

РОССИЙСКАЯ ФЕДЕРАЦИЯ
АДМИНИСТРАЦИЯ ГОРОДСКОГО ОКРУГА «ГОРОД КАЛИНИНГРАД»
КОМИТЕТ ПО ОБРАЗОВАНИЮ
муниципальное автономное учреждение дополнительного образования
города Калининграда Станция юных техников

Принята на заседании
педагогического совета МАУДО СЮТ
протокол № ____ от « ____ » _____ 2026 г.

УТВЕРЖДАЮ
Директор МАУДО СЮТ
_____/Стецюк О.Е./
« ____ » _____ 2026 г.

**Дополнительная общеобразовательная общеразвивающая программа
технической направленности
«Юный программист»**

Возраст обучающихся: 8-14 лет

Срок реализации: 9 месяцев

Автор программы:
Трущелев Александр Николаевич,
педагог дополнительного образования
г.Калининград

Калининград, 2026 г.

Пояснительная записка

Описание предмета, дисциплины которому посвящена программа

Предмет, дисциплина, которой посвящена программа "Юный программист", является введением в программирование с использованием визуального языка Scratch. Scratch - это интерактивная среда разработки, специально созданная для обучения детей основам программирования и развития вычислительного мышления.

Программа "Юный программист" предназначена для детей, которые хотят научиться программировать и разрабатывать собственные интерактивные проекты, игры и анимации. Целью программы является ознакомление детей с основами программирования, развитие творческого мышления и пространственной логики, а также стимулирование интереса к технологиям и информатике.

В течение программы учащиеся изучают основные концепции программирования, такие как алгоритмы, условия, циклы и переменные. Они изучают основные блоки, используемые в Scratch, и учатся комбинировать эти блоки для создания разнообразных проектов.

Программа включает в себя практические занятия, где учащиеся активно создают свои собственные работы на Scratch. Они могут создавать интерактивные игры, анимации, истории и мультимедийные проекты, воплощая свои идеи и фантазии. Учащиеся также имеют возможность работать в команде, обмениваться идеями и улучшать свои проекты на основе обратной связи от преподавателя и своих коллег.

Предмет "Юный программист" помогает развить навыки решения проблем, логического мышления, творческого подхода к задачам, а также способность к сотрудничеству и командной работе. Это также является первым шагом в освоении программирования, который может послужить основой для изучения других языков программирования в будущем.

Программное обеспечение полностью бесплатно. Можно работать прямо на сайте, или скачать и установить на компьютере.

Раскрытие ведущих идей, на которых базируется программа

Программа "Юный программист" базируется на нескольких ведущих идеях, которые являются основой для обучения детей программированию на языке Scratch. Вот некоторые из этих идей:

1. Развитие вычислительного мышления: Программа направлена на развитие у детей навыков и способностей, связанных с вычислительным мышлением. Она помогает учащимся учиться анализировать задачи, разбивать их на более простые компоненты, решать проблемы логически и систематически, а также формулировать и реализовывать алгоритмы для достижения желаемого результата.

2. Визуальное программирование: Программа использует визуальный язык программирования Scratch, который позволяет детям создавать

программы, соединяя блоки с помощью графического интерфейса. Это позволяет учащимся сосредоточиться на логике и структуре программы, не беспокоясь о синтаксических правилах, свойственных текстовым языкам программирования. Визуальное программирование делает процесс программирования доступным и понятным для детей.

3. Творческий подход и самовыражение: Программа поощряет творческое мышление и самовыражение учащихся. Дети имеют возможность создавать свои собственные проекты, игры и анимации, основываясь на своих интересах и идеях. Они могут развивать свою фантазию, экспериментировать с различными концепциями и дизайном, и превращать свои идеи в реальность через программирование.

4. Коллаборативное обучение: Программа стимулирует сотрудничество и командную работу. Учащиеся имеют возможность работать в группах, обмениваться идеями, помогать друг другу и учиться от опыта своих коллег. Коллективное обучение способствует развитию коммуникативных и социальных навыков, а также умению работать в команде.

5. Проектная деятельность: Программа основана на принципе проектной деятельности, где учащиеся активно создают свои собственные проекты на Scratch.

Они учатся применять свои знания и навыки программирования для реализации конкретных идей и целей. Проектная деятельность способствует практическому применению знаний, развитию проблемно-ориентированного мышления и самостоятельности учащихся.

Все эти идеи объединены в рамках программы "Юный программист", чтобы обеспечить эффективное и интересное обучение программированию для детей.

Описание ключевых понятий

Программа "Юный программист" включает в себя несколько ключевых понятий, которые являются основой для обучения детей программированию на языке Scratch. Вот некоторые из этих ключевых понятий:

1. **Scratch:** Это визуальный язык программирования, позволяющий детям создавать собственные интерактивные истории, игры и анимацию. Особенность Scratch в том, что это визуальный язык программирования. Это означает, что дети учатся программированию с помощью разноцветных и интересных блоков, а не сложных программных кодов. Это делает Scratch особенно интуитивным в изучении, простым и веселым языком программирования.
2. **Язык программирования:** Формальный язык, предназначенный для записи компьютерных программ.
3. **Программирование:** Это процесс создания программ по алгоритму для управления компьютером или роботом. Для этого программисты пишут исходный код на одном из языков программирования.
4. **Алгоритм:** Алгоритм представляет собой последовательность шагов или инструкций, необходимых для решения определенной задачи. Учащиеся изучают, как разрабатывать алгоритмы для выполнения

конкретных действий или достижения желаемого результата.

5. **Инструкция:** Это команда или набор команд на языке программирования.
6. **Блоки программирования:** В языке Scratch программирование осуществляется путем соединения различных блоков. Блоки представляют собой графические элементы, которые содержат определенные команды или операции. Учащиеся изучают различные типы блоков, такие как блоки движения, блоки управления, блоки внешнего вида и другие, и учатся комбинировать их для создания программ.
7. **Спрайты:** Спрайты представляют собой графические объекты или персонажей, которые участвуют в проектах Scratch. Учащиеся учатся создавать и управлять спрайтами, добавлять им анимацию и взаимодействие с другими спрайтами или пользователем.
8. **Сцены:** Сцены представляют различные фоны или места, на которых происходят действия в проекте Scratch. Учащиеся учатся создавать разные сцены и переключаться между ними в зависимости от хода программы.
9. **Условные операторы:** Условные операторы позволяют управлять выполнением определенных действий в зависимости от заданных условий. Учащиеся изучают такие условные операторы, как if-else, и учатся использовать их для принятия решений в своих программах.
10. **Циклы:** Циклы позволяют повторять определенные действия или блоки кода несколько раз. Учащиеся изучают различные типы циклов, такие как циклы "повторить N раз" или "пока условие выполняется", и учатся использовать их для эффективного выполнения повторяющихся задач.
11. **Проектная деятельность:** Это метод обучения, основанный на организации и выполнении реальных проектов, которые требуют активного участия учащихся и интеграции различных знаний, навыков и ресурсов для достижения конкретных целей. В контексте программы "Юный программист", проектная деятельность представляет собой процесс, в котором учащиеся создают свои собственные проекты, игры и анимации на языке программирования Scratch.

Освоение этих ключевых понятий позволяет детям освоить основы программирования и развить навыки вычислительного мышления, креативности и логического мышления.

Направленность (профиль) программы

Дополнительная общеобразовательная общеразвивающая программа «Юный программист» имеет техническую направленность, практико-ориентированный характер обучения и ориентирована на введение детей в мир программирования с использованием языка программирования Scratch. Таким образом, профиль программы сфокусирован на техническом обучении программированию с использованием языка Scratch и развитии связанных с этим навыков и компетенций у детей.

Уровень освоения программы

Уровень освоения программы "Юный программист" - базовый. Программа разработана таким образом, чтобы дети, не имеющие предварительного опыта программирования, могли успешно освоить основы программирования на языке Scratch.

В ходе обучения на базовом уровне учащиеся ознакомятся с основными понятиями программирования, блоками программирования в Scratch, спрайтами и их взаимодействием, условными операторами и циклами. Они изучат основные принципы создания алгоритмов и решения задач с помощью программирования на языке Scratch.

Программа "Юный программист" на базовом уровне позволяет детям овладеть базовыми навыками программирования и стимулирует их к развитию логического мышления, творческого мышления и проблемного мышления. Они смогут создавать простые проекты, игры и анимации на Scratch и расширять свои возможности по мере продвижения в программе.

Важно отметить, что базовый уровень не ограничивается только начальными навыками, а представляет собой основу для дальнейшего развития в программировании и более сложных проектов.

Актуальность образовательной программы

Начало 21 века ознаменовано бурным развитием ИТ-технологий. Рост и развитие таких компаний как Яндекс, Google, Apple, Facebook подтверждают это. Мировые лидеры ИТ-индустрии периодически обращаются к школьникам с призывом изучать программирование. Становится понятно, что чем раньше ученик начнет овладевать навыками программирования, тем больший запас знаний и технологий он получит к моменту выбора основного рода деятельности. Даже если в будущем карьерный путь ребенка не будет связан с программированием, умение разбираться в сложных системах и взаимодействовать с новыми технологиями ему пригодится в любой сфере, ведь цифровые технологии используются повсеместно.

Образовательная программа "Юный программист" остается актуальной и значимой в современном образовательном контексте. Вот несколько причин, почему эта программа остается актуальной:

1. Развитие навыков вычислительного мышления: В современном мире компьютеры и технологии играют все более важную роль. Программа помогает развить у детей навыки вычислительного мышления, которые включают логическое и алгоритмическое мышление, способность анализировать и решать проблемы, а также творческое и инновационное мышление.

2. Подготовка к цифровой экономике: В современном информационном обществе программирование и цифровые навыки становятся все более важными. Обучение программированию на языке Scratch позволяет детям

понять основы программирования, а также развить навыки, необходимые для работы с технологиями в будущем. Это может открыть для них широкий спектр карьерных возможностей в области информационных технологий.

3. Стимулирование креативности и самовыражения: Программа активно привлекает детей к творческому процессу. Создание собственных проектов, игр и анимаций на Scratch развивает их креативность, способность генерировать идеи и превращать их в действительность. Это позволяет детям выразить себя и развить свои творческие потенциалы.

4. Интерактивное и наглядное обучение: Scratch предоставляет визуальную и интерактивную среду программирования, которая делает обучение более доступным и интересным для детей. Они могут видеть результат своей работы непосредственно на экране и мгновенно тестировать свои идеи. Это способствует более глубокому пониманию программирования и повышает мотивацию учащихся.

5. Развитие ключевых навыков: Программа также развивает ряд ключевых навыков, которые важны не только в программировании, но и в других сферах жизни. К таким навыкам относятся логическое мышление, решение проблем, творческое мышление, коммуникация, сотрудничество и критическое мышление.

В целом, программа "Юный программист" актуальна, поскольку помогает детям освоить основы программирования, развить вычислительное мышление и подготовить их к цифровому будущему, а также развить навыки, которые они смогут применить в различных сферах своей жизни.

Педагогическая целесообразность образовательной программы

Образовательная программа "Юный программист" обладает рядом педагогических целесообразностей, которые делают ее значимой и полезной в образовательном процессе. Вот некоторые из них:

1. Развитие вычислительного мышления: Программа специально разработана для развития вычислительного мышления у детей. Вычислительное мышление включает в себя навыки анализа, решения проблем, логического и алгоритмического мышления, которые являются важными для успешной адаптации в цифровом обществе. Он помогает детям развить способность решать задачи системно и логически мыслить.

2. Технологическая грамотность: В современном мире технологии играют ключевую роль во многих сферах жизни. Программа помогает детям развить технологическую грамотность, что означает умение использовать и адаптировать технологии в своих задачах и проектах. Они изучают основы программирования и приобретают практические навыки, которые могут применять в различных контекстах.

3. Творческое самовыражение: Программа поощряет творческое самовыражение детей. Они имеют возможность создавать собственные проекты, игры и анимации, используя язык программирования Scratch. Это стимулирует их фантазию, креативность и умение превращать свои идеи в

конкретные продукты. Программа предоставляет детям пространство для самовыражения и развития их индивидуальности.

4. Коллективная работа и сотрудничество: Программа поддерживает коллективную работу и сотрудничество. Дети могут работать в группах, обмениваться идеями, делиться знаниями и решать задачи вместе. Это способствует развитию коммуникативных и социальных навыков, а также способности работать в команде.

5. Мотивация и интерес к обучению: Программа предлагает детям интересный и захватывающий способ изучения программирования. Scratch предоставляет визуальную и интерактивную среду, которая делает обучение более привлекательным для детей. Они могут непосредственно видеть результаты своей работы, создавать игры и анимации, что повышает их мотивацию и интерес к обучению.

Образовательная программа "Юный программист" обладает педагогической целесообразностью, поскольку она развивает важные навыки и компетенции у детей, помогает им успешно адаптироваться в цифровом обществе и развивает их творческий потенциал, применение принципов программирования и алгоритмизации в учебном процессе формирует инженерный подход к решению задач, дает возможность развития творческого мышления у детей, привлекает школьников к исследованиям в междисциплинарных областях.

Практическая значимость образовательной программы

Образовательная программа "Юный программист" имеет значительную практическую значимость, поскольку она помогает обучающимся приобрести ценные навыки и компетенции. Вот некоторые практические навыки, которые обучающиеся получают после прохождения обучения по программе:

1. Навыки программирования: Обучающиеся осваивают основы программирования на языке Scratch. Они научатся создавать алгоритмы, использовать блоки программирования, работать с переменными, условиями и циклами. Эти навыки будут полезны не только в области информационных технологий, но и во многих других сферах, где требуется автоматизация и решение задач с помощью компьютеров.

2. Вычислительное мышление: Программа развивает навыки вычислительного мышления, которые включают логическое и алгоритмическое мышление, аналитическое мышление и способность решать проблемы. Обучающиеся научатся разбивать сложные задачи на более простые шаги, анализировать их и разрабатывать соответствующие алгоритмы.

3. Творческое мышление и самовыражение: При создании собственных проектов, игр и анимаций на Scratch, обучающиеся развивают свое творческое мышление. Они учатся генерировать идеи, преобразовывать их в конкретные проекты и выражать свою индивидуальность через программирование.

4. Работа в команде и коммуникация: Программа поощряет коллективную работу и сотрудничество. Обучающиеся будут работать в группах, обмениваться идеями, решать задачи вместе и учиться коммуницировать эффективно.

5. Презентационные навыки: В процессе создания проектов на Scratch, обучающиеся будут иметь возможность представлять и публично представлять свои работы. Это развивает навыки публичных выступлений, уверенность в себе и способность эффективно представлять свои идеи и результаты работы.

Практические навыки, полученные обучающимися, являются ценными в цифровой эпохе и могут быть применены в различных сферах, включая информационные технологии, инженерию, науку, медиа и многое другое.

Принципы отбора содержания образовательной программы

Принципы отбора содержания образовательной программы "Юный программист" основаны на:

1. Логическая последовательность: Содержание программы разделено на логически связанные разделы, обеспечивающие последовательное освоение материала. Каждый раздел строится на основе предыдущих знаний и навыков, обеспечивая прогрессивное развитие учащихся.

2. Связь с основной образовательной программой: Программа разработана с учетом связей с основной образовательной программой на определенной ступени образования. Она помогает развить ключевые компетенции, такие как вычислительное мышление, творческое мышление и коллективная работа, которые являются важными для учебного процесса в общем образовании.

3. Межпредметные связи: Программа активно взаимодействует с другими предметами, такими как информатика, математика, физика и т.д. Обучающиеся могут применять полученные знания и навыки в других учебных предметах, что способствует интеграции и применению программирования в различных контекстах.

4. Преемственность: Программа имеет преемственность с другими программами, связанными с программированием и информационными технологиями. Она может служить основой для продолжения обучения в более продвинутых программных продуктах и языках программирования.

5. Взаимодействие с другими организациями и родителями: Программа предполагает взаимодействие с другими образовательными организациями, такими как школы, клубы программирования или образовательные центры. Она может предоставлять возможности для организации соревнований, мероприятий или совместных проектов. Кроме того, включение родителей в обучающий процесс может стимулировать интерес и поддержку детей в изучении программирования.

Принципы отбора содержания образовательной программы "Юный программист" обеспечивают логическую и последовательную структуру

обучения, связи с основным образованием, интеграцию с другими предметами и практическую значимость для учащихся. Такой подход способствует эффективному освоению материала и развитию необходимых навыков для современного цифрового мира.

Отличительные особенности программы

Программа "Юный программист" имеет ряд отличительных особенностей, которые делают ее уникальной и инновационной:

1. Индивидуальные образовательные маршруты: Программа предлагает гибкий подход к обучению, позволяющий адаптировать содержание и уровень сложности в соответствии с потребностями и способностями каждого обучающегося.

2. Расширенное содержание программы: Программа предлагает дополнительные разделы и темы, которые расширяют учебный материал, включая более сложные алгоритмы, продвинутые функции языка Scratch и использование дополнительных инструментов и ресурсов для создания проектов.

3. Региональный компонент: Программа может включать региональный компонент, который отражает особенности и потребности конкретной местности или культуры. Это может быть связано с созданием проектов, отражающих местную историю, культуру или проблемы, а также с привлечением местных экспертов и ресурсов.

4. Проектная и исследовательская деятельность: Программа активно использует проектные и исследовательские методы обучения. Обучающиеся не только изучают концепции программирования, но и применяют их на практике через создание собственных проектов и исследование новых идей и возможностей.

5. Стимулирование творчества и инноваций: Программа поддерживает творческое мышление и развитие инновационных навыков. Обучающиеся могут создавать оригинальные проекты, игры и анимации, применяя свою фантазию и уникальные идеи.

6. Взаимодействие с родителями: Программа активно включает родителей в образовательный процесс, предоставляя им информацию о проектах и конкурсах, где родители могут увидеть достижения своих детей и поддержать их в учебе.

Все эти особенности программы "Юный программист" направлены на развитие личностных качеств обучающихся, их творческого и критического мышления, умений работать в команде и решать проблемы, а также на поддержку их интересов и потребностей в области программирования и цифровых технологий.

Цель образовательной программы

Цель образовательной программы "Юный программист" заключается в создании высокотехнологичной образовательной среды для развития и самореализации детей, воспитании высоконравственной, гармонично развитой и социально-ответственной личности, а также возрождения престижа инженерных и научных профессий, подготовка кадрового резерва.

Образовательные цели программы включают:

1. Формирование знаний: Обеспечение обучающимся необходимыми знаниями о языке программирования Scratch, алгоритмах, базовых концепциях программирования и цифровых технологиях. Программа стремится развить у детей понимание основных принципов программирования и способность применять их для создания собственных проектов.

2. Развитие навыков: Развитие практических навыков программирования на Scratch, включая способность создавать интерактивные проекты, игры, анимации и другие приложения. Программа способствует развитию творческого мышления, аналитических и проблемно-ориентированных навыков, коммуникативных и коллаборативных умений.

3. Способствование цифровой грамотности: Программа помогает развить у обучающихся навыки работы с компьютером и цифровыми технологиями, а также повысить их понимание важности безопасности в онлайн-среде. Обучающиеся осваивают основные принципы программирования, логического мышления и алгоритмического мышления.

Социально-педагогические цели программы включают:

1. Социализация и социальная адаптация: Помощь обучающимся в развитии навыков социального взаимодействия, сотрудничества и коммуникации. Программа способствует формированию у обучающихся уважения к другим, развитию эмпатии и способности работать в коллективе.

2. Поддержка творческого потенциала: Программа стремится развить творческое мышление, самовыражение и оригинальность у обучающихся. Обучение на Scratch стимулирует детей к созданию собственных проектов, что позволяет им развивать свои таланты и выражать свою индивидуальность.

3. Поддержка развития личности: Программа помогает детям развивать самостоятельность, самоорганизацию и ответственность за свои действия. Она способствует развитию критического мышления, решения проблем и принятия информированных решений.

Образовательная программа "Юный программист" объединяет образовательные и социально-педагогические цели, чтобы обеспечить разносторонний и гармоничный развитие детей, а также помочь им освоить ключевые компетенции в области программирования и цифровых технологий.

Задачи образовательной программы

Задачи образовательной программы "Юный программист" можно классифицировать следующим образом:

Личностные задачи:

1. Формирование интереса к программированию и цифровым технологиям.
2. Развитие творческого мышления и самовыражения через создание собственных проектов.
3. Воспитание ответственности и умения работать в команде.
4. Развитие навыков коммуникации и сотрудничества.
5. Поддержка развития самоуверенности и самооценки.

Метапредметные задачи:

1. Развитие логического и алгоритмического мышления.
2. Стимулирование саморазвития и самообразования.
3. Поддержка активности и инициативности в учебной и творческой деятельности.
4. Развитие навыков самоорганизации и планирования.
5. Формирование потребности в постоянном обновлении и расширении своих знаний и умений.

Образовательные (предметные) задачи:

1. Овладение основами программирования на языке Scratch.
2. Приобретение знаний о базовых концепциях программирования, таких как циклы, условия, переменные.
3. Развитие навыков создания интерактивных проектов, игр и анимаций.
4. Понимание принципов работы алгоритмов и их применение в решении задач.
5. Овладение навыками отладки и исправления ошибок в программном коде.

Развивающие задачи:

1. Развитие творческого мышления и воображения.
2. Улучшение способности к аналитическому мышлению и проблемному решению.
3. Развитие логического и пространственного мышления.
4. Развитие навыков абстрактного и системного мышления.
5. Способствование развитию критического мышления и умения оценивать свою работу.

Воспитательные задачи:

1. Формирование ценностных ориентаций, таких как творчество, сотрудничество и уважение к другим.
2. Воспитание у учащихся ответственности за свои действия в цифровом пространстве.
3. Поддержка развития этических принципов и цифровой грамотности.

4. Подготовка к осознанному и безопасному использованию цифровых технологий.

5. Содействие формированию позитивного отношения к обучению и саморазвитию.

Формулировки задач тесно связаны с планируемыми результатами образовательной программы и направлены на достижение целей, связанных с формированием ключевых компетенций, развитием личностных качеств и овладением практическими навыками в области программирования и творчества.

Психолого-педагогические характеристики обучающихся, участвующих в реализации образовательной программы

Образовательная программа "Юный программист" ориентирована на обучающихся определенного возрастного диапазона и предполагает следующие психолого-педагогические характеристики участвующих в ней обучающихся:

1. Возраст: Программа рассчитана на детей и подростков в возрасте от 8 до 14 лет. Этот возрастной диапазон связан с особенностями развития когнитивных функций, логического мышления и способности к творческому самовыражению.

2. Предварительная подготовка: Обучающиеся не требуют специальной предварительной подготовки в области программирования. Программа разработана для начинающих, и все необходимые знания и навыки будут постепенно введены и объяснены в процессе обучения.

3. Уровень образования: Программа может быть реализована в рамках дополнительного образования или внеурочной деятельности. Она не требует определенного уровня образования и может быть доступна обучающимся с различными уровнями знаний и навыков.

4. Сформированность интересов: Программа предполагает формирование интереса к программированию и цифровым технологиям. Однако предварительное знакомство или увлечение программированием не является обязательным условием для участия.

5. Набор обучающихся: На образовательную программу принимаются все желающие, соответствующие возрастным ограничениям, и не требует специфической предварительной подготовки. Однако, так как для успешного освоения программы требуются средние и высокие когнитивные способности, то набор детей в объединение осуществляется на основании собеседования и входного тестирования.

Особенности организации образовательного процесса

Учащиеся принимаются в группу, без каких-либо первоначальных знаний в области программирования. Количество обучающихся в группе 10-15 человек, обучение очное.

Вид деятельности группы – профильная, состав - постоянный.

Курс доступен школьнику средних и высоких когнитивных способностей.

Формы обучения по образовательной программе

Форма обучения - очная. Обучение проводится непосредственно в учебных заведениях или специализированных центрах, где обучающиеся присутствуют на занятиях в определенное время и месте. В рамках очной формы обучения учащиеся могут иметь возможность активного взаимодействия с преподавателем и другими учащимися, а также использования специализированного оборудования и материалов.

Весь учебный материал, содержащийся в программе, можно разделить на теоретический и практический. Теоретическим вопросам отводится не более 30 минут на каждом двухчасовом занятии. Теоретическая часть проходит в виде лекций, разбираются возможные задачи и методы их решения, практическая часть – закрепление пройденного материала, проверка знаний обучающихся, а также выполнение практических заданий.

Программа предусматривает использование следующих форм работы:

- фронтальной - подача учебного материала всему коллективу учеников;
- индивидуальной - самостоятельная работа обучающихся с оказанием помощи педагогом при возникновении затруднения, не уменьшая активности учеников и содействуя выработке навыков самостоятельной работы;
- групповой - когда учащимся предоставляется возможность самостоятельно построить свою деятельность на основе принципа взаимозаменяемости, ощутить помощь со стороны, учесть возможности каждого на конкретном этапе деятельности.

Режим занятий, периодичность и продолжительность занятий

Общее количество часов в год – 144 ч. Занятия проходят два раза в неделю по два учебных часа с перерывом. Продолжительность занятий исчисляется в академических часах – 45 минут, между занятиями установлены 10-минутные перемены.

Объем и срок освоения образовательной программы

Дополнительная образовательная программа рассчитана на изучение курса «Юный программист» в системе дополнительного образования детей в течение одного учебного года на 144 часа аудиторных занятий.

Учебная нагрузка рассчитана на весь календарный год. В период школьных каникул внеаудиторные занятия с учащимися объединения проводятся в разных видах и формах: экскурсии в музеи города, на выставки, в парковую зону; участие в работе летней школы, летнего оздоровительного лагеря с дневным пребыванием детей, участие в организации праздников и акций, самостоятельная проектная деятельность.

Основные методы обучения

В современных условиях использования технологий, процесс обучения требует адаптации методологии с учетом новых ресурсов и их специфики. Участие в образовательных мероприятиях предоставляет учащимся возможность испытать свои навыки в конкурсной среде и продемонстрировать свои успехи и достижения. При организации таких мероприятий сочетаются индивидуальная и групповая деятельность, творчество, сотрудничество разных возрастных групп, возможность соревнования в команде, рефлексивная практика, а также время для отдыха, неформального общения и релаксации. Это способствует повышению познавательной активности учащихся, развитию их потенциала и формированию умения конструктивно взаимодействовать друг с другом.

Каждое занятие включает как теоретическую, так и практическую части для закрепления материала. Такой подход позволяет учащимся развивать такие качества, как умение решать практические задачи, постановка целей и планирование их достижения. Занятие условно делится на три части, которые вместе составляют единое целое:

1. Организационные моменты, включая представление нового материала, инструктаж, планирование и распределение работы между учащимися на данное занятие.

2. Практическая работа учащихся, выполняемая индивидуально или в группе, самостоятельно или под руководством педагога. Здесь закрепляется теоретический материал, отрабатываются навыки и приемы, формируются успешные методы профессиональной деятельности.

3. Анализ проделанной работы и подведение итогов, который является коллективной деятельностью и включает аналитическую работу каждого учащегося, педагога и всей группы. Широко используются творческие методы, которые придают смысл обучению, мотивируют учащихся на дальнейшее развитие и позволяют интересно и доступно изучать материал.

Метод дискуссии обучает учащихся защищать свои мнения и слушать других. Например, при изготовлении... учащимся требуется высказаться и аргументированно защитить свою работу. Учебные дискуссии обогащают представления учащихся по теме, систематизируют и укрепляют их знания.

Деловая игра используется как средство моделирования различных условий профессиональной деятельности, включая экстремальные ситуации, и позволяет учащимся рассмотреть возможность выбора данной сферы деятельности в будущем.

Ролевая игра позволяет участникам вжиться в предложенную ситуацию, реально ощутить различные состояния, анализировать последствия различных действий и принимать решения.

Существуют методы, основанные на активной деятельности учащихся:

- Исследовательский метод, который предполагает самостоятельную творческую работу учащихся.

- Репродуктивный метод, где учащиеся воспроизводят полученные знания и овладевают способами деятельности.

- Объяснительно-иллюстративный метод, в рамках которого дети воспринимают и усваивают готовую информацию.

- Частично-поисковый метод, где дети участвуют в коллективном поиске и решении поставленных задач совместно с педагогом.

Также существуют методы, основанные на способе организации занятий:

- Наглядный метод, который включает использование мультимедийных материалов, иллюстраций, наблюдение и демонстрацию педагогом, работу по образцу и другие средства визуализации.

- Практический метод, где учащиеся выполняют работы по инструкционным чертежам, схемам и т.д.

- Словесный метод, включающий устное изложение, беседу, рассказ, лекцию и т.д.

В рамках образовательного процесса используются различные методы, основанные на форме организации деятельности учащихся на занятиях, такие как:

- Проблемное изложение и исследовательский метод, которые способствуют развитию самостоятельности мышления, творческого подхода к выполнению заданий и исследовательских навыков.

- Объяснительно-иллюстративный метод, который помогает формированию знаний и представлений о действиях.

- Репродуктивный метод, который направлен на формирование умений, навыков и способов деятельности.

- Словесный метод, включающий рассказы, объяснения, беседы, лекции, исследование и формирование сознания учащихся.

- Стимулирующий метод, включающий соревнования, выставки и поощрения.

Таким образом, в современных условиях обучения используются разнообразные методы, которые позволяют адаптировать процесс обучения к новым ресурсам и особенностям, активизировать познавательную активность учащихся, развивать их потенциал, способствовать конструктивному взаимодействию и мотивировать на дальнейшее развитие.

Планируемые результаты

В процессе освоения программы "Юный программист" учащиеся достигнут следующих планируемых образовательных, развивающих и воспитательных результатов:

Образовательные результаты:

- Обучающийся сможет объяснить основные понятия и принципы языка Scratch.

- Обучающийся сможет создавать и настраивать спрайты, использовать блоки программирования и создавать алгоритмы для решения простых задач.

- Обучающийся сможет создавать простые интерактивные проекты с использованием графического интерфейса Scratch.

- Обучающийся сможет использовать события, условия и циклы для создания интерактивности в своих проектах.

Развивающие результаты:

- Обучающийся будет развивать способность генерировать оригинальные идеи и решать проблемы с помощью программирования в языке Scratch.

- Обучающийся будет учиться критически мыслить, анализировать и улучшать свои проекты.

- Обучающийся будет развивать навык построения последовательностей действий и алгоритмического мышления при создании программ в Scratch.

- Обучающийся будет улучшать способность предвидеть и анализировать результаты своих действий в программе.

Воспитательные результаты:

- Обучающийся будет развивать навык самостоятельного решения задач и принятия решений при создании проектов в Scratch.

- Обучающийся будет осознавать свою ответственность за свои действия в программе и их результаты.

- Обучающийся будет учиться работать в команде, обмениваться идеями и решениями с другими участниками программы.

- Обучающийся будет развивать навык эффективного общения и взаимодействия с другими участниками при создании и представлении своих проектов.

Механизм оценивания образовательных результатов

Механизм оценивания образовательных результатов для программы "Юный программист" включает следующие диагностические методики:

1. Наблюдение: Педагог может наблюдать за деятельностью обучающихся во время выполнения программных задач в Scratch. Это позволит оценить их способность применять изученные концепции и принципы, а также их уровень владения программными навыками.

2. Тестирование: Проведение тестов или квизов по темам, связанным с языком программирования Scratch, позволит оценить понимание обучающимися ключевых понятий, синтаксиса и функциональности Scratch.

3. Анализ продуктов деятельности: Изучение и анализ проектов, созданных обучающимися в Scratch, позволит оценить их способность применять основные концепции программирования, создавать интерактивные проекты и решать проблемы с использованием Scratch.

4. Оценка творческих достижений: Отслеживание и оценка творческих достижений обучающихся в создании оригинальных и интересных проектов в Scratch позволит оценить развитие их творческого мышления и способности к инновационным решениям.

Форма, порядок и периодичность проведения промежуточной аттестации могут быть определены следующим образом:

- Проведение регулярных наблюдений и анализа продуктов деятельности в течение всего курса обучения.
- Регулярное тестирование по завершению каждой темы или модуля для оценки понимания материала.
- Проведение заключительной оценки на основе создания и представления самостоятельных проектов в Scratch, включающих элементы творчества и применение изученных навыков.

Очень важно создать условие, чтобы каждый ученик чувствовал успешность в освоение программы, поэтому оценочная система не должна пугать и закрепощать учащихся, тесты составляются с элементами игры и юмора.

Формы подведения итогов реализации образовательной программы

Для выявления уровня усвоения содержания программы и своевременного внесения коррекции в образовательный процесс, проводится текущий контроль в виде контрольного среза знаний (итоговый проект) освоения программы в конце каждого раздела.

При оценивании результатов используют виды контроля: начальный или входной контроль, текущий контроль, промежуточный мониторинг, итоговая аттестация.

Начальный или входной контроль – проводится с целью определения уровня развития учащегося, выявление исходного уровня знаний и умений детей для корректировки программного материала.

Текущий контроль – проводится с целью определения степени усвоения обучающимися учебного материала.

Промежуточный мониторинг – проводится с целью оценки качества освоения учащимися программы по итогам учебного периода (раздела программы) в виде теста, беседы, опроса, конкурсного задания.

Итоговая аттестация – проводится с целью оценки качества освоения уровня знаний, умений и навыков, измерения уровня развития детей, выявления творческого роста и интереса к занятиям, определения степени достижения результатов обучения, в виде итогового годового проекта.

Кроме классической схемы подведения итогов, необходимо стимулировать участие в конкурсах, выставках, фестивалях детского творчества, связанных с информационными технологиями различных уровней, проводить в процессе обучения: мини-конкурсы, коллективные творческие дела и проекты, познавательные экскурсии (технические выставки, учреждения с информационными технологиями).

Организационно-педагогические условия реализации образовательной программы.

Научно-методическое обеспечение реализации программы направлено на обеспечение широкого, постоянного и устойчивого доступа для всех участников образовательного процесса к любой информации, связанной с реализацией общеразвивающей программы, планируемыми результатами, организацией образовательного процесса и условиями его осуществления.

Социально-психологические условия реализации образовательной программы обеспечивают:

- учет специфики возрастного психофизического развития обучающихся;
- вариативность направлений сопровождения участников образовательного процесса (сохранение и укрепление психологического здоровья обучающихся);
- формирование ценности здоровья и безопасного образа жизни; дифференциация и индивидуализация обучения; мониторинг возможностей и способностей обучающихся, выявление и поддержка одаренных детей, детей с ограниченными возможностями здоровья;
- формирование коммуникативных навыков в разновозрастной среде и среде сверстников.

Материально-техническое обеспечение

Наименование объектов и средств материально-технического обеспечения	Количество	Примечания
Книгопечатная продукция (кол-во на группу)		
Образовательная программа	1 шт.	
Рабочая программа	1 шт.	
Методические разработки	15 шт.	Методические разработки занятий, оценочные материалы.
Информационно-коммуникативные средства (кол-во на группу)		
сайты	1. Scratch.mit.edu	
	2. Scratch.ru	
Технические средства обучения (кол-во на группу)		
Интерактивная доска	1 шт	
Проектор	1 шт	
Компьютер (ноутбук)	15 шт	
Экранно-звуковые пособия, программное обеспечение (кол-во на группу)		
Программное обеспечение	Программное обеспечение Scratch	распространяется бесплатно

Электронные методические разработки	1 шт. на каждого ученика отдельно	Комплект обучающих материалов
Учебно-практическое оборудование (на группу)		
столы, парты	15 шт.	
стулья	15 шт.	
стеллажи для оборудования	6 шт.	
мебель для книг и оборудования	1 шт.	

Кадровые условия.

Педагог дополнительного образования, реализующий данную программу, должен иметь высшее профессиональное образование или среднее профессиональное образование в области, соответствующей профилю кружка, без предъявления требований к стажу работы, а также дополнительное профессиональное образование по направлению «Образование и педагогика» без предъявления требований к стажу работы.

Методическое обеспечение программы

Выбор методов и форм для реализации программы определяется:

- Поставленными целями и задачами.
- Следующими принципами обучения: от практической деятельности к внутреннему развитию всех качеств личности.
- Возможностями учащихся на данном этапе (возраст, уровень подготовки, мотивации и др.)
- Наличием соответствующей материальной базы.

Для реализации программы используются следующие *методы*:

- Развивающего обучения (проблемный, поисковый, творческий)
- Дифференцированного обучения (уровневые, индивидуальные задания)
- Игровые программы.

Одними из важнейших методов обучения по данной программе являются методы стимулирования познавательной деятельности: метод дискуссии, создание ситуации успеха.

Приоритетная технология обучения по программе – метод проектов. Обучающиеся выполняют исследовательские, творческие проекты при завершении тем, а также творческий проект, как итоговую зачетную работу, интегрирующую знания и умения по всему материалу за год.

Основной формой обучения по данной программе является учебно-практическая деятельность обучающихся. Приоритетными методами её организации служат практические, поисково-творческие работы. Все виды практической деятельности в программе направлены на освоение различных технологий работы с информацией и компьютером как инструментом обработки информации.

Дидактическое обеспечение занятий: онлайн задания, обучающие видео-уроки, учебная литература, опорные конспекты, компьютерные обучающие программы, тесты, логические схемы, различные виды раздаточных материалов, наглядные пособия, положения о конкурсах.

Оценочная система.

Вся оценочная система делится на три уровня сложности:

1. Обучающийся может ответить на общие вопросы по большинству тем, с помощью педагога может создать и объяснить работу небольшой программы (на выбор).

2. Обучающийся отвечает на все вопросы, поднимаемые за период обучения. Может самостоятельно создать и объяснить работу любой программы из предложенных ему.

3. Обучающийся отвечает на все вопросы, поднимаемые за период обучения. Может самостоятельно создать и объяснить алгоритм работы и особенности любой из предложенных программ. Но, располагает сведениями сверх программы, проявляет интерес к теме. Проявил инициативу при выполнении конкурсной работы или проекта. Вносил предложения, имеющие смысл.

Также, успехи обучающегося оцениваются по разделам:

- теория;
- практика;

Содержание программы *1 год обучения (144 часа, 4 часа в неделю)*

РАЗДЕЛ “БАЗОВЫЙ”

1. **Тест:** входящий тест.

Теория: Техника безопасности в компьютерном классе. Ознакомление с курсом "Юный программист Scratch".

2. **Викторина:** повторение ранее изученного материала.

Теория: Основные понятия Scratch. Знакомство с интерфейсом. Сцена 480 x 360.

3. **Викторина:** повторение ранее изученного материала.

Теория: Путешествие в сообщество Scratch.

Практика: Регистрация на сайте scratch.mit.edu.

4. **Викторина:** повторение ранее изученного материала.

Теория: Библиотеки спрайтов и сцен. Импорт спрайтов и сцен из интернета, поддерживаемые форматы файлов. Создание, сохранение и открытие проектов.

Практика: пробуем добавлять разные спрайты и сцены из библиотек и интернета, сохранять и открывать свои проекты в своем аккаунте и на компьютере.

5. **Викторина:** повторение ранее изученного материала.

Теория: Scratch - графический редактор. Использование клавиши Shift.

Практика: Придумываем и создаем в графическом редакторе своего спрайта с несколькими костюмами и свой фон.

6. **Викторина:** повторение ранее изученного материала.

Теория: Блоки внешнего вид (говорить, сказать, думать). Блок управление (ждать). Блок события (когда нажат «Зеленый флажок», когда клавиша ... нажата)

Практика: создание диалогов между спрайтами с использованием изученных блоков.

7. **Викторина:** повторение ранее изученного материала.

Теория: Создание диалогов, методы расчета пауз для спрайтов и сцен.

Практика: Задание создать осмысленный диалог между несколькими спрайтами.

8. **Викторина:** повторение ранее изученного материала.

Теория: Направления. Блоки движения (идти, повернуть на ..., повернуться в направлении, если касается края..., установить способ вращения).

Практика: Создаем с помощью изученных блоков программы в которой:

- спрайт движется прямо от края сцены до края.
- спрайт движется по периметру сцены и на краях поворачивается на 90 °.
- спрайт движется по сцене и поворачивает на 15, 30, 45 ° влево/вправо.

- спрайт движется по сцене и поворачивает то влево, то вправо на разное число градусов.
 - во все вышеперечисленные программы добавляем блоки “если касается края” и “установить способ вращения” и смотрим что получается.
9. **Викторина:** повторение ранее изученного материала.
Теория: Изучаем циклы (повторить 10 раз, повторять всегда). Изучаем «фишечки» scratch для более быстрой и удобной работы. (Клонирование костюмов, клонирование скриптов между спрайтами, использование рюкзака и т.д.).
Практика: С помощью изученных циклов оптимизируем задания из урока №7.
10. **Викторина:** повторение ранее изученного материала.
Теория: Блоки внешнего вида (изменить костюм на ..., следующий костюм, изменить фон на ..., следующий фон, показаться, спрятаться).
Практика: Делаем проект “Распускающийся цветок”. (если останется время, то пробуем для спрайтов, у которых есть много костюмов, создать анимацию. Багаем от края до края, бегаем по кругу, бегаем с использованием блоков “следующий костюм”, “следующий фон”, “показаться”, “спрятаться”).
11. **Викторина:** повторение ранее изученного материала.
Теория: Анимация спрайтов. Правила и способы создания анимации в Scratch.
Практика: Выбираем несколько спрайтов с разными костюмами, сцену и с ними придумываем игровой сценарий, потом все программируем.
12. **Викторина:** повторение ранее изученного материала.
Теория: Пространство (координаты и направления). Отрицательные числа.
Практика: Делаем простую игру - воздушные шарики появляются в случайном месте, а нам надо ввести координаты центра шарика и попытаться его сбить. Играем в игру, тренируем определение координат.
13. **Викторина:** повторение ранее изученного материала.

Теория: Блоки движения с координатами (перейти на случ.положение, перейти в х.. у., плыть на случ.положение, плыть в х.. у., установить/изменить х/у, положение х/у, направление).

Практика: Написать программу когда шарик переходит по координатам. (четырьмя разными способами)

14. **Викторина:** повторение ранее изученного материала.

Теория: Условия. Блоки управления (если, если-иначе). Блоки операторы. Сравнения (больше, меньше, равно). Блок сенсоры (касается...).

Практика: Напишем игру “Простой лабиринт”, где надо провести по лабиринту шарик.

15. **Викторина:** повторение ранее изученного материала.

Теория: Бинарная логика (И/ИЛИ/НЕ).

Практика: Переделаем (упростим) игру “Простой лабиринт” с помощью блоков “И” / “ИЛИ” / “НЕ”

16. **Викторина:** повторение ранее изученного материала.

Теория: Основные арифметические операции. Блоки операторы (плюс, минус, умножить, делить, выдать случайное число). Блоки внешнего вида (изменить размер, установить размер).

Практика: Сделаем небольшую анимацию. Несколько спрайтов размером 10%, если выпадает случайное число (от 1 до 10) больше 5, то у спрайта анимация увеличения размера, когда размер становится 300%, спрайт начинает все заново.

17. **Викторина:** повторение ранее изученного материала.

Теория: Установка начальных значений (координаты, направление, видимость, эффекты, способ вращения, костюм, размер, фон и т.д.)

Практика: Придумать сценарий анимации с несколькими спрайтами и несколькими сценами. Обязательно предусмотреть для каждого запуска анимации - установку начальных значений.

18. **Викторина:** повторение ранее изученного материала.

Теория: Сообщения в мультипликации Scratch. Блоки события (когда я получу сообщение, передать сообщение, передать сообщение и ждать).

Практика: Сделаем анимацию “Сбор команды ракеты и полет на луну” (команду будем собирать с помощью отправки сообщений).

19. **Викторина:** повторение ранее изученного материала.

Проектная деятельность: Придумать самостоятельно сценарий анимации с несколькими спрайтами и сценами и написать программу в соответствии со сценарием. Возможна помощь товарищей и преподавателя.

РАЗДЕЛ “ПРОДВИНУТЫЙ”

20. **Викторина:** повторение ранее изученного материала.

Теория: Понятие алгоритм, виды алгоритмов.

Практика: Придумать и реализовать простые линейный, ветвящийся и циклический алгоритмы.

21. **Викторина:** повторение ранее изученного материала.

Теория: События. Блоки события (рассмотреть все оставшиеся события). Взаимодействие спрайтов и обработка событий. Блоки сенсоры (касается..., касается цвета, цвет...касается цвета..., клавиша ... нажата). Управление движением спрайта.

Практика: Делаем усложненную игру “Лабиринт” с противниками, с касанием стен, с препятствиями, с ключами, с дверями.

22. **Викторина:** повторение ранее изученного материала.

Теория: Графические эффекты. Слои и управление ими. Блоки внешний вид (перейти на ... слой, перейти на х слоя).

Практика: Делаем проект “Кривое зеркало”.

23. **Викторина:** повторение ранее изученного материала.

Теория: Блоки управления (ждать до..., стоп всё). Процедуры (Другие блоки). Оптимизация процедур.

Практика: Делаем проект «Кот-акробат». Используем процедуры, где каждая процедура будет выполнять разные акробатические прыжки.

24. **Викторина:** повторение ранее изученного материала.

Теория: Цикл Повторять пока не....

Практика: Делаем проект “Угадай число” используя цикл “Повторять пока не...”

25. **Викторина:** повторение ранее изученного материала.

Теория: Области координат.

Практика: Программируем движение спрайта только в определенной области

26. **Викторина:** повторение ранее изученного материала.

Теория: Переменные и их виды.

Практика: Делаем проект "Магазин". Используем переменные для хранения денег и предметов.

27. **Викторина:** повторение ранее изученного материала.

Теория: Общие, локальные и облачные переменные. Как получить доступ к использованию облачных переменных.

Практика: Делаем проект игры "Ловим мяч чашкой". Используем переменные для подсчета очков.

28. **Викторина:** повторение ранее изученного материала.

Теория: Ведение расчетов с помощью переменных.

Практика: Делаем проект "Простой калькулятор".

29. **Викторина:** повторение ранее изученного материала.

Теория: Внутренние переменные Scratch (положение x, положение y, направление, костюм номер, фон номер, размер, громкость звука, ответ, x мыши, y мыши, громкость, таймер, текущий год, дней с 2000 г, имя пользователя).

Практика: Делаем проект "Следим за мышкой".

30. **Тест:** Полугодовое тестирование на усвоение ранее изученного материала.

Теория: Правила поведения во время зимних каникул.

Беседа: Рефлексия о проделанной работе за полгода, что получилось, что не получилось, что понравилось, а что не понравилось, что хотелось бы сделать и чему научиться.

31. **Викторина:** повторение ранее изученного материала.

Теория: Управление состоянием через переменные.

Практика: Делаем проект "Движение фона".

32. **Викторина:** повторение ранее изученного материала.

Теория: Понятие объектов и классов и реализация через клоны.

Практика: Делаем проект “Змейка”.

33. **Викторина:** повторение ранее изученного материала.

Теория: Планирование движения клонов.

Практика: Делаем проект “Падающие снежинки”.

34. **Викторина:** повторение ранее изученного материала.

Теория: Блоки для использования с мышью (перейти на указатель мыши, повернуться к указателю мыши, мышь нажата?, х мыши, у мыши).

Практика: Делаем проект игры “Агар-агар”.

35. **Викторина:** повторение ранее изученного материала.

Теория: Понятие списка, задание значений.

Практика: Разбираем и реализуем проект игры “Смешные предложения”.

36. **Викторина:** повторение ранее изученного материала.

Теория: Изменение значений списка.

Практика: Доделяваем проект игры “Смешные предложения”.

37. **Викторина:** повторение ранее изученного материала.

Теория: Проход по списку.

Практика: Доделяваем проект игры “Смешные предложения”.

38. **Викторина:** повторение ранее изученного материала.

Теория: Блоки сенсоры (спросить и ждать + ответ).

Практика: С помощью изученных блоков сделаем викторину по математике.

39. **Викторина:** повторение ранее изученного материала.

Теория: Блоки сенсоры (расстояние до ..., фон от сцена).

Практика: Делаем проект игры “Танчики”.

40. **Викторина:** повторение ранее изученного материала.

Теория: Блоки операторы. (объединить, буква № в ..., длина ..., ... содержит ...?).

Практика: Делаем проект “Угадай слово”.

41. **Викторина:** повторение ранее изученного материала.
Теория: Блоки операторы. (mod, округлить, модуль от ...).
Практика: Делаем проект “Сложный калькулятор”
42. **Викторина:** повторение ранее изученного материала.
Теория: Работа со звуком. Блоки звук.
Практика: Изменяем готовые проекты, добавляя в них звуки.
43. **Викторина:** повторение ранее изученного материала.
Теория: Использование расширений и дополнительных блоков в Scratch. Дополнительные блоки "Перо".
Практика: Делаем проект “Графический редактор”.
44. **Викторина:** повторение ранее изученного материала.
Теория: Использование расширений и дополнительных блоков в Scratch. Дополнительные блоки "Перо".
Практика: Делаем проект “Рисуем пером фигуры”.
45. **Викторина:** повторение ранее изученного материала.
Теория: Использование расширений и дополнительных блоков в Scratch. Дополнительные блоки “Текст в речь”.
Практика: Переделываем проект “Смешные предложения” с помощью дополнительных блоков “Текст в речь”.
46. **Викторина:** повторение ранее изученного материала.
Теория: Использование расширений и дополнительных блоков в Scratch. Дополнительные блоки "Перевести".
Практика: Делаем проект “Переводчик иностранных слов”.
47. **Викторина:** повторение ранее изученного материала.
Теория: Использование расширений и дополнительных блоков в Scratch. Дополнительные блоки "Музыка".
Практика: Кодлируем мелодию “Елочка”

РАЗДЕЛ “ПРОЕКТНЫЙ”

48. **Викторина:** повторение ранее изученного материала.
Теория: Разбираем проект игры “Мемори”.

Проектная деятельность: Ученики самостоятельно составляют и реализуют алгоритм программы. Возможна помощь товарищей и преподавателя.

49. **Викторина:** повторение ранее изученного материала.

Теория: Разбираем проект игры “Арканойд”.

Проектная деятельность: Ученики самостоятельно составляют и реализуют алгоритм программы. Возможна помощь товарищей и преподавателя.

50. **Викторина:** повторение ранее изученного материала.

Проектная деятельность: Продолжаем проект “Арканойд”. Ученики самостоятельно составляют и реализуют алгоритм программы.

Возможна помощь товарищей и преподавателя.

51. **Викторина:** повторение ранее изученного материала.

Теория: Разбираем проект игры “Динозаврик Гугл”.

Проектная деятельность: Ученики самостоятельно составляют и реализуют алгоритм программы. Возможна помощь товарищей и преподавателя.

52. **Викторина:** повторение ранее изученного материала.

Проектная деятельность: Продолжаем проект “Динозаврик Гугл”.

Ученики самостоятельно составляют и реализуют алгоритм программы. Возможна помощь товарищей и преподавателя.

53. **Викторина:** повторение ранее изученного материала.

Теория: Разбираем проект игры “Морской бой”.

Проектная деятельность: Ученики самостоятельно составляют и реализуют алгоритм программы. Возможна помощь товарищей и преподавателя.

54. **Викторина:** повторение ранее изученного материала.

Проектная деятельность: Продолжаем проект “Морской бой”.

Ученики самостоятельно составляют и реализуют алгоритм программы. Возможна помощь товарищей и преподавателя.

55. **Викторина:** повторение ранее изученного материала.

Теория: Разбираем проект игры “Флэппи бёрд”.

Проектная деятельность: Ученики самостоятельно составляют и реализуют алгоритм программы. Возможна помощь товарищей и преподавателя.

56. **Викторина:** повторение ранее изученного материала.

Проектная деятельность: Продолжаем проект “Флэппи бёрд”.

Ученики самостоятельно составляют и реализуют алгоритм программы. Возможна помощь товарищей и преподавателя.

57. **Викторина:** повторение ранее изученного материала.

Теория: Разбираем проект игры “Платформер с несколькими уровнями”.

Проектная деятельность: Ученики самостоятельно составляют и реализуют алгоритм программы. Возможна помощь товарищей и преподавателя.

58. **Викторина:** повторение ранее изученного материала.

Проектная деятельность: Продолжаем проект “Платформер с несколькими уровнями”. Ученики самостоятельно составляют и реализуют алгоритм программы. Возможна помощь товарищей и преподавателя.

59. **Викторина:** повторение ранее изученного материала.

Проектная деятельность: Продолжаем проект “Платформер с несколькими уровнями”. Ученики самостоятельно составляют и реализуют алгоритм программы. Возможна помощь товарищей и преподавателя.

60. **Викторина:** повторение ранее изученного материала.

Теория: Разбираем проект игры “Раннер”.

Проектная деятельность: Ученики самостоятельно составляют и реализуют алгоритм программы. Возможна помощь товарищей и преподавателя.

61. **Викторина:** повторение ранее изученного материала.

Проектная деятельность: Продолжаем проект “Раннер”. Ученики самостоятельно составляют и реализуют алгоритм программы. Возможна помощь товарищей и преподавателя.

62. **Викторина:** повторение ранее изученного материала.
Теория: Разбираем проект игры “Крестики - нолики”.
Проектная деятельность: Ученики самостоятельно составляют и реализуют алгоритм программы. Возможна помощь товарищей и преподавателя.
63. **Викторина:** повторение ранее изученного материала.
Проектная деятельность: Продолжаем проект “Крестики - нолики”.
Ученики самостоятельно составляют и реализуют алгоритм программы. Возможна помощь товарищей и преподавателя.
64. **Викторина:** повторение ранее изученного материала.
Теория: Разбираем проект игры “Спецназ против Зомби”.
Проектная деятельность: Ученики самостоятельно составляют и реализуют алгоритм программы. Возможна помощь товарищей и преподавателя.
65. **Викторина:** повторение ранее изученного материала.
Проектная деятельность: Продолжаем проект “Спецназ против Зомби”. Ученики самостоятельно составляют и реализуют алгоритм программы. Возможна помощь товарищей и преподавателя.
66. **Викторина:** повторение ранее изученного материала.
Проектная деятельность: Продолжаем проект “Спецназ против Зомби”. Ученики самостоятельно составляют и реализуют алгоритм программы. Возможна помощь товарищей и преподавателя.
67. **Викторина:** повторение ранее изученного материала.
Теория: Разбираем проект игры “Перейти дорогу”.
Проектная деятельность: Ученики самостоятельно составляют и реализуют алгоритм программы. Возможна помощь товарищей и преподавателя.
68. **Викторина:** повторение ранее изученного материала.
Проектная деятельность: Продолжаем проект “Перейти дорогу”.
Ученики самостоятельно составляют и реализуют алгоритм программы. Возможна помощь товарищей и преподавателя.
69. **Викторина:** повторение ранее изученного материала.

Теория: Разбираем проект игры “Сокобан”.

Проектная деятельность: Ученики самостоятельно составляют и реализуют алгоритм программы. Возможна помощь товарищей и преподавателя.

70. **Викторина:** повторение ранее изученного материала.

Проектная деятельность: Разбираем проект “3Д лабиринт”. Ученики самостоятельно составляют и реализуют алгоритм программы. Возможна помощь товарищей и преподавателя.

71. **Викторина:** повторение ранее изученного материала.

Проектная деятельность: Продолжаем проект “3Д лабиринт”. Ученики самостоятельно составляют и реализуют алгоритм программы. Возможна помощь товарищей и преподавателя.

72. **Тест:** Годовое тестирование на усвоение ранее изученного материала.

Теория: Правила поведения во время летних каникул.

Беседа: Рефлексия о проделанной работе за года, что получилось, что не получилось, что понравилось, что не понравилось, что хотелось бы сделать и чему научиться.

Учебный план

№ п/п	Название раздела, темы	Количество часов		
		Всего	Теория	Практика
Раздел «Базовый»				
1	Техника безопасности в компьютерном классе. Ознакомление с курсом "Юный программист Scratch".	2	2	0
2	Основные понятия Scratch.	2	2	0
3	Сообщество Scratch. Регистрация на сайте.	2	0,5	1,5
4	Библиотеки спрайтов и сцен. Создание, сохранение и открытие проектов.	2	0,5	1,5
5	Scratch - графический редактор.	2	0,5	1,5
6	Самые используемые блоки.	2	0,5	1,5
7	Создание диалогов.	2	0,5	1,5
8	Направления. Блоки движения.	2	0,5	1,5
9	Изучаем циклы.	2	0,5	1,5
10	Блоки внешнего вида.	2	0,5	1,5
11	Анимация спрайтов.	2	0,5	1,5
12	Пространство (координаты и направления).	2	0,5	1,5

13	Блоки движения с координатами.	2	0,5	1,5
14	Условия.	2	0,5	1,5
15	Бинарная логика (И/ИЛИ/НЕ).	2	0,5	1,5
16	Основные арифметические операции.	2	0,5	1,5
17	Установка начальных значений.	2	0,5	1,5
18	Сообщения в мультипликации Scratch.	2	0,5	1,5
19	Итоговый проект раздела.	2	0	2
Раздел «Продвинутый»				
20	Понятие алгоритм.	2	0,5	1,5
21	События. Блоки события.	2	0,5	1,5
22	Графические эффекты.	2	0,5	1,5
23	Блоки управления.	2	0,5	1,5
24	Цикл “Повторять пока не”.	2	0,5	1,5
25	Области координат.	2	0,5	1,5
26	Переменные и их виды.	2	0,5	1,5
27	Общие, локальные и облачные переменные.	2	0,5	1,5
28	Ведение расчетов с помощью переменных.	2	0,5	1,5
29	Внутренние переменные Scratch.	2	0,5	1,5
30	Полугодовое тестирование.	2	0,5	1,5
31	Управление состоянием через переменные.	2	0,5	1,5
32	Клоны.	2	0,5	1,5
33	Планирование движения клонов.	2	0,5	1,5
34	Блоки для использования с мышью.	2	0,5	1,5
35	Списки.	2	0,5	1,5
36	Изменение значений списка.	2	0,5	1,5
37	Проход по списку.	2	0,5	1,5
38	Блоки сенсоры.	4	1	3
39	Блоки операторы.	4	1	3
40	Работа со звуком.	2	0,5	1,5
41	Дополнительные блоки "Перо". Проект “Графический редактор”.	2	0,5	1,5
42	Дополнительные блоки "Перо". Проект “Рисуем фигуры”	2	0,5	1,5
43	Дополнительные блоки "Текст в речь”.	2	0,5	1,5
44	Дополнительные блоки "Перевести”.	2	0,5	1,5
45	Дополнительные блоки "Музыка”.	2	0,5	1,5
Раздел «Проектный»				
46	Проект игры “Мемори”.	2	1	1
47	Проект игры “Арканоид”.	4	1	3
48	Проект игры “Динозаврик Гугл”	4	1	3
49	Проект игры “Морской бой”.	4	1	3
50	Проект игры “Флэппи бёрд”.	4	1	3
51	Проект игры “Платформер с несколькими уровнями”.	6	1	5
52	Проект игры “Раннер”.	4	1	3

53	Проект игры “Крестики - нолики”.	4	1	3
54	Проект игры “Спецназ против Зомби”.	6	1	5
55	Проект игры “Перейти дорогу”.	4	1	3
56	Проект игры “Сокобан”.	2	1	1
57	Проект игры “3Д лабиринт”.	4	1	3
58	Годовое тестирование	2	1	1
	ИТОГО:	144	39	105

Календарный учебный график дополнительной общеразвивающей программы «Юный программист» с 01.09.2025 г. по 31.05.2026 г.

Режим деятельности	Дополнительная общеобразовательная общеразвивающая программа «Юный программист»
Начало учебного года	15.09.2025
Окончание учебного года	31.05.2026
Продолжительность учебного периода	36 учебных недель
Периодичность учебных занятий	2 раза в неделю
Количество занятий	72 занятия
Количество часов	144 часа
Комплектование групп	с 01.09.2025 г. по 15.09.2025 г.

Рабочая программа воспитания

Воспитательный компонент осуществляется по следующим направлениям организации воспитания и социализации обучающихся:

- 1) гражданско-патриотическое
- 2) нравственное и духовное воспитание;
- 3) воспитание положительного отношения к труду и творчеству;
- 4) интеллектуальное воспитание;
- 5) здоровьесберегающее воспитание;
- 6) правовое воспитание и культура безопасности;
- 7) воспитание семейных ценностей;
- 8) формирование коммуникативной культуры;
- 9) экологическое воспитание.

Цель – формирование гармоничной личности с широким мировоззренческим кругозором, с серьезным багажом теоретических знаний и практических навыков, посредством информационно-коммуникативных технологий.

Используемые формы воспитательной работы: викторина, экскурсии, игровые программы, диспуты.

Методы: беседа, мини-викторина, моделирование, наблюдения, столкновения взглядов и позиций, проектный, поисковый.

Планируемый результат: повышение мотивации к изобретательству и созданию собственных конструкций; сформированность настойчивости в достижении цели, стремление к получению качественного законченного результата; умение работать в команде; сформированность нравственного, познавательного и коммуникативного потенциалов личности.

Календарный план воспитательной работы

№ п/п	Название мероприятия, события	Направления воспитательной работы	Форма проведения	Сроки проведения
1.	Инструктаж по технике безопасности при работе с компьютером, правила поведения на занятиях.	Безопасность и здоровый образ жизни	В рамках занятий	Сентябрь
2.	Игры на знакомство и командообразование	Нравственное воспитание	В рамках занятий	Сентябрь
3.	Беседа о сохранении материальных ценностей, бережном отношении к оборудованию	Гражданско-патриотическое воспитание, нравственное воспитание	В рамках занятий	Сентябрь, январь
4.	Беседа о празднике «День народного единства»	Нравственное воспитание, трудовое воспитание	В рамках занятий	Октябрь
5.	Участие в соревнованиях и конкурсах, различного уровня	Воспитание интеллектуально познавательных интересов	В рамках занятий	Октябрь - май
6.	Инструктаж на тему “Безопасное поведение в зимний период”	Безопасность и здоровый образ жизни	В рамках занятий	Декабрь
7.	Беседа о празднике «День защитника Отечества»	Гражданско-патриотическое, нравственное и духовное воспитание; воспитание семейных ценностей	В рамках занятий	Февраль
8.	Беседа о «Штурме Кенигсберга»	Гражданско-патриотическое, нравственное и духовное воспитание; воспитание семейных ценностей	В рамках занятий	Апрель
9.	Беседа о празднике «День Победы»	Гражданско-патриотическое, нравственное и духовное воспитание; воспитание семейных ценностей	В рамках занятий	Май
10.	Инструктаж на тему “Безопасное поведение в летний период”	Безопасность и здоровый образ жизни	В рамках занятий	Май

Список использованной литературы:

Нормативные правовые акты:

1. Конституция Российской Федерации. (принята всенародным голосованием 12 декабря 1993 года с изменениями, одобренными в ходе общероссийского голосования 1 июля 2020 года.) [Электронный ресурс] Режим доступа: <http://pravo.gov.ru/constitution/>
2. Конвенция о правах ребенка: [принята резолюцией № 44/25 Генеральной Ассамблеи ООН 20 ноября 1989г., вступила в силу в СССР 15 сентября 1990 г.]. – М.: Омега-Л, 2015. – 19 с. (Законы РФ)
3. Семейный кодекс Российской Федерации. – М.: Эксмо, 2012. – 96 с. – (Актуальное законодательство)
4. Об образовании в Российской Федерации: Федер. закон: [принят Гос. Думой 21 декабря 2012 г., одобрен Советом Федерации 26 декабря 2012 г.]. – М.: Эксмо, 2014. – 208 с. – (Актуальное законодательство)
5. Концепция развития дополнительного образования детей до 2030 г.: [распоряжение Правительством Российской Федерации 31 марта 2022 г. № 678-п]. [Электронный ресурс] // Режим доступа: <http://static.government.ru/media/files/3flgkklAJ2ENBbCFVEkA3cTOsiypicBo.pdf>
6. "Об утверждении санитарных правил СП 2.4. 3648-20 "Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи". [Постановление Главного государственного санитарного врача Российской Федерации от 28.09.2020 № 28] Официальный интернет-портал правовой информации. [Электронный ресурс] // Режим доступа: <http://publication.pravo.gov.ru/Document/View/0001202012210122>
7. Об утверждении порядка организации и осуществления образовательной деятельности по дополнительным образовательным программам: [Приказ Министерства образования и науки Российской Федерации от 27.07.2022 № 629.] Официальный интернет-портал правовой информации. [Электронный ресурс] // Режим доступа: <http://publication.pravo.gov.ru/Document/View/0001202209270013>

Для педагога дополнительного образования:

1. Д. Голиков, "Scratch для юных программистов". Издательство: БХВ-Петербург, 2018, с. 192
2. Й. Алудден, Федерико Вальясинди, Федерика Гамбел, Вивиана Фигас, "Добро пожаловать в увлекательный мир программирования!" — РОСМЭН - 128 с.
3. Э. Свейгарт, "Программирование для детей. Делай игры и учи язык Scratch!", Издательство Эксмо, 2017 - 301 с.
4. А.Г. Кушниренко, Г.В. Лебедев: Основы информатики и вычислительной техники, Издательство Просвещение, 1990

5. К. Бреннан, К. Болкх, М. Чунг, Креативное программирование на языке Scratch, Гарвардская Высшая школа образования, интернет-издание <http://Scratched.gse.harvard.edu/guide/>
6. Программирование для детей. – М.: Манн, Иванов и Фербер, 2015. – 224 с.
7. “Обучение детей основам создания компьютерных игр на языке программирования Scratch” : пособие для учителей учреждений общ. сред. образования с белорус. и рус. яз. обучения : 5-6 классы / О. Е. Елисеева. — Минск : Народная асвета, 2017. — 166 с.

Интернет-ресурсы:

<https://scratch.mit.edu/>
<https://scratch.ru>
<https://code.org>