизмы (например, задержка срабатывания, скрытые ловушки), автоматический сброс лавы после активации; присутствует логическая интеграция с системой безопасности; реалистичный интерьер, скрытые редстоун-схемы, использование декоративных элементов (ковры, картины, растения), все системы защищены от ошибок. Четкая демонстрация всех функций, схемы редстоуна показаны и объяснены, предложены идеи для улучшения

В рамках текущего контроля по разделу **«Продвинутый Python»** обучающиеся должны создать «Умного города» с управляемым освещением (уличные фонари включаются автоматически в реальном времени, синхронизированном с вашим часовым поясом), автоматизированным транспортом (вагонетки перемещаются между станциями, избегая столкновений и выбирая оптимальный маршрут). Добавить дашборд в Minecraft, отображающий данные через таблички: время суток, погода, активность мобов. Настроить энергопотребление (солнечные панели активны днём, ветрогенераторы — при сильном ветре (данные из API).

Форма контроля	Уровень освоения материала	Зачётные требования
Практическая работа. Раздел «Продвинутый Python»	низкий	обучающийся демонстрирует низкий уровень заинтересованности в освоении разделов Программы: освещение включается/выключается по фиксированному времени без синхронизации с реальным часовым поясом, вагонетки движутся по фиксированным рельсам без избегания столкновений; таблички отображают статичные данные; солнечные панели и ветрогенераторы активируются по фиксированным условиям (без API); данные погоды введены вручную; системы работают изолированно (освещение не зависит от энергии). Обучающийся не может продемонстрировать работу ключевых систем
	средний	обучающийся демонстрирует достаточную заинтересованность в освоении разделов Программы: синхронизация с часовым поясом, но задержки в обновлении (например, раз в 30 минут); в задании базовое обнаружение препятствий (датчики из редстоуна), простые маршруты; динамическое обновление времени, погоды и активности мобов присутствуют, но с ошибками; интеграция с погодным API, но ограниченная обработка данных; у систем частичная связь. Обучающийся может объяснить основные функции, но без деталей
	высокий	обучающийся демонстрирует высокую заинтересованность в освоении Программы: в задании точная синхронизация в реальном времени (библиотеки datetime, pytz), адаптация к сезонным изменениям освещения; использованы алгоритмы поиска пути, динамическое изменение маршрутов, интеграция с временем суток; Реальное время, погода из API, подсчет мобов в зоне, визуализация через цветовые индикаторы; оптимизация энергопотребления на основе реальных данных, баланс между источниками; произведена полная синхронизация: энергия влияет на освещение и транспорт, активность мобов зависит от времени суток. Обучающийся демонстрирует все компоненты

Промежуточная аттестация: защита проекта.

Обучающиеся разрабатывают и защищают проект на основании заданных тем в рамках изученной образовательной Программы.

Форма контроля	Уровень освоения материала	Система баллов
Защита проекта	низкий	1-5 баллов
	средний	6-14 баллов
	высокий	15-18 баллов

Метапредметные результаты освоения Программы оцениваются по карте педагогического наблюдения (приложение 3).

Личностные результаты освоения Программы оцениваются специально подобранным психологическим инструментарием (приложение 3)

Раздел 5. Организационно-педагогические условия реализации Программы

5.1 Методические материалы

Форма обучения – очная; с применением дистанционных образовательных технологий.

Основные формы учебных занятий:

1.Беседа – диалогический метод обучения, при котором преподаватель путем постановки тщательно продуманной системы вопросов подводит обучающегося к пониманию нового материала или проверяет усвоение уже изученного.

2.Практическое занятие – это занятие, проводимое под руководством преподавателя, направленное на углубление научно-теоретических знаний и овладение определенными методами самостоятельной работы, которое формирует практические умения.

3.Защита проектов – это педагогическая технология, стержнем которой является самостоятельная деятельность детей – исследовательская, познавательная, продуктивная, в процессе которой обучающийся познаёт окружающий мир и воплощает новые знания в реальные продукты.

4. Объяснение – словесный метод обучения, заключающийся в истолковании закономерностей, существенных свойств изучаемых объектов, отдельных понятий, фактов или явлений при изучении, как правило, теоретического материала различных наук, при раскрытии коренных причин и следствий в явлениях природы и общественной жизни.

