

**Муниципальный орган Управление образованием – Управление образованием
Тавдинского городского округа**
Муниципальное автономное общеобразовательное учреждение
Основная общеобразовательная школа № 14
(МАОУ ООШ №14)

Согласовано:
Протокол педсовета
МАОУ ООШ № 14
от 31.05.2022 № 19

Утверждено:
Приказом директора
МАОУ ООШ № 14
от 31.05.2022 № 132

**РАБОЧАЯ ПРОГРАММА ОБЪЕДИНЕНИЯ ВНЕУРОЧНОЙ
ДЕЯТЕЛЬНОСТИ**

ТЕХНИЧЕСКОЙ НАПРАВЛЕННОСТИ

«Основы программирования»

Возраст обучающихся: 9 – 12 лет

Срок реализации 1 год

Автор – составитель:
Вдовин Евгений Игоревич

1. Перечень нормативно-правовых документов:

Данная программа разработана с учетом основных законодательных и нормативных актов Российской Федерации и Свердловской области:

1. Федерального закона «Об образовании в Российской Федерации» от 29 декабря 2012 № 273-ФЗ.
2. Концепции развития дополнительного образования детей (утверждена распоряжением Правительства РФ от 04.09.2014 № 1726-р).
3. Приказа Министерства просвещения Российской Федерации от 9 ноября 2018 г. № 196 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам».
4. Постановления Главного государственного санитарного врача РФ от 4 июля 2014 г. № 41 «Об утверждении СанПиН 2.4.4.3172-14 «Санитарно - эпидемиологические требования к устройству, содержанию и организации режима работы образовательных организаций дополнительного образования детей».
5. Письма Минобрнауки России от 18.11.2015 № 09-3242 «О направлении информации».
6. Письма Минобрнауки России от 29.03.2016 № ВК-641/09 «О направлении методических рекомендаций».
7. Приказа Минобрнауки России от 09.01.2014 № 2 «Об утверждении Порядка применения организациями, осуществляющими образовательную деятельность, электронного обучения, дистанционных образовательных технологий при реализации образовательных программ».
8. Приоритетного проекта «Доступное дополнительное образование для детей» (утвержден президиумом Совета при Президенте Российской Федерации по стратегическому развитию и приоритетным проектам протоколом 30.11.2016 № 11).
9. Постановления правительства Свердловской области от 01.08.2019 г. № 461 ПП «О региональном модельном центре дополнительного образования детей Свердловской области».
10. Постановления правительства Свердловской области от 06.08.2019 г. № 503 ПП «О системе персонифицированного финансирования дополнительного образования детей на территории Свердловской области».
11. Приказа Министерства образования и молодежной политики Свердловской области от 30.03.2018 № 162-Д «Об утверждении Концепции развития образования на территории Свердловской области на период до 2035 года»
12. Приказа Министерства образования и молодежной политики Свердловской области от 26.06.2019 № 170-Д «Об утверждении методических рекомендаций «Правила персонифицированного финансирования дополнительного образования детей в Свердловской области».

2. Комплекс основных характеристик

2.1. Пояснительная записка

2.1. Направленность. Техническая.

2.2. Актуальность программы.

Дополнительная общеразвивающая программа «Начала программирования» (далее Программа) реализуется в соответствии с технической направленностью образования.

В последние годы стал популярным язык и одноименная среда программирования - Scratch. Это можно объяснить потребностью и педагогического сообщества, и самих детей в средстве, которое позволит легко и просто, но не бездумно, исследовать и проявить свои творческие способности.

Данная программная среда дает принципиальную возможность составлять сложные по своей структуре программы, не заучивая наизусть ключевые слова, и при этом в полной мере проявить свои творческие способности и понять принципы программирования.

Отличительные особенности среды программирования Scratch это:

- ✓ объектная ориентированность;
- ✓ поддержка событийно-ориентированного программирования; параллельность выполнения скриптов;
- ✓ дружелюбный интерфейс;
- ✓ разумное сочетание абстракции и наглядности; организация текстов программ из элементарных блоков;
- ✓ наличие средств взаимодействия программ на Scratch с реальным миром посредством дополнительного устройства;
- ✓ встроенная библиотека объектов; встроенный графический редактор; активное интернет-сообщество пользователей.

