

Управление образования
администрации городского округа город Выкса
Муниципальное автономное учреждение дополнительного образования
«Центр молодежных инженерных и научных компетенций «Квант ОМК»

ПРИНЯТО

на Педагогическом совете
МАУ ДО «ЦМИНК «Квант ОМК»
Протокол № 1 от «12» января 2026 г.



УТВЕРЖДЕНО

Директор

МАУ ДО «ЦМИНК «Квант ОМК»

Блинова А.Н.

Приказ № 3 от «12» января 2026 г.

**Дополнительная общеобразовательная
(общеразвивающая) программа
«IT-квантум»**

Уровень программы: ознакомительный

Возраст обучающихся: 10–16 лет

Срок реализации: 2 месяца

Автор-составитель:

Кудашов Матвей Дмитриевич

педагог дополнительного образования

г.о.г. Выкса

Нижегородская область

2026

Содержание программы

Раздел 1. Комплекс основных характеристик программы

1.1	Пояснительная записка	3
1.2	Учебный план	11
1.3	Содержание учебного плана	12

Раздел 2. Комплекс организационно-педагогических условий

2.1	Календарный учебный график	14
2.2	Формы аттестации	15
2.3	Оценочные материалы	16
2.4	Методическое обеспечение программы	17
2.5	Условия реализации программы	20
2.6	Список литературы	21

Раздел 1. Комплекс основных характеристик программы

1.1 Пояснительная записка

Дополнительная общеобразовательная (общеразвивающая) программа «ИТ-квантум» имеет техническую направленность и ориентирована на формирование у обучающихся базовых представлений об алгоритмизации, программировании, цифровой среде и проектной деятельности в области информационных технологий с учетом возрастных особенностей обучающихся.

Программа рассчитана на обучающихся 10–16 лет и реализуется в объеме 16 академических часов. Уровень программы - стартовый. Содержание программы имеет единый характер и направлено на освоение начальных представлений об алгоритмизации, программировании и создании учебных цифровых проектов. В рамках программы используются визуальная среда Scratch, начальные элементы текстового программирования на Python, первичное знакомство со средой Unity и отдельные элементы C# на уровне демонстрации связи кода с поведением объекта.

Нормативно-правовая база

В основе программы лежат следующие нормативные документы:

- Федеральный закон Российской Федерации от 29.12.2012г. №273 «Об образовании в Российской Федерации» (ред. от 14.07.2022г.);
- Концепция развития дополнительного образования детей до 2030 года (распоряжение Правительства РФ от 31 марта 2022 г. № 678-р);
- Приказ Минпросвещения России от 27.07.2022г. № 629 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»;
- Письмо Минобрнауки России от 18.11.2015г. №09-3242 «О направлении информации» (вместе с «Методическими рекомендациями по проектированию дополнительных общеразвивающих программ (включая разноуровневые программы)»);
- Письмо Минобрнауки России от 29.03.2016г. №ВК-641/09 «О направлении методических рекомендаций» (вместе с «Методическими рекомендациями по реализации адаптированных дополнительных общеобразовательных программ, способствующих социально-психологической реабилитации, профессиональному самоопределению детей с ограниченными возможностями здоровья, включая детей-инвалидов, с учетом их особых образовательных потребностей»);
- 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»;
- СанПиН 1.2.3685-21 «Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания»;
- Методические рекомендации по разработке (составлению) дополнительной общеобразовательной общеразвивающей программы ГБОУ ДПО НИРО;
- Минимальные требования к оказанию государственной услуги в социальной сфере по реализации дополнительных общеразвивающих программ, утвержденные приказом Министерства образования и науки Нижегородской области от 08.08.2025г., №316-01-63-1685/25;

- Устав МАУ ДО «ЦМИиНК «Квант ОМК» от 17.12.2025 г № 5154;
- Режим занятий учащихся в детских объединениях МАУ ДО «ЦМИиНК «Квант ОМК» от 12.01.2026 г.

Актуальность программы

Актуальность программы определяется устойчивым ростом роли информационных технологий в учебной, профессиональной и повседневной деятельности человека. Освоение основ программирования развивает алгоритмическое и логическое мышление, формирует культуру работы с цифровыми инструментами, учит анализировать задачу, планировать последовательность действий, проверять результат и корректировать решение. Для обучающихся 10–16 лет такие умения являются значимой основой дальнейшего технического образования.

Адресат программы

Программа предназначена для обучающихся 10–16 лет, проявляющих интерес к программированию, компьютерным технологиям, созданию интерактивных проектов и цифровому творчеству. Содержание программы выстроено как единая стартовая траектория, доступная обучающимся без специальной предварительной подготовки. При реализации программы педагог учитывает возраст, темп работы, уровень самостоятельности и индивидуальные образовательные возможности обучающихся, регулируя сложность практических заданий и объем помощи на занятии.