5. Визуализация - это метод обучения, при котором информация передается через графические, интерактивные или мультимедийные средства.

Методы обучения: объяснительно-иллюстративный, репродуктивный.

Форма организации образовательного процесса: групповая.

Методы воспитания: убеждение, поощрение, стимулирование, мотивация.

Педагогические технологии: технология группового обучения, технология коллективного взаимообучения, здоровьесберегающие технологии и др.

Список литературы

1. Иванова, Г.С. Основы программирования: Учебник для вузов. — Москва:Изд-во МГТУ им.Н.Э.Баумана, 2002. — 416 с.
2. 365 задач на смекалку. — Москва: АСТ-Пресс, 2007. — 272 с. 3. Олехник С.Н., Нестеренко Ю.В., Потапов М.К. Старинные занимательные задачи. — Москва: Дрофа, 2006. - 176 с.
3. Шведова, Л.М. Развитие логического мышления и интеллекта. — Москва: Бао-Пресс, 2006. — 352 с.
4. Окулов, С.М., Ашихмина, Т.В., Бушмелева, Н.А. Задачи по программированию. — Москва: Бином. Лаборатория знаний, 2020. – 823 с:
5. Винницкий, Ю. А Scratch и Arduino для юных программистов и конструкторов — СПб.: БХВ-Петербург, 2018. — 176 с.

6. Гаррисон, Д. Программирование с Minecraft: Создай свой мир с помощью Python. — Москва: Питер, 2021. — 256 с.
7. Мак-Манус, К. Официальная книга идей для Minecraft. — Москва: Эксмо, 2020. — 96 с.
8. Проктор, Н. Minecraft в образовании: Как использовать игру в обучении. — СПб.: БХВ-Петербург, 2019. — 144 с.
9. Бриггс, Дж. Python для детей: Играй и программируй. — М.: Манн, Иванов и Фербер, 2022. — 320 с.
10. Стэмпер, Д. Изучаем Python через Minecraft. — М.: ДМК Пресс, 2023. — 180 с.
11. Minecraft Education Edition. Официальное руководство для преподавателей [Электронный ресурс]. — Режим доступа: https://education.minecraft.net (дата обращения: 10.04.2025).
12. Microsoft MakeCode. Документация по программированию в Minecraft [Электронный ресурс]. — Режим доступа: https://minecraft.makecode.com (дата обращения: 10.04.2025).
13. Raspberry Pi Foundation. Программирование на Python с Minecraft Pi [Электронный ресурс]. — Режим доступа: https://projects.raspberrypi.org (дата обращения: 10.04.2025).
14. Иванов, А.А. Геймификация в обучении: Minecraft как инструмент развития логического мышления // Цифровое образование. — 2022. — № 4. — С. 45–52.
15. Smith, J. Using Minecraft to Teach Coding: A Case Study // Journal of Educational Technology. — 2021. — Vol. 15. — P. 78–89.
16. Code.org. Курс «Программирование с Minecraft» [Электронный ресурс]. — Режим доступа: https://code.org/minecraft (дата обращения: 10.04.2025).
17. Уроки по Minecraft и Python на платформе Udemy [Электронный ресурс]. — Режим доступа: https://www.udemy.com (дата обращения: 10.04.2025).

5.2 Материально-техническое обеспечение и оснащенность образовательного процесса по дополнительной общеобразовательной общеразвивающей программе «Программирование»

№ п/п	Наименование основного оборудования	Кол-во единиц
	I. Технические средства обучения	
1.	персональный компьютер (рабочее место педагога)	1
2.	персональный компьютер (рабочее место учащегося)	12
3	внешний накопитель информации	1
4.	мультимедийный проектор	1
II.	Информационно-коммуникационные средства (программные средства)	
1.	операционная система	Windows
2.	антивирусная программа	любая
3.	пакет Microsoft Office	Word, Power Point, Publisher, Excel
4.	Программное обеспечение для работы в среде Minecraft	Minecraft
	III. Учебно-практическое оборудование	
1.	ручки	20-25
2.	бумага	200
	IV. Мебель	
1.	стол (для учащихся)	15
2.	компьютерные кресла (для учащихся)	15
3.	стол преподавателя	1
4.	стеллаж для хранения оборудования	1
5.	стол для оборудования	1