2.3. Отличительные особенности.

Программа модифицированная; за основу взят и переработан ряд программ других педагогических работников школ и дополнительного образования.

Преимущества данной программы перед аналогичными состоит в развитии у обучающихся логического и пространственного мышления.

Можно ли научиться программировать играя? Оказывается, можно. Американские учёные, задумывая новую учебную среду для обучения школьников программированию, стремились к тому, чтобы она была понятна любому ребёнку, умеющему читать.

Название «Scratch» в переводе с английского имеет несколько значений. Это и царапина, которую оставляет Котёнок – символ программы, и каракули, символизирующие первый, ещё неуклюжий самостоятельный опыт, и линия старта. Со Scratch удобно стартовать. Сами разработчики характеризуют программу так: «Scratch предлагает низкий пол (легко начинать), высокий потолок (возможность создавать сложные проекты) и широкие стены (поддержка большого многообразия проектов)».

Подобно тому, как дети только-только начинающие говорить, учатся складывать из отдельных слов фразы, и Scratch обучает из отдельных кирпичиков-команд собирать целые программы.

Scratch приятен «на ощупь». Его блоки, легко соединяемые друг с другом и так же легко, если надо, разбираемые, сделаны явно из пластичных материалов. Они могут многократно растягиваться и снова ужиматься без намёка на изнашиваемость. Scratch зовёт к экспериментам! Важной особенностью этой среды является то, что в ней принципиально невозможно создать неработающую программу.

В Scratch можно сочинять истории, рисовать и оживлять на экране придуманные персонажи, создавать презентации, игры, в том числе и интерактивные, исследовать параметрические зависимости.

Любой персонаж в среде Scratch может выполнять параллельно несколько действий – двигаться, поворачиваться, изменять цвет, форму и т.д.; благодаря чему юные скретчисты учатся осмысливать любое сложное действие как совокупность простых. В результате они не только осваивают базовые концепции программирования (циклы, ветвления, логические операторы, случайные числа, переменные, массивы), которые пригодятся им при изучении более сложных языков, но и знакомятся с полным циклом решения задач, начиная с этапа описания идеи и заканчивая тестированием и отладкой программы.

Scratch легко перекидывает мостик между программированием и другими школьными науками. Так возникают межпредметные проекты. Они помогут сделать наглядными понятия отрицательных чисел и координат, уравнения плоских фигур, изучаемых на уроках математики. В них оживут исторические события и географические карты. А тесты по любым предметам сделают процесс обучения весёлым и азартным.

Scratch хорош как нечто необязательное в школьном курсе, но оттого и более привлекательное, ведь, как известно, именно необязательные вещи делают нашу жизнь столь разнообразной и интересной!

Scratch – свободно распространяемая программа. Она одинаково хорошо устанавливается и в Windows, и в Ubuntu, и в Macintosh.

Scratch создали американцы Митч Резник и Алан Кей. На русский язык программа переведена доцентом Нижегородского университета Евгением Патаракиным.

Обоснование необходимости реализации программы

Курс построен таким образом, чтобы помочь учащимся заинтересоваться программированием вообще и найти ответы на вопросы, с которыми им приходится сталкиваться в повседневной жизни при работе с большим объемом информации; при решении практических и жизненных задач. Программа «Начала программирования» позволяет создавать собственные программы для решения конкретной задачи. Это является отличительной особенностью данной программы.

Актуальность проектной деятельности сегодня осознаётся всеми. ФГОС нового поколения требует использования в образовательном процессе технологий деятельного типа, методы проектно-исследовательской деятельности определены как одно из условий реализации основной образовательной программы образования.

Следует иметь в виду, что возрастные особенности школьника среднего возраста не позволяют в полной мере реализовать проведение полноценных научных исследований. Раннее включение в организованную специальным образом проектную деятельность творческого характера позволяет сформировать у школьника познавательный интерес и исследовательские навыки. В будущем они станут основой для организации научно-исследовательской деятельности в вузах, колледжах, техникумах и т.д.