Возрастные особенности обучающихся

Программа рассчитана на обучающихся 10–16 лет и реализуется по единому учебному плану. Все обучающиеся последовательно проходят предусмотренные программой темы: алгоритмизацию в Scratch, начальные конструкции Python, первичное знакомство со средой Unity и элементами C#. Учет возрастных особенностей осуществляется через дифференциацию учебных заданий, объема педагогической помощи, степени самостоятельности и глубины объяснения изучаемого материала.

Для обучающихся 10–11 лет характерна потребность в наглядности, практическом действии, четкой последовательности учебных шагов и видимом результате работы. При освоении всех тем программы педагог использует подробный показ действий, пошаговые инструкции, опору на визуальные примеры, частую проверку промежуточного результата и дополнительные пояснения. Акцент делается на понимании команды, события, движения, условия, цикла и переменной через действия объектов на сцене.

Для обучающихся 12–13 лет возрастает интерес к самостоятельному изменению проекта, поиску ошибки, сравнению вариантов решения и объяснению причин полученного результата. При изучении всех тем программы педагог постепенно увеличивает долю самостоятельной работы: обучающиеся изменяют параметры проекта, проверяют разные варианты выполнения программы, находят и исправляют ошибки, объясняют назначение отдельных команд, блоков или строк кода. Задания строятся так, чтобы обучающийся не только повторял показанное действие, но и осознанно вносил изменения в учебный проект.

Для обучающихся 14–16 лет более значимыми становятся осознанность учебной деятельности, понимание практического смысла задания, развитие самоконтроля и интерес к современным цифровым инструментам. При этом возраст обучающегося не

является единственным показателем уровня подготовки: в одной группе могут обучаться дети с разным опытом, разной скоростью освоения материала и разной степенью самостоятельности. Поэтому при работе со старшими подростками сохраняются доступная структура объяснения, демонстрация, практическая направленность занятий и возможность выполнения задания на базовом уровне. Для обучающихся, готовых к усложнению, используются дополнительные элементы: самостоятельный подбор параметров, изменение условий задачи, более подробное объяснение логики работы программы, поиск и исправление ошибок, представление результата группе.

В разновозрастной группе содержание программы остается единым, а различия в организации обучения проявляются в уровне сложности практических заданий. Обучающиеся с меньшим опытом выполняют базовый вариант задания с опорой на инструкцию и помощь педагога. Обучающиеся с более высоким уровнем подготовки выполняют расширенный вариант: добавляют дополнительные условия, изменяют параметры, объясняют программную логику, проводят самопроверку и представляют результат группе.

Такое построение программы обеспечивает соблюдение единого учебного плана и одновременно позволяет учитывать возраст, темп работы, уровень подготовки и степень самостоятельности обучающихся.

Новизна и отличительные особенности программы

Отличительной особенностью программы является единое стартовое построение содержания: от наглядного освоения алгоритмических действий в Scratch к начальному знакомству с текстовым программированием на Python и первичному представлению об объектно-компонентной модели Unity с элементами C#. Учебный материал представлен через практические задания, в которых понятия программирования связываются с наблюдаемым результатом: движением объекта, обработкой события, изменением значения переменной, выполнением условия, повторением действия или взаимодействием объектов на сцене.

Объем и срок освоения программы

Объем программы - 16 академических часов. Срок освоения - 2 месяца. Режим занятий: 1 раз в неделю по 2 академических часа. Продолжительность академического часа - 45 минут. Между занятиями предусматривается перерыв в соответствии с санитарно-гигиеническими требованиями.

Форма обучения и особенности организации образовательного процесса

Форма обучения - очная. Основная форма организации деятельности - групповое занятие с индивидуальным сопровождением обучающихся. На занятиях используются объяснение нового материала, демонстрация, практическая работа за компьютером, обсуждение результата, выполнение контрольных заданий, самооценка и защита учебного проекта.

Цель программы

Формирование у обучающихся начальных компетенций в области алгоритмизации и программирования через выполнение учебных цифровых проектов в визуальной и текстовой программных средах.

Задачи программы

Образовательные задачи:

- сформировать представление об алгоритме, программе, исполнителе, команде и последовательности действий;
- познакомить обучающихся с базовыми алгоритмическими конструкциями: линейным алгоритмом, условием, циклом и событием;
- сформировать умение использовать переменные, выражения, ввод и вывод данных в учебных программах;
- познакомить обучающихся с понятиями объекта, компонента, сцены, скрипта, поля и метода в Unity и C# на стартовом уровне;
- сформировать навык чтения и объяснения программной логики на уровне изучаемых тем;
- обеспечить выполнение учебного проекта с применением изученных алгоритмических конструкций и цифровых инструментов.

Развивающие задачи:

- развивать логическое, алгоритмическое и пространственное мышление;
- развивать способность анализировать условие задачи и выбирать последовательность действий;
- формировать умение сопоставлять ожидаемый и полученный результат;
- развивать внимание, самостоятельность, учебную дисциплину и навыки самопроверки;
- формировать способность находить причину ошибки и корректировать решение.

Воспитательные задачи:

- воспитывать ответственное отношение к работе с компьютерной техникой и учебными файлами;
- формировать культуру безопасной и аккуратной работы в цифровой среде;
- развивать уважительное отношение к результатам труда других обучающихся;
- формировать навыки сотрудничества, взаимопомощи и корректного обсуждения учебного результата.

Планируемые результаты**По итогам освоения программы обучающиеся будут знать:**

- назначение алгоритма, программы, команды и исполнителя;
- принципы работы линейных алгоритмов, условий, циклов и событий;
- назначение переменных, выражений, ввода и вывода данных;
- назначение объекта, сцены, компонента, скрипта, поля и метода на стартовом уровне;
- основные правила безопасной работы в компьютерном классе.

Обучающиеся будут уметь:

- создавать и сохранять учебный цифровой проект в используемой программной среде;
- составлять последовательность команд для решения учебной задачи;
- использовать переменные, условия и циклы при решении учебных задач;
- проверять работу программы и объяснять полученный результат;
- представлять учебный проект и отвечать на вопросы по его логике.

У обучающихся будут сформированы:

- интерес к программированию, инженерному и техническому творчеству;
- первичный опыт проектной деятельности в области информационных технологий;

- навыки самоконтроля, аккуратной работы и последовательного выполнения учебной задачи.

Воспитательный компонент программы

Воспитательный потенциал программы

Воспитательный потенциал программы направлен на формирование у обучающихся культуры безопасной, ответственной и осмысленной работы в цифровой среде, развитие трудолюбия, эстетического вкуса, навыков командного взаимодействия и уважительного отношения к результатам интеллектуального труда.

Цель воспитательной работы

Формирование социально ответственной, коммуникативной и целеустремленной личности, готовой к командному взаимодействию, саморазвитию, соблюдению норм цифровой культуры и осознанному выбору профессиональной траектории в сфере информационных технологий.

Задачи воспитательной работы

1. В области нравственного и гражданского воспитания:

- формировать культуру цифрового общения, соблюдения сетевого этикета и уважительного взаимодействия в учебной цифровой среде;
- воспитывать уважение к чужому интеллектуальному труду, понимание значения авторского права и недопустимости присвоения результатов работы других людей;
- воспитывать уважение к труду, дисциплину, ответственность за качество выполненной работы и соблюдение сроков выполнения учебных заданий;
- формировать гражданскую позицию и чувство сопричастности к жизни родного города через выполнение учебных задач, связанных с современными цифровыми технологиями и техническим развитием региона.

2. В области социального развития:

- развивать навыки эффективной командной работы, распределения ролей, согласования решений и взаимопомощи при выполнении практических заданий;
- формировать умение аргументированно высказывать собственную позицию, корректно принимать замечания и конструктивно разрешать разногласия;
- развивать навыки публичного представления учебного проекта, уверенного объяснения программной логики и ответа на вопросы аудитории.

3. В области личностного роста:

- развивать самостоятельность, инициативность, настойчивость в достижении цели и навыки самоорганизации при работе над цифровым продуктом;
- способствовать развитию креативности, критического мышления, внимательности к деталям и способности анализировать полученный результат;
- формировать ответственное отношение к собственному здоровью при работе за компьютером, соблюдение режима зрительной нагрузки, правил посадки и организации рабочего места.

Ожидаемые результаты воспитательной работы

В результате реализации программы у обучающихся будут сформированы следующие качества и навыки:

1. В области гражданской позиции, нравственности и цифровой культуры:

- сформирована культура цифрового общения, понимание норм сетевого этикета и уважительного взаимодействия в образовательной среде;
- сформировано понимание значения авторского права, уважение к интеллектуальному труду и ответственное отношение к использованию цифровых материалов;

- воспитаны трудовая дисциплина, аккуратность, ответственность за качество выполнения задания и бережное отношение к компьютерной технике и учебным файлам;
- сформировано чувство сопричастности к жизни родного города и уважение к техническому и промышленному развитию региона.

2. В области социального развития:

- развиты навыки командной работы: обучающиеся умеют распределять роли, договариваться, оказывать помощь и принимать совместные решения;
- сформированы навыки публичного выступления, демонстрации результата и объяснения собственной проектной идеи;
- сформировано умение конструктивно обсуждать ошибки, принимать обратную связь и корректировать результат деятельности.

3. В области личностного роста:

- развиты самостоятельность, инициативность, настойчивость в достижении учебной цели и навыки самоорганизации;
- развиты креативность при решении нестандартных задач и критическое мышление при анализе информации и программного результата;
- сформировано ответственное отношение к собственному здоровью при работе в цифровой среде.

Формы проведения воспитательных мероприятий

- групповая работа: выполнение практических заданий, кейсов и учебных проектов в малых группах;
- проектная деятельность: разработка учебных цифровых продуктов, интерактивных проектов, программ и сцен с последующей демонстрацией результата;
- публичная защита проектов: мини-конференции, презентации работ перед группой, педагогом.
- дискуссии: обсуждение тем «Этика в интернете», «Авторское право в цифровой среде», «Фейковые новости», «Профессии будущего», «Ответственное использование цифровых инструментов»;
- рефлексия: обсуждение процесса командного взаимодействия, трудностей, способов их преодоления и личного вклада обучающегося в общий результат.

Методы воспитательного воздействия

- метод проектов: создание ситуации личной ответственности за общий результат команды и качество конечного цифрового продукта;
- создание ситуации успеха: педагогическая поддержка, дозированная помощь и поэтапное сопровождение, позволяющие обучающемуся поверить в собственные силы;
- рефлексия: обсуждение не только технических аспектов выполненной работы, но и процесса взаимодействия в команде, возникших трудностей и способов их преодоления;
- метод примера: анализ успешных кейсов в сфере информационных технологий, цифровых продуктов и образовательных проектов; знакомство с профессиональными траекториями разработчиков, инженеров и специалистов цифровой индустрии.

Концептуальный подход к реализации воспитательного компонента

Концептуальным подходом к построению программы являются принципы сознательности и активности, доступности, последовательности, наглядности, связи теории с практикой, уважения к личности обучающегося и ответственности за результат совместной деятельности.

С учетом цели и задач содержание программы реализуется поэтапно с постепенным усложнением учебных действий. На вводном этапе у обучающихся формируются начальные представления о безопасной работе в цифровой среде, алгоритме, программе и учебной дисциплине. На основном этапе проводится освоение базовых понятий программирования, закрепление навыков практической работы, развитие самостоятельности и командного взаимодействия. На завершающем этапе обучающиеся представляют учебный мини-проект, объясняют его логику, оценивают собственный вклад и получают опыт публичной защиты результата.

Таким образом, воспитательная работа в рамках программы осуществляется через практическую творческую деятельность, последовательное выполнение учебных задач, командное взаимодействие, обсуждение результата и формирование ответственного отношения к цифровой деятельности.

1.2 Учебный план

№ п/п	Название раздела, темы	Всего	Теория	Практика	Формы аттестации / контроля
1	Вводное занятие. Алгоритм и среда Scratch.	2	1	1	Опрос. Входное тестирование. Наблюдение.
2	Scratch: движение и события.	2	0,5	1,5	Опрос. Практическая работа.
3	Scratch: циклы, условия и переменные.	2	0,5	1,5	Опрос. Практическая работа.
4	Scratch: счет и взаимодействие объектов.	2	1	1	Практическая работа. Тестирование.
5	Python: переменные, ввод и вывод данных.	2	0,5	1,5	Практическая работа. Педагогическое наблюдение.
6	Python: условия, циклы и списки.	2	1	1	Опрос. Практическая работа.
7	Unity и C#: сцена, объект и скрипт.	2	0,5	1,5	Проверка работы. Самооценка.
8	Защита учебного мини-проекта.	2	1	1	Защита проекта. Беседа.
	Всего	16	6	10	

1.3 Содержание учебного плана

Тема 1. Вводное занятие. Алгоритм и среда Scratch.

Теория: Инструктаж по технике безопасности. Правила поведения в компьютерном классе. Понятие алгоритма, программы, команды и исполнителя. Назначение среды Scratch. Сцена, спрайт, блоки команд, область скриптов.

Практика: Входное тестирование. Создание первого учебного проекта в Scratch. Добавление спрайта, настройка простого действия, запуск и сохранение результата.

Тема 2. Scratch: движение и события.

Теория: Линейный алгоритм. Последовательное выполнение команд. Координаты, направление, движение объекта на сцене. Событие как условие запуска программы.

Практика: Составление последовательности команд движения. Настройка запуска по флажку и нажатию клавиши. Проверка поведения спрайта на сцене.

Тема 3. Scratch: циклы, условия и переменные.

Теория: Циклы «повторить» и «всегда». Условная конструкция «если». Сенсоры касания цвета, края сцены и другого спрайта. Переменная как средство хранения значения.

Практика: Организация повторяющихся действий. Настройка реакции объекта на условие. Создание переменной, отображение значения на сцене и проверка изменения значения.

Тема 4. Scratch: счет и взаимодействие объектов.

Теория: Система счета. Начисление и уменьшение баллов. Сообщения между спрайтами. Связь событий, условий, переменных и правил взаимодействия объектов.

Практика: Настройка правил взаимодействия объектов. Добавление счета и сообщений. Выполнение промежуточного тестирования. Проверка логики работы проекта.

Тема 5. Python: переменные, ввод и вывод данных.

Теория: Назначение Python как языка текстового программирования. Переменная, значение, присваивание. Типы учебных значений: число и строка. Команды print() и input(). Простые выражения.

Практика: Написание программы с вводом и выводом данных. Изменение значений переменных. Проверка результата выполнения программы на разных входных данных.

Тема 6. Python: условия, циклы и списки.

Теория: Условный оператор if. Сравнение значений. Цикл как форма повторения действия. Счетчик, условие продолжения и завершение повторения. Список как упорядоченный набор значений.

Практика: Создание программы с условием и повторением действий. Выполнение операций со списком на стартовом уровне: добавление, вывод, подсчет количества элементов.

Тема 7. Unity и C#: сцена, объект и скрипт.

Теория: Интерфейс Unity. Сцена, GameObject, Transform, компонент. Скрипт как компонент объекта. Первичное представление о C#-скрипте, поле, методах Start и Update, публичном параметре в Inspector.

Практика: Работа со сценой Unity. Изменение положения, масштаба и визуальных свойств объекта. Подключение учебного скрипта движения, изменение параметра и проверка поведения объекта в режиме запуска.

Тема 8. Защита учебного мини-проекта.

Теория: Повторение понятий: алгоритм, программа, команда, событие, условие, цикл, переменная, объект, компонент, скрипт, параметр. Критерии защиты учебного мини-проекта.

Практика: Демонстрация учебного мини-проекта, объяснение использованных команд, блоков или строк кода, ответы на вопросы. Промежуточная аттестация.

Раздел 2. Комплекс организационно-педагогических условий

Раздел 2. Комплекс организационно-педагогических условий

2.1 Календарный учебный график

Календарный учебный график представлен для реализации программы в течение 2 месяцев. Конкретные даты занятий уточняются образовательной организацией с учетом расписания объединения, праздничных дней и локальных актов.

№ занятия	Дата	Тема занятия	Кол-во часов	Форма занятия	Форма контроля
1	Апрель	Вводное занятие. Алгоритм и среда Scratch.	2	Комбинированная. Беседа. Практическая работа.	Опрос. Входное тестирование. Наблюдение.
2	Апрель	Scratch: движение и события.	2	Комбинированная. Демонстрация. Практическая работа.	Опрос. Практическая работа.
3	Апрель	Scratch: циклы, условия и переменные.	2	Комбинированная. Учебный практикум. Работа в парах.	Опрос. Практическая работа.
4	Апрель	Scratch: счет и взаимодействие объектов.	2	Комбинированная. Беседа. Практическая работа.	Практическая работа. Тестирование.
5	Май	Python: переменные, ввод и вывод данных.	2	Комбинированная. Демонстрация. Индивидуальная работа.	Практическая работа. Педагогическое наблюдение.
6	Май	Python: условия, циклы и списки.	2	Комбинированная. Учебный практикум. Практическая работа.	Опрос. Практическая работа.
7	Май	Unity и C#: сцена, объект и скрипт.	2	Комбинированная. Демонстрация. Практическая работа.	Проверка работы. Самооценка.
8	Май	Защита учебного мини-проекта.	2	Комбинированная. Защита проекта. Беседа.	Защита проекта. Беседа.

2.2 Формы аттестации

Для отслеживания результатов освоения программы используются входной контроль, текущий контроль, промежуточное тестирование, педагогическое наблюдение, самооценка и промежуточная аттестация.

Текущий контроль проводится на каждом занятии в форме устного опроса, наблюдения за работой обучающихся, проверки выполнения практического задания и обсуждения результата.

Промежуточная аттестация проводится в форме защиты учебного мини-проекта. Обучающийся демонстрирует результат, объясняет использованные программные конструкции, отвечает на вопросы педагога и проводит самооценку выполненной работы.

Формы контроля:

- входное тестирование;
- устный опрос;
- практическая работа;
- промежуточное тестирование;
- защита учебного мини-проекта;
- самооценка обучающегося;
- педагогическое наблюдение.

2.3 Оценочные материалы

Оценка результатов освоения программы проводится на основе индивидуальной динамики обучающегося, качества выполнения практических заданий, понимания изученных понятий и способности объяснить логику созданного цифрового продукта.

Уровень освоения	Характеристика результата
Высокий	Обучающийся самостоятельно выполняет учебную задачу, применяет изученные конструкции, проверяет результат, объясняет назначение основных блоков или строк кода, исправляет ошибки с минимальной помощью.
Средний	Обучающийся выполняет учебную задачу с отдельными подсказками, понимает общий смысл программы, допускает ошибки и исправляет их после разъяснения педагога.
Базовый	Обучающийся выполняет часть действий при сопровождении педагога, затрудняется в самостоятельном объяснении отдельных команд, нуждается в повторном показе и дополнительной тренировке.

Примерные критерии защиты учебного мини-проекта:

- проект открывается и запускается;
- в проекте реализованы действия, соответствующие заданию;
- обучающийся показывает основную программную логику;
- обучающийся объясняет не менее двух изученных команд, блоков, параметров или строк кода;
- соблюдены требования безопасности и аккуратного хранения файлов.

2.4 Методическое обеспечение программы

Методическое обеспечение программы направлено на последовательное формирование у обучающихся алгоритмического мышления, навыков практической работы в цифровой среде, культуры самостоятельной проверки результата и опыта представления собственного учебного продукта. Занятия строятся как сочетание объяснения, демонстрации, практической работы, анализа ошибок, самопроверки и мини-защиты результата.

Формы проведения занятий: комбинированное занятие, беседа, объяснение, демонстрация, практическая работа за компьютером, учебный практикум, мини-проект, защита учебного результата, разбор ошибок, индивидуальная консультация, групповое обсуждение.

Виды организации деятельности обучающихся: фронтальная работа, индивидуальная работа за компьютером, работа в парах, работа в малых группах, коллективное обсуждение решений, взаимопроверка, самооценка по критериям.

Методы организации учебно-воспитательного процесса:

- словесные методы: объяснение, беседа, инструктаж, постановка учебной задачи, обсуждение результата;
- наглядные методы: демонстрация интерфейса среды, показ алгоритма, разбор схемы программы, визуализация выполнения команд;
- практические методы: выполнение заданий за компьютером, настройка объектов, запуск и проверка проекта, исправление ошибок;
- репродуктивный метод: выполнение учебных действий по образцу с последующим объяснением результата;
- частично-поисковый метод: самостоятельный выбор команды, параметра, условия или способа исправления ошибки в пределах изученной темы;
- проектный метод: создание учебного цифрового продукта с последующей демонстрацией и защитой;
- метод проблемного обучения: анализ ситуации, в которой программа работает не так, как ожидалось, поиск причины и корректировка решения;
- метод рефлексии: обсуждение трудностей, качества командного взаимодействия, аккуратности файловой работы и способов улучшения результата.

Логика занятия включает постановку задачи, актуализацию изученных понятий, показ педагогом ключевого приема, выполнение обучающимися практической работы, проверку результата, исправление ошибок, фиксацию вывода и краткую самооценку. При необходимости педагог использует дозированную помощь: вопрос-подсказку, повторную демонстрацию, карточку с алгоритмом действий, индивидуальное объяснение.

Средства обучения и дидактические материалы:

- инструкционные карты и чек-листы выполнения практических работ;
- учебные Scratch-проекты и карточки с блоками команд;
- учебные программы на Python для демонстрации переменных, условий, циклов и списков;
- учебные Unity-сцены и C#-скрипты для демонстрации объектов, компонентов, полей, методов и событий;
- презентации, схемы алгоритмов, демонстрационные примеры, материалы для входного и промежуточного тестирования;
- критерии проверки учебного мини-проекта и бланки самооценки обучающегося.

Особенности работы с Python, C# и Unity

Работа с программными средами организуется в рамках единого стартового курса и строится по принципу постепенного перехода от визуального программирования к начальному знакомству с текстовым кодом и объектно-компонентной средой.

Python используется для освоения текстового программирования и базовых алгоритмических конструкций. C# и Unity применяются для изучения связи программного кода с поведением объекта в сцене, понимания роли компонентов, параметров, методов и событий взаимодействия. Практическая работа строится от наблюдаемого результата к объяснению программной логики: обучающийся запускает программу, изменяет параметр, проверяет поведение объекта и формулирует вывод.

Описание занятий, методик и средств обучения

№ п/п	Наименование разделов	Формы занятий	Приемы и методы организации учебно-воспитательного процесса	Дидактический материал	Техническое оснащение занятий	Форма подведения итогов
1	Вводное занятие. Алгоритм и среда Scratch.	Комбинированное занятие, беседа, инструктаж, практическая работа	Словесный, наглядный, практический методы; педагогическое наблюдение; входная диагностика	Презентация о программе, инструкция по технике безопасности, бланки входного тестирования, памятка по работе с файлами	ПК педагога, проектор или интерактивная панель, компьютеры обучающихся, браузер, Scratch	Опрос. Входное тестирование. Наблюдение.
2	Scratch: движение и события.	Комбинированное занятие, демонстрация, практическая работа за компьютером	Объяснение, показ образца, репродуктивное выполнение, проверка результата, взаимопроверка, разбор ошибок	Карточки с последовательностями команд, схемы алгоритмов, задания на движение объекта и запуск по событию	Компьютеры обучающихся, демонстрационный экран, Scratch	Опрос. Практическая работа.
3	Scratch: циклы, условия и переменные.	Практическое занятие, учебный практикум, работа в парах	Наглядный и практический методы; частично-поисковый метод; разбор ошибок; самопроверка	Карточки с условиями, схемы ветвления и повторения, задания на изменение переменных	Компьютеры обучающихся, демонстрационный экран, Scratch	Опрос. Практическая работа.
4	Scratch: счет и взаимодействие объектов.	Комбинированное занятие, беседа, индивидуальная и парная практическая работа	Объяснение, демонстрация, практическое закрепление, пооперационный контроль, обсуждение правил взаимодействия	Таблицы значений переменных, карточки с сообщениями между спрайтами, задания на счет	Компьютеры обучающихся, Scratch, проектор или демонстрационный экран	Практическая работа. Тестирование.
5	Python: переменные, ввод и вывод данных.	Комбинированное занятие, демонстрация, индивидуальная работа	Объяснение, показ кода, выполнение по образцу, изменение данных, проверка результата	Примеры программ, карточки с понятиями переменная, значение, ввод, вывод	Компьютеры обучающихся, Python, демонстрационный экран	Практическая работа. Педагогическое наблюдение.

6	Python: условия, циклы и списки.	Комбинированное занятие, учебный практикум, практическая работа	Чтение фрагмента программы, изменение условий, организация повторения, анализ ошибок	Примеры программ с условием, циклом и списком; задания на сравнение и вывод данных	Компьютеры обучающихся, Python, проектор или интерактивная панель	Опрос. Практическая работа.
7	Unity и C#: сцена, объект и скрипт.	Комбинированное занятие, демонстрация, практическая работа	Наглядный и практический методы; пошаговая инструкция; изменение параметров; запуск и наблюдение результата	Учебная Unity-сцена, схема интерфейса, карточки с понятиями объект, сцена, компонент, скрипт	Компьютеры обучающихся, Unity Hub, Unity Editor, Visual Studio Code или Visual Studio	Проверка работы. Самооценка.
8	Защита учебного мини-проекта.	Защита проекта, беседа, демонстрация результата, групповое обсуждение	Проектный метод, рефлексия, публичное выступление, педагогический контроль, самооценка	Критерии защиты, бланк самооценки, чек-лист проверки проекта, материалы аттестации	Компьютеры обучающихся, демонстрационный экран, используемые программные среды	Защита учебного мини-проекта. Беседа.

2.5 Условия реализации программы

Материально-техническое обеспечение:

- учебный кабинет, соответствующий санитарно-эпидемиологическим требованиям и правилам охраны труда;
- рабочее место педагога с возможностью демонстрации учебных материалов;
- персональные компьютеры или ноутбуки для обучающихся;
- проектор, интерактивная панель или демонстрационный экран;
- компьютерная мышь, клавиатура, наушники при необходимости;
- доступ к сети Интернет для работы с образовательными ресурсами и документацией.

Программное обеспечение:

- операционная система Windows или Linux;
- современный веб-браузер;
- Scratch или доступ к онлайн-версии Scratch;
- Python 3.x и среда для запуска учебных программ;
- Unity Hub и установленная версия Unity для учебной работы;
- Visual Studio Code, Visual Studio или аналогичная среда для работы с C#;
- офисный пакет или текстовый редактор для подготовки материалов.

Кадровые условия:

Реализацию программы осуществляет педагог дополнительного образования, владеющий основами программирования, методикой проведения практических занятий, навыками проектной организации учебной деятельности и требованиями безопасной работы обучающихся в компьютерном классе.

2.6 Список литературы

Нормативно-правовые документы

1. Концепция развития дополнительного образования детей до 2030 года, утвержденная распоряжением Правительства Российской Федерации от 31.03.2022 г. № 678-р.
2. Методические рекомендации по разработке (составлению) дополнительной общеобразовательной общеразвивающей программы ГБОУ ДПО НИРО.
3. Минимальные требования к оказанию государственной услуги в социальной сфере по реализации дополнительных общеразвивающих программ, утвержденные приказом Министерства образования и науки Нижегородской области от 08.08.2025 г. № 316-01-63-1685/25.
4. Письмо Министерства образования и науки Российской Федерации от 18.11.2015 г. № 09-3242 «О направлении информации» вместе с «Методическими рекомендациями по проектированию дополнительных общеразвивающих программ, включая разноуровневые программы».
5. Письмо Министерства образования и науки Российской Федерации от 29.03.2016 г. № ВК-641/09 «О направлении методических рекомендаций» вместе с «Методическими рекомендациями по реализации адаптированных дополнительных общеобразовательных программ, способствующих социально-психологической реабилитации, профессиональному самоопределению детей с ограниченными возможностями здоровья, включая детей-инвалидов, с учетом их особых образовательных потребностей».
6. Постановление Главного государственного санитарного врача Российской Федерации от 28.09.2020 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”».
7. Приказ Министерства просвещения Российской Федерации от 27.07.2022 г. № 629 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам».
8. Режим занятий учащихся в детских объединениях МАУ ДО «ЦМИиНК “Квант ОМК”» от 12.01.2026 г.
9. СанПиН 1.2.3685-21 «Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания».
10. Устав МАУ ДО «ЦМИиНК “Квант ОМК”» от 17.12.2025 г. № 5154.
11. Федеральный закон Российской Федерации от 29.12.2012 г. № 273-ФЗ «Об образовании в Российской Федерации» (ред. от 14.07.2022 г.).

Список литературы, используемой педагогом для работы

1. Албахари Дж. С# 10.0. Справочник. СПб.: Диалектика.
2. Беспалько В. П. Основы теории педагогических систем. Воронеж: Издательство Воронежского университета.
3. Бонд Дж. Unity и С#. Геймдев от идеи до реализации. СПб.: Питер.
4. Босова Л. Л., Босова А. Ю. Информатика. 5–6 классы: учебник. М.: БИНОМ. Лаборатория знаний.
5. Босова Л. Л., Босова А. Ю. Информатика. 7–9 классы: учебник. М.: БИНОМ. Лаборатория знаний.
6. Доусон М. Програмуємо на Python. СПб.: Питер.

7. Златопольский Д. М. Основы программирования на языке Python. М.: ДМК Пресс.
8. Златопольский Д. М. Сборник задач по программированию. М.: ДМК Пресс.
9. Лутц М. Изучаем Python. СПб.: Символ-Плюс.
10. Мартин Р. Чистый код. Создание, анализ и рефакторинг. СПб.: Питер.
11. Полат Е. С. Метод проектов. М.: Академия.
12. Саммерфилд М. Программирование на Python 3. Подробное руководство. СПб.: Символ-Плюс.
13. Селевко Г. К. Современные образовательные технологии. М.: Народное образование.
14. Семакин И. Г., Хеннер Е. К. Информатика. Базовый курс. М.: БИНОМ. Лаборатория знаний.
15. Троелсен Э., Джепикс Ф. Язык программирования C# и платформа .NET. СПб.: Вильямс.
16. Угринович Н. Д. Информатика и информационные технологии. М.: БИНОМ. Лаборатория знаний.
17. Хокинг Дж. Unity в действии. Мультиплатформенная разработка на C#. СПб.: Питер.
18. Шелл Дж. Геймдизайн. Как создать игру, в которую будут играть все. М.: Альпина Паблишер.
19. Шилдт Г. C#: полное руководство. М.: Вильямс.
20. Microsoft Learn: документация по языку C#. Режим доступа: <https://learn.microsoft.com/ru-ru/dotnet/csharp/>
21. Microsoft Learn: .NET documentation. Режим доступа: <https://learn.microsoft.com/dotnet/>
22. PEP 8. Style Guide for Python Code. Режим доступа: <https://peps.python.org/pep-0008/>
23. Unity Learn. Режим доступа: <https://learn.unity.com/>
24. Unity Manual. Режим доступа: <https://docs.unity3d.com/Manual/>

Список литературы и электронных ресурсов для учащихся

1. Бриггс Дж. Python для детей. Самоучитель по программированию. М.: Манн, Иванов и Фербер.
2. Голиков Д. В. Scratch 3 для юных программистов. СПб.: БХВ-Петербург.
3. Маккарти У. Программирование игр на Scratch. М.: ДМК Пресс.
4. Маттес Э. Изучаем Python. Программирование игр, визуализация данных, веб-приложения. СПб.: Питер.
5. Пейн Б. Python для детей и родителей. М.: Манн, Иванов и Фербер.
6. Свейгарт Э. Автоматизация рутинных задач с помощью Python. СПб.: Питер.
7. Свейгарт Э. Программирование для детей. Делай игры и учи язык Scratch! М.: Эксмо.
8. Официальный сайт Scratch. Режим доступа: <https://scratch.mit.edu/>
9. Учебные материалы Python для начинающих. Режим доступа: <https://www.python.org/about/gettingstarted/>
10. Code.org: основы программирования. Режим доступа: <https://code.org/>
11. Khan Academy: Computer Programming. Режим доступа: <https://www.khanacademy.org/computing/computer-programming>
12. Microsoft Learn: введение в программирование на C#. Режим доступа: <https://learn.microsoft.com/ru-ru/dotnet/csharp/>
13. Python Tutor: визуализация выполнения программ. Режим доступа: <https://pythontutor.com/>
14. Scratch Ideas. Режим доступа: <https://scratch.mit.edu/ideas>

15. Unity Learn: учебные материалы по созданию игр. Режим доступа:
<https://learn.unity.com/>