Приложение 1

Календарный учебный график образовательной программы «Программирование в Minecraft»

Комитет по делам образования города Челябинска

Муниципальное автономное учреждение дополнительного образования "Дворец пионеров и школьников им. Н.К. Крупской г. Челябинска"

Утверждаю: Директор МАУДО "ДПШ" Ю.В. Смирнова 1 сентября 2025 г.

Первый заместитель директора _____ А.А. Завьялов 1 сентября 2025 г.

Календарный учебный график

Филиала на 2025-2026 год

название программы, группа	Сентябрь					Октябрь				Ноябрь					Декабрь				Январь				Февраль				Март				Апрель				Май				Июнь				Июль				Август			
	1-7 сентября	8-14 сентября	15-21 сентября	22-28 сентября	29 сентября -5 октября	6-12 октября	13-19 октября	20-26 октября	27 октября-2 ноября	3-9 ноября	10-16 ноября	17-23 ноября	24-30 ноября	1-7 декабря	8-14 декабря	15-21 декабря	22-28 декабря	29 декабря-4 января	5-11 января	12-18 января	19-25 января	26 января-1 февраля	2-8 февраля	9-15 февраля	16-22 февраля	23 февраля-1 марта	2-8 марта	9-15 марта	16-22 марта	23-29 марта	30 марта-5 апреля	6-12 апреля	13-19 апреля	20-26 апреля	27 апреля-3 мая	4-10 мая	11-17 мая	18-24 мая	25-31 мая	1-7 июня	8-14 июня	15-21 июня	22-28 июня	29 июня-5 июля	6-12 июля	13-19 июля	20-26 июля	27 июля-2 августа	3-9 августа	10-16 августа	17-23 августа	24-30 августа
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52
полугодие	первое полугодие 01.09.2025 - 31.12.2025																		второе полугодие 01.01.2026 - 31.05.2026																																			
"Программирование в Minecraft"	4	4	4	4	4/т	4	4	4	4	4	4	4	4	4	4/т	4	4	в	4	4	4	4	4	4	4	4	4	4/т	4	4	4	4	4	4	4	4	4	4	4/п												

т- текущий контроль

п-промежуточная аттестация

в-выходные праздничные дни

Приложение 2

КАРТОЧКА
дополнительной общеобразовательной общеразвивающей программы «Программирование в Minecraft»

Наименование	Содержание
название ДООП/модуля (каждый модуль отдельно)	«Программирование в Minecraft»
краткое название ДООП/модуля	«Программирование в Minecraft»
направленность Программы	техническая
краткое описание	Программа предполагает знакомство со средой программирования «Minecraft», с текстовым языком программирования «Python» и с визуальным языком программирования «MakeCode». В ходе освоения Программы обучающиеся получат возможность изучить основы алгоритмики, принципы создания программ и основные элементы создания компьютерных игр
содержание Программы учебного плана (наименование разделов и тем)	1. Раздел 1. Введение в Minecraft и Make Code 1.1 Вводное занятие «Мой Дворец»: история Дворца пионеров. Инструктаж по технике безопасности. Знакомство с интерфейсом Minecraft (Creative/Survival) 1.2 Основы командных блоков. Создание телепортов 1.3 Введение в MakeCode: события и циклы. Проект: «Автоматическая дверь» 1.4 Переменные в MakeCode 1.5 Создание лабиринта с ловушками 1.6 Ферма с автоматическим сбором урожая. Текущий контроль 2. Раздел 2. Основы Python для Minecraft 2.1 Установка и настройка Python + библиотека mcpi. Знакомство с синтаксисом 2.2 Работа с координатами: построение геометрических фигур (пирамида, спираль) 2.3 Условия if-else. Дождь из TNT при нажатии кнопки 2.4 Циклы for и while. Спираль из алмазов 2.5 Функции. Генератор случайных лабиринтов 2.6 Работа с массивами 2.7 Интеграция Python и MakeCode. Автоматический маяк 3. Раздел 3. Игровая механика и автоматизация 3.1 Redstone: основы логических схем. Секретная дверь 3.2 Создание мини-игры «Сапёр» на Python. Генерация мин, подсчёт очков 3.3 Разработка квеста: «Поиск сокровищ» с использованием условий и циклов 3.4 Автоматизация: «Ферма с самонаводящимися вагонетками» 3.5 Многопользовательские игры: «Королевская битва» (зоны, бонусы) 3.6 Создание умного дома с системой безопасности. Текущий контроль 4. Раздел 4. Новые функции и расширенные возможности

	4.1 Датчики и события в Minecraft 4.2 Графические интерфейсы (GUI) 4.3 Многопользовательские взаимодействия 4.4 Интеграция внешних данных 4.5 Сложные механизмы на Redstone 5. Раздел 5. Продвинутый Python 5.1 Классы и объекты. Создание NPC (персонаж с диалогами) 5.2 Работа с файлами: сохранение прогресса игрока 5.3 API Minecraft: взаимодействие с сервером. Онлайн-чат 5.4 Игра-РПГ: инвентарь, квесты, система уровней 5.5 Интеграция с базами данных (SQLite). Рейтинг игроков 5.6 Многопоточность. Параллельное выполнение задач 5.7 Создание модов. Добавление новых предметов и блоков 5.8 Искусственный интеллект. Простые алгоритмы для NPC (поиск пути) 5.9 Сетевые запросы. Получение данных из внешних API (погода, время) 5.10 Мини-игра с ИИ. «Охота на мобов» с умными противниками 5.11 «Умный город» с API и ИИ. Текущий контроль 6. Раздел 6. Многопользовательские игры 6.1 Настройка локального сервера (Minecraft Java + Spigot) 6.2 Создание мини-игр для сервера: «Королевская битва» 6.3 Многопользовательский квест. Головоломки, требующие взаимодействия 6.4 Плагины: добавление экономики (виртуальные деньги) 6.5 Командная разработка: парк аттракционов с ролями (строитель, программист) 7. Раздел 7. Креативное проектирование: 3D-моделирование и анимация 7.1 Основы 3D-моделирования в Minecraft 7.2 Анимация и визуальные эффекты 7.3 Интерактивное искусство: от пикселей к инсталляциям 8. Раздел 8. От идеи до реализации 8.1 «Мой выбор»: профессия программист 8.2 Разработка идеи: масштабный проект 8.3 Реализация проекта. Консультации 8.4 Итоговое занятие «Программирование в Minecraft». Презентация игр. Промежуточная аттестация
ключевые слова для поиска Программы	программирование, Minecraft, MakeCode, Python, проектная работа
цель и задачи	обучение обучающихся основам программирования с использованием MakeCode и Python в результате разработки игрового контента на базе Minecraft.
Результат	самостоятельная разработка как индивидуального, так и командного игрового проекта для среды Minecraft
материальная база	учебный класс с доступом в Интернет, оборудованный учебной мебелью, компьютерами, ПО
требования к состоянию здоровью	нет
наличие медицинской справки для зачисления	нет
возрастной диапазон	10-13
число учащихся в группе	15
способ оплаты	бюджет
Продолжительность	1 год
общее количество и количество часов в неделю	148/4

Приложение 3

Контрольно-измерительные материалы по дополнительной общеобразовательной общеразвивающей программе «Программирование в Minecraft»

Текущий контроль
Форма контроля: практическая работа.
Задание для раздела «Введение в Minecraft и Make Code»: создать автоматическую ферму пшеницы, добавить редстоун-механизм для автоматического полива (ведро с водой). Цикл сбора урожая каждые 10 минут.
Работа обучающихся оценивается в соответствии со следующими критериями:

Уровень освоения материала	Зачетные требования
низкий	— ферма требует ручного вмешательства, механизмы работают нестабильно (1 балл); — цикл сбора не соответствует 10 минутам (отклонение больше 3 минут) (1 балл); — скрипты отсутствуют или не интегрированы с фермой (1 балл)
средний	— урожай собирается автоматически, но с ошибками; механизмы полива работают нестабильно (2 балла); — в задании используются простые скрипты для активации механизмов, но без синхронизации циклов (2 балла); — обучающий может объяснить ход выполнения задания, но поверхностно (2 балла)
высокий	— задание выполнено в полном объеме, все системы работают исправно и не требуют вмешательства (3 балла); — цикл сбора четко 10 минут, соблюден компактный дизайн и эстетичное использование декоративных блоков. (3 балла); — четко описан процесс, продемонстрировано понимание этапов работы (3 балла)

0-3 баллов - обучающийся освоил раздел Программы на низком уровне;
4-6 баллов - обучающийся освоил раздел Программы на среднем уровне;
7-9 баллов - обучающийся освоил раздел Программы на высоком уровне.

Текущий контроль
Форма контроля: практическая работа
Задание для раздела «Игровая механика и автоматизация»: создать умный дом. Настроить освещение, включающееся при приближении, ловушки: ямы с лавой, активируемые при вторжении, видеонаблюдение: камеры (созданные из стекла и табличек с текстом).
Работа обучающихся оценивается в соответствии со следующими критериями:

Уровень освоения материала	Зачетные требования
низкий	— освещение включается/выключается вручную; статичные ямы с лавой без механизма активации (1 балл); — компоненты умного дома работают изолированно (1 балл); — дом построен без детализации, механизмы выглядят неаккуратно, ловушки вредят владельцу. (1 балл)
средний	— в задании присутствуют датчики приближения, которые активируют свет, но есть задержка или нестабильность. (2 балла); — камеры имитируют обзор, есть связь с ловушками через редстоун. (2 балла); — частичная интеграция систем, соблюден функциональный дизайн, но минимальная эстетика (2 балла)
высокий	— задание выполнено в полном объеме, системы работают исправно и не вредят владельцу (3 балла); — реалистичный интерьер, скрытые редстоун-схемы, использование декоративных элементов. (3 балла); — обучающий четко демонстрирует все функции, схемы редстоуна показывает и объясняет (3 балла)

0-3 баллов - обучающийся освоил раздел Программы на низком уровне;
4-6 баллов - обучающийся освоил раздел Программы на среднем уровне;
7-9 баллов - обучающийся освоил раздел Программы на высоком уровне.

Текущий контроль
Форма контроля: практическая работа
Задание для раздела «Продвинутый Python»: создать «Умного города» с управляемым освещением (уличные фонари включаются автоматически в реальном времени, синхронизированном с вашим часовым поясом), автоматизированным транспортом (вагонетки перемещаются между станциями, избегая столкновений и выбирая оптимальный маршрут). Добавить дашборд в Minecraft, отображающий данные через таблички: время суток, погода, активность мобов. Настроить энергопотребление (солнечные панели активны днём, ветрогенераторы — при сильном ветре (данные из API).
Работа обучающихся оценивается в соответствии со следующими критериями:

Уровень освоения материала	Зачетные требования
низкий	— системы не синхронизированы, объекты двигаются без избегания столкновений и освещение включается по фиксированному времени (1 балл); — системы работают изолировано (1 балл); — обучающийся не может продемонстрировать работу ключевых систем (1 балл)
средний	— присутствует синхронизация с часовым поясом, базовое обнаружение препятствий соблюдено (2 балла); — динамическое обновление времени, погоды и активности мобов соблюдены, но с ошибками (2 балла); — обучающийся понимает основные механики, но объясняет их без деталей (2 балла)
высокий	— задание выполнено в полном объеме (3 балла); — полная синхронизация: энергия влияет на освещение и транспорт, активность мобов зависит от времени суток. (3 балла); — обучающийся демонстрирует все компоненты. (3 балла)

0-3 баллов - обучающийся освоил раздел Программы на низком уровне;

4-6 баллов - обучающийся освоил раздел Программы на среднем уровне;
7-9 баллов - обучающийся освоил раздел Программы на высоком уровне.

Промежуточная аттестация

Форма контроля: защита проекта.

Темы проектов:

1. Постройка города с автоматическими системами: светофоры, лифты, самооткрывающиеся двери, фермы с автосбором урожая.
2. Квест «Спасение деревни», игроку необходимо выполнить задания, чтобы разблокировать портал и спасти жителей.
3. Создание музея технологий с интерактивными экспонатами.
4. Разработка умного дома с системой безопасности и ловушками для мобов.
5. Создание многопользовательской игры с зонами, бонусами и системой подсчета очков.
6. Разработка игры в жанре РПГ с инвентарем и квестами.
7. Создание викторины на время, тема школьных предметов или любая другая, при выборе верного ответа открывается дверь.
8. Разработка лабиринта времени с динамичными стенами.
9. Создание виртуальной экскурсии.
10. Разработка автоматизированного зоопарка с управляемыми животными.
11. Создание экологической станции переработки. Игрок бросает предметы в контейнеры, а станция автоматически перерабатывает их в ресурсы.
12. Разработка игры «Змейка», где игроку предстоит управлять блоком, собирая ресурсы для роста.

Критерии оценивая проекта:

1. В игре использованы изученные языки (Python, MakeCode) для автоматизации процессов:

3 балла - все процессы автоматизированы с использованием изученных языков, код оптимален;

2 балла - автоматизация присутствует, но не все процессы реализованы или код можно улучшить;

1 балл - автоматизация не выполнена или использованы другие языки.

2. Осуществлена работа с Redstone: логические схемы, автоматизация механизмов:

3 балла - логические схемы сложные и функциональные, механизмы работают без сбоев;

2 балла - схемы работают, но есть небольшие недочеты или не все механизмы автоматизированы;

1 балл - логические схемы не работают или не реализованы.

3. В проекте применены алгоритмы (циклы, условия, функции).

Код чистый, оставлены комментарии для участников проекта:

3 балла - алгоритмы использованы корректно, код чистый, есть комментарии;

2 балла - алгоритмы применены, но есть недочеты в коде или недостаточно комментариев;

1 балл - алгоритмы не используются или код нечитабелен.

4. Проект работает без критических ошибок. Все заявленные функции реализованы и взаимодействуют между собой:

3 балла - проект работает стабильно, все функции реализованы и взаимодействуют;

2 балла - проект работает, но есть незначительные ошибки или недоработки;

1 балл - проект имеет критические ошибки, которые мешают его работе.

5. В проекте произведена детализация мира: использование декоративных блоков, анимаций, эффектов. Оригинальность идеи и визуальное оформление:

3 балла - проект визуально привлекателен, оригинальный и продуманный;

2 балла - визуальные элементы присутствуют, но не все аспекты проработаны;

1 балл - проект выглядит неаккуратно, недостаточно деталей или оригинальности.

Критерии оценки защиты проекта

№ п/п	Наименование показателя	Количество баллов
1.	Умение выделить цель и задачи работы	1-3 балла
2.	Умение представить содержание работы	1-3 балла
3.	Владение понятийным аппаратом	1-3 балла
4.	Умение следовать алгоритму выступления	1-3 балла
6.	Умение давать аргументированные ответы на вопросы	1-3 балла

Оценочная шкала:

3 балла - превосходное выполнение критерия, все аспекты учтены и реализованы на высоком уровне;

2 балла - критерий выполнен, но есть незначительные недочеты или недостатки;

1 балл - критерий выполнен частично или с серьезными ошибками.

Итоговая оценка:

Необходимо суммировать баллы по всем критериям, чтобы получить общую оценку для каждого обучающегося. Максимально возможный балл - 33, минимально возможный - 1. Эта шкала поможет объективно оценить достижения обучающихся в рамках образовательной программы «Программирование в Minecraft».

1-11 баллов - обучающийся освоил образовательную программу на низком уровне;

12-24 балла - обучающийся освоил образовательную программу на среднем уровне;

25-33 баллов - обучающийся освоил образовательную программу на высоком уровне.

Карта педагогического наблюдения метапредметных результатов

№ п/п	Фамилия, имя обучающегося	Критерии оценки										
		Развитие навыков постановки цели, планирования и осуществления деятельности по ее достижению, коррекции своих действий в изменяющейся ситуации и соотнесения своих действий с результатом на основе самоанализа				Развитие навыков конструктивного взаимодействия внутри коллектива на основе принятых норм взаимоотношений и освоение различных социальных ролей, умения работать на общий результат и нести ответственность за свои обязанности и поручения.				Развитие навыков бесконфликтного и конструктивного общения с окружающими посредством освоения различных средств коммуникации и способов саморегуляции своего поведения		
		Развиты навыки планирования своей работы	Умеет нести ответственность за результаты действий	Умеет определять способы действия в рамках предложенных условий и требований	Способен ставить цель и выбирать пути её достижения	Умеет взаимодействовать со сверстниками и педагогом	Развита социальная компетентность, готовность к осуществлению общественно значимой деятельности	Владеет различными социальными ролями	Владеет и применяет нормы взаимоотношения в коллективе	Умеет взаимодействовать со сверстниками и педагогом	Владеет различными средствами коммуникации	Владеет навыками саморегуляции в процессе общения

+ 1 – владеет в совершенстве
0 – средний уровень
- 1 – не владеет

Педагог дополнительного образования _____

Анкета определения сформированности знаниевого компонента личностных результатов дополнительной общеобразовательной программы

Ценностное основание/ориентир: Знания

№	Утверждение/основание/вопрос	Варианты ответа (подчеркните выбранный)
1.	Стремление к знанию – одна из основных черт человека	4 – полностью согласен (-а) 3 – в общем, это верно 2 – это не совсем так 1 – это неверно
2.	Каждое полученное знание несёт в себе цель и значимость, пусть даже оно покажется слишком простым	4 – полностью согласен (-а) 3 – в общем, это верно 2 – это не совсем так 1 – это неверно
3.	Самообразование — это изучение новой информации и получение знаний, навыков самостоятельно	4– полностью согласен (-а) 3 – в общем, это верно 2 – это не совсем так 1 – это неверно

7 – 12 баллов – показатель полностью сформирован
5 – 6 баллов – показатель частично сформирован
0 – 4 баллов - показатель не сформирован

Ценностное основание/ориентир: Человек как представитель моего социального окружения

№	Утверждение/основание/вопрос	Варианты ответа (подчеркните выбранный)
1.	Общение для человека - это главное условие его психического и социального становления	4 –полностью согласен (-а) 3 – в общем, это верно 2 – это не совсем так 1 – это неверно
2.	Коммуникация – это конструктивный процесс взаимодействия между людьми или их группами с целью передачи информации либо обмена сведениями	4– полностью согласен (-а) 3 – в общем, это верно 2 – это не совсем так 1 – это неверно
3.	Дружба – это искренние, бескорыстные взаимоотношения, построенные на доверии и взаимном уважении	4– полностью согласен (-а) 3 – в общем, это верно 2 – это не совсем так 1 – это неверно
4.	Командная работа — это огромная возможность для личностного и профессионального роста всех членов команды	4– полностью согласен (-а) 3 – в общем, это верно 2 – это не совсем так 1 – это неверно

10 – 16 баллов – показатель полностью сформирован
6 – 9 баллов – показатель частично сформирован
0 – 5 баллов – показатель не сформирован

Ценностное основание/ориентир: Труд

№	Утверждение/основание/вопрос	Варианты ответа (подчеркните выбранный)
1.	Труд нужен человеку для саморазвития, получения каких-то новых навыков или знаний	4 -- полностью согласен (-а) 3 – в общем, это верно 2 – это не совсем так 1 – это неверно
2.	Труд нужен, чтобы получать деньги	4 -- полностью согласен (-а) 3 – в общем, это верно 2 – это не совсем так 1 – это неверно
3.	Когда, ты трудишься, ты делаешь себе лучше	4 – полностью согласен (-а) 3 – в общем, это верно 2 – это не совсем так 1 – это неверно
4.	Труд является существенным признаком отличия человека от животного	4 – полностью согласен (-а) 3 – в общем, это верно 2 – это не совсем так 1 – это неверно

10 – 16 баллов - показатель полностью сформирован
6 – 9 баллов – показатель частично сформирован
0 – 5 баллов – показатель не сформирован