Организация научно-познавательной деятельности школьника требует использования инструмента (средства) для выполнения как исследовательских, так и творческих проектов. В качестве такого инструмента я вижу среду программирования Scratch.

Предлагаемый курс является отличной средой для проектной деятельности. В ней есть все необходимое:

- ✓ графический редактор для создания и модификации визуальных объектов;
- ✓ библиотека готовых графических объектов (некоторые из них содержат наборы скриптов);
- ✓ библиотека звуков и музыкальных фрагментов;
- ✓ большое количество примеров.

Scratch является отличным инструментом для организации научно-познавательной деятельности школьника благодаря нескольким факторам:

- ✓ эта программная среда легка в освоении и понятна школьникам, но при этом - она позволяет составлять сложные программы;
- ✓ эта программа позволяет заниматься и программированием, и созданием творческих проектов;
- ✓ вокруг Scratch сложилось активное, творческое международное сообщество.

Язык Scratch особенно интересен для начального уровня изучения программирования. Обучение основам программирования в этой среде наиболее эффективно при выполнении небольших (поначалу) проектов. При этом естественным образом ученик овладевает интерфейсом новой для него среды, постепенно углубляясь как в возможности Scratch, так и в идеи собственно программирования. Базовый проект един для всех учеников и выполняется совместно с учителем. Затем предлагаются возможные направления развития базового проекта, которые у разных учеников могут быть различными.

При создании сложных проектов ученик не просто освоит азы программирования, но и познакомится с полным циклом разработки программы, начиная с этапа описания идеи и заканчивая тестированием и отладкой.

Scratch не просто среда для программирования, через нее можно выйти на многие другие темы школьной информатики. Создавая свои собственные игры и мультфильмы, дети научатся разрабатывать проекты, ставить цели и задачи. Чтобы оформить это, нужно поработать в текстовом редакторе. Потом надо нарисовать героя, окружение. Разработать алгоритм действий героя, алгоритмы его реакций на события. Надо будет озвучить героя и события (записать, обработать звук). Важно и то, что ребенок имеет возможность поделиться результатами своего творчества с друзьями или другими пользователями: кнопка для размещения созданного проекта в Сети находится непосредственно в программе.

2.4. Адресат программы и возрастные особенности.

Программа рассчитана на детей среднего школьного возраста.

Сегодня, в начале XXI в. наблюдается процесс дицелерации (замедление темпов развития). Современные дети по всей совокупности морфологических характеристик значительно уступают своим родителям в их детстве, и этот процесс, исходя из предположений антропологов, по всей вероятности будет продолжаться. Вместе с тем в связи со стремительными темпами научно-

технического прогресса наши дети более информированы и эрудированы. Существуют ли специфические, характерные для сегодняшних школьников особенности в развитии их интеллектуальных способностей, и каковы эти особенности? Ответ на этот вопрос принципиально важен как для современной науки, так и для практики организации учебно-образовательного процесса. Исходя из вышесказанного, можно констатировать, что изучение закономерностей развития познавательных способностей и их формирования в учебной деятельности в период роста и развития человека является одной из актуальных теоретических и научно-практических задач современности.

Организация учебной деятельности подростков – важнейшая и сложнейшая задача. Ученик среднего школьного возраста вполне способен понять аргументацию педагога, родителя, согласиться с разумными доводами. Однако в виду особенностей мышления, характерных для данного возраста, подростка уже не удовлетворит процесс сообщения сведений в готовом, законченном виде. Ему захочется проверить их достоверность, убедиться в правильности суждений. Споры с учителями, родителями, приятелями – характерная черта данного возраста. Их важная роль заключается в том, что они позволяют обменяться мнениями по теме, проверить истинность своих воззрений и общепринятых взглядов, проявить себя. В частности, в обучении большой эффект дает внедрение проблемных задач.

В этот период происходит дальнейшее физическое и психофизиологическое развитие ребенка, обеспечивающее возможность систематического обучения в школе. Прежде всего, совершенствуется работа головного мозга и нервной системы. По данным физиологов, к 9 годам кора больших полушарий является уже в значительной степени зрелой. Однако наиболее важные, специфически человеческие отделы головного мозга, отвечающие за программирование, регуляцию и контроль сложных форм психической деятельности, у детей этого возраста еще не завершили своего формирования (развитие лобных отделов мозга заканчивается лишь к 12 годам), вследствие чего регулирующее и тормозящее влияние коры на подкорковые структуры оказывается недостаточным. Несовершенство регулирующей функции коры проявляется в свойственных детям данного возраста особенностях поведения, организации деятельности и эмоциональной сферы: младшие школьники легко отвлекаются, не способны к длительному сосредоточению, возбудимы, эмоциональны.

Средний школьный возраст является периодом интенсивного развития и качественного преобразования познавательных процессов: они начинают приобретать опосредствованный характер и становятся осознанными и произвольными. Ребенок постепенно овладевает своими психическими процессами, учится управлять восприятием, вниманием, памятью.

Важным условием для формирования теоретического мышления в среднем школьном возрасте является формирование **научных** понятий. Теоретическое мышление позволяет ученику решать задачи, ориентируясь не на внешние, наглядные признаки и связи объектов, а на внутренние, существенные свойства и отношения.

Занятия по Scratch программированию развивают логику, повышают системность мышления, а также развивают творческие способности. Все это так же влияет на степень осознанности в принимаемых решениях. Даже, если ребенок не станет программистом, то понимание, как составляются программы обязательно пригодятся в другой деятельности, какую бы профессию ребенок не выбрал в будущем.

2.5. Режим занятий, периодичность и продолжительность занятий.

Программа рассчитана на детей в возрасте от 9 до 12 лет. Группа формируется из 10-ти человек - по количеству рабочих мест (компьютеров).

Объем программы составляет 34 учебных часа.

Срок реализации программы составляет 34 учебных недель в период времени с 1 сентября по 31 мая. Запланированный срок реален и достаточен для достижения цели и ожидаемых результатов.

Режим занятий: 1 раз в неделю по 45 минут (часов), что соответствует требованиям санитарно-эпидемиологического законодательства.

2.6. Объем и срок освоения программы.

Общее количество учебных часов, запланированных на весь период обучения и необходимых для освоения программы – 34 часа.

Срок освоения программы - 1 год.

2.7. Уровневость освоения программы.

Реализация программы предполагает освоение стартового уровня.

2.8. Формы обучения.

Формы занятий определяются количеством детей и особенностями материала. Занятия проводятся по группам в аудитории. Занятий для самостоятельного изучения материала вне аудитории не предусмотрено.

Форма занятий — по количеству детей, фронтальная, при которой, работа преподавателя ведется сразу со всей группой в едином темпе и с общими задачами. При этом выделяется время на индивидуальную работу с каждым обучающимся через определенные промежутки времени (индивидуальная форма обучения). На занятиях применяются групповые формы обучения и практикумы.

2.9. Виды занятий.

Урок-консультация, урок-практическая работа, уроки с групповыми формами работы, уроки-конкурсы.

2.10. Формы подведения результатов.

Для отслеживания результатов обучения по программе используется метод педагогического наблюдения, беседа с учащимися, педагогический анализ проводимых отчетных мероприятий.

2.2. Цель и задачи общеразвивающей программы

Цель – основной целью программы является обучение программированию через создание творческих проектов, развитие личности ребенка, способного к творческому самовыражению, обладающего технической культурой, аналитическим мышлением, навыками и умениями робототехники и программирования, умеющего работать в коллективе, способного применять полученные знания при решении бытовых и учебных задач.

Задачи:

Обучающие задачи

- ✓ овладеть навыками составления алгоритмов;
- ✓ овладеть понятиями «объект», «событие», «управление», «обработка событий»;
- ✓ изучить функциональность работы основных алгоритмических конструкций;
- ✓ сформировать представление о профессии «программист»;
- ✓ сформировать навыки разработки, тестирования и отладки несложных программ;
- ✓ познакомить с понятием проекта и алгоритмом его разработки;
- ✓ сформировать навыки разработки проектов: интерактивных историй, интерактивных игр, мультфильмов, интерактивных презентаций.

Развивающие задачи

- ✓ способствовать развитию критического, системного, алгоритмического и творческого мышления;
- ✓ развивать внимание, память, наблюдательность; познавательный интерес;
- ✓ развивать умение работать с компьютерными программами и дополнительными источниками информации;
- ✓ развивать навыки планирования проекта, умение работать в группе.

Воспитательные задачи

- ✓ формировать положительное отношение к информатике и ИКТ;
- ✓ развивать самостоятельность и формировать умение работать в паре, малой группе, коллективе;
- ✓ формировать умение демонстрировать результаты своей работы.

2.3. Содержание общеразвивающей программы

Учебно-тематический план

№ п/п	Наименование раздела и темы	Количество часов			Форма контроля
		Всего	Теория	Практика	
1	Введение	1	0,5	0,5	
1.1	Инструктаж по ТБ. Знакомство со средой программирования Scratch.	1	0,5	0,5	Устный опрос. Практическое задание
2	Среда программирования Scratch	33	3,25	29,75	
2.1	Исполнитель Scratch, цвет и размер пера.	1	0,25	0,75	Устный опрос, практическое задание
2.2	Основные инструменты встроенного графического редактора программной среды SCRATCH.	1	0,25	0,75	Устный опрос, практическое задание
2.3	Алгоритм. Линейный алгоритм. Создание блок-схемы. Основные графические примитивы векторного редактора LibreOfficeDraw.	1	0,25	0,75	Устный опрос, практическое задание
2.4	Линейный алгоритм. Рисование линий исполнителем Scratch.	1		1	Практическое задание
2.5	Линейный алгоритм. Исполнитель Scratch рисует квадраты и прямоугольники линейно.	1		1	Практическое задание
2.6	Конечный цикл. Scratch рисует квадраты, линии.	1	0,25	0,75	Устный опрос, практическое задание
2.7	Конечный цикл. Scratch рисует несколько линий и фигур. Копирование фрагментов программы.	1		1	Практическое задание
2.8	Циклический алгоритм. Цикл в цикле.	1	0,25	0,75	Устный опрос, практическое задание
2.9	Цикл в цикле. Повторение пунктирной линии с поворотом. Блок-схема цикла.	1		1	Практическое задание
2.10	Бесконечный цикл. Анимация исполнителя Scratch на основе готовых костюмов.	1		1	Практическое задание
2.11	Сцена как исполнитель. Создаем модель таймера.	1		1	Практическое задание
2.12	Одинаковые программы для нескольких исполнителей.	1		1	Практическое задание
2.13	Параллельное выполнение действий несколькими исполнителями	1		1	Практическое задание
2.14	Разбиение программы на части для параллельного выполнения исполнителями. Таймер.	1		1	Практическое задание
2.15	Два исполнителя со своими программами. Мини-проект «Часы с кукушкой».	1		1	Практическое задание
2.16	Алгоритмы с ветвлением. Условие ЕСЛИ.	1	0,25	0,75	Устный опрос, практическое задание
2.17	Цикл с условием. Мини-проект «Шарики в лабиринте»	1		1	Практическое задание
2.18	Цикл с условием. Исполнитель определяет цвет. Сенсор «касается цвета»	1		1	Практическое задание
2.19	Оператор случайных чисел.	1	0,25	0,75	Устный опрос, практическое задание

2.20	Перемещение исполнителей между слоями.	1	0,25	0,75	Устный опрос, практическое задание
2.21	Действия исполнителей в разных слоях.	1		1	Практическое задание
2.22	Взаимодействие исполнителей.	1	0,25	0,75	Устный опрос, практическое задание
2.23	Последовательное выполнение команд исполнителями.	1	0,25	0,75	Устный опрос, практическое задание
2.24	Программирование клавиш. Мини-проект «Лабиринт»	1	0,25	0,75	Устный опрос, практическое задание
2.25	Управление событиями.	1		1	Практическое задание
2.26	Координатная плоскость. Геометрические фигуры.	1		1	Практическое задание
2.27	Координатная плоскость. Переменные.	1		1	Практическое задание
2.28	Создание списков.	1	0,25	0,75	Устный опрос, практическое задание
2.29	Использование подпрограмм.	1		1	Практическое задание
2.30	Сообщество Scratch.	1	0,25	0,75	Устный опрос, практическое задание
2.31	Отладка программ с ошибками.	1		1	Практическое задание
2.32	Итоговый проект.	2		2	Практическое задание, презентация
Всего:		34	3,75	30,25	

Содержание учебного (тематического) плана

1. Введение

1.1 Теория: обзор программного обеспечения "Scratch", знакомство с компьютером, знакомство с правилами поведения и техники безопасности в рамках компьютерного класса. Основные элементы пользовательского интерфейса программной среды Scratch. Внешний вид рабочего окна. Блочная структура систематизации информации. Функциональные блоки. Блоки команд, состояний, программ, запуска, действий и исполнителей. Установка русского языка для Scratch. Создание и сохранение документа. Понятия спрайта, сцены, скрипта. Очистка экрана.

Библиотека персонажей. Сцена и разнообразие сцен, исходя из библиотеки данных. Систематизация данных библиотек персонажей и сцен. Иерархия в организации хранения костюмов персонажа и фонов для сцен. Импорт костюма, импорт фона.

Практика: Кейс 1. Научи кота бегать и мяукать.

2. Среда программирования Scratch

2.1 Теория: Понятие ИСПОЛНИТЕЛЯ.

Команды управления пером: «Опустить перо», «Поднять перо», «Очистить», «Установить цвет пера», «Установить размер пера».

Практика: Кейс 2. Рисуем разноцветные лужи и облака для прогулки кота.

2.2 Теория: Инструменты растрового графического редактора — кисточка, ластик, заливка, квадрат, круг, линия.

Копирование, поворот, горизонтальное отражение, вертикальное отражение во встроенном редакторе программной среды Scratch.

Командный блок внешность (фиолетовый) — команды начальной установки эффекта цвет «Установить эффект цвет в значение 0 и команда начальной установки размера «Установить размер 100%».

Команды: «Изменить цвет эффект на _», «Изменить размер на _»

Командный блок управления пером (зелёный) — команда «Печать» для копирования графического изображения исполнителя в нужном месте экрана.

Эффекты, которые могут быть применены к графическим изображениям действующего исполнителя.

В графическом редакторе можно создавать сложные графические изображения, почти не прикладывая собственных усилий, а лишь правильно применяя встроенные возможности программной среды.

Практика: Кейс 3. Свободное рисование

2.3 Теория: Создание собственных изображений в других программах (например, LibreOfficeDraw) и импортирование их в программную среду Scratch.

Знакомство с основными графическими примитивами векторного редактора LibreOfficeDraw. Возможность создания геометрических фигур без внутренней заливки, но с текстовым блоком внутри. Стрелки, их направление.

Алгоритм, блок-схема как способ записи.

Практика: Кейс 4. Запиши мой алгоритм!

Ребята разбиваются на пары, в паре формулируют друг другу какую-то повседневную задачу, для решения которой нужен линейный алгоритм, придумывают этот алгоритм и рисуют блок-схему для задачи, заданной им напарником.

2.4 Практика: Кейс 5. Что бывает полосатое?

Придумать линейный алгоритм для создания полосатого объекта (заранее придумайте, что это будет - шарф, зебра, забор?) Создайте блок-схему этого алгоритма и реализуйте алгоритм в среде Scratch.

Решение поставленной задачи в виде последовательного выполнения команд. Создание блок-схемы линейного алгоритма средствами редактора векторной графики. Последовательное выполнение команд. Изменение параметров пера.

2.5 Теория: Создание алгоритма для рисования исполнителем квадрата путем последовательного выполнения команд. Центр костюма исполнителя Scratch.

Практика: Рисование линейного алгоритма, состоящего из двух колонок блоков команд.

Использование векторного редактора офисного пакета LibreOffice в качестве инструмента для создания блок-схем. Выбор нужного значения из предлагаемого списка вариантов. Отладка программы для получения верного результата. Команда «повернуть в направление».

Пошаговое выполнение программы для её отладки.

2.6 Практика: Упражнение.

Сравнение двух алгоритмов, рисующих квадрат. Идти 100 шагов Повторить 4

Повернуться на 90° Идти 100 шагов

Идти 100 шагов Повернуться на 90°

Повернуться на 90°

Идти 100 шагов Повернуться на 90°

Идти 100 шагов Повернуться на 90°

Рекомендуется объяснить, что команды, обеспечивающие повторяющееся выполнение одной или нескольких команд называются циклическими, показать, что использование команды цикла всегда сокращает количество команд в программе, а значит делает её более красивой. В более короткой программе легче обнаружить ошибку или неточность.

Сохранение готовых программ для дальнейшего использования.

Рисование блок-схемы циклического алгоритма. Использование векторного редактора офисного пакета LibreOffice в качестве инструмента для создания блок-схем; Использование команд поворота на прямой угол (90°) по часовой и против часовой стрелки;

Использование циклического алгоритма для рисования исполнителем квадрата;

Оптимизация линейного алгоритма за счёт использования циклической конструкции в

программной среде Scratch; Команда открыть... из пункта меню File;

Команда сохранить как... из пункта меню File;

Циклический алгоритм;

Блок-схема циклического алгоритма;

Команды: «повернуться на 90° по часовой стрелке», «повернуться на 90° против часовой стрелки», «повторить _».

2.7 Теория: Использование операции цикла для решения учебных задач. Применение поворота на прямой угол (90°) при создании геометрических фигур и перемещении исполнителя. Оптимизация линейного алгоритма за счёт использования циклической конструкции в программной среде Scratch.

Тело цикла.

Конечный и бесконечный циклы.

Блок-схема бесконечного цикла.

Имя спрайта и костюма.

Изменение костюма исполнителя.

Копирование фрагмента программы.

Команды: «Следующий костюм», «Перейти к костюму». Рекомендуется объяснить, что циклические конструкции могут применяться не только для рисования фигур. Например, в цикле можно менять внешность исполнителя.

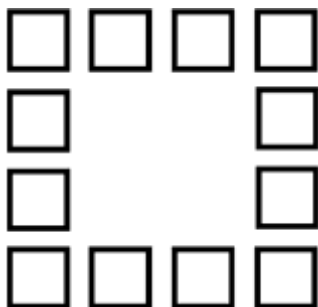
Практика: Кейс 6. Создай картинку из квадратов.

Предложите ребятам нарисовать свою картинку из квадратов, может быть, это будет лестница, может быть, ковер, может быть, что-то еще.

2.8 Практика: Упражнение.

Использование циклов «повторить _»

Нарисуйте квадраты, как показано на рисунке, используя конструкцию цикл в цикле.



Использование операции цикла в цикле для решения учебных задач.

Использование поворота на прямой угол (90°) по часовой и против часовой стрелки.

Оптимизация алгоритма за счёт использования конструкции «цикл в цикле».

Блок-схема конструкции цикл в цикле.

Например, расписание уроков составляется на неделю, в месяце в среднем четыре недели. Если расписание не меняется в течение учебного года, то получаем цикл из девяти месяцев по четыре недели с одинаковым расписанием.

2.9 Практика: Кейс 7. Мой необычный дом.

Предложите ребятам нарисовать дом и что-то рядом с ним, используя линейные и циклические алгоритмы, а также смену костюмов Исполнителя. Может быть, над домом будут звезды разного размера, а около дома будет пунктирная разноцветная дорога, или деревья разного размера.

Использование конструкции «цикл в цикле».

Создание и реализация алгоритма рисования квадрата не сплошными линиями;

Оптимизация алгоритма за счёт использования конструкции «цикл в цикле» в программной среде Scratch;

Использование операции копирования внешности исполнителя путём копирования костюма.

Не сплошные линии.

Алгоритм рисования не сплошных линий с использованием циклических конструкций.

Рисование квадрата не сплошными линиями, используя конструкцию «цикл в цикле».

При перемещении исполнителя можно рисовать линии отличные от сплошной. Эти возможности определяются использованием команд из блока ПЕРО.

2.10 Практика: Упражнение с использованием бесконечного цикла.

Анимация исполнителя с помощью смены костюмов.

Эффект «призрак».

Изменение размера исполнителя.

Интерактивное взаимодействие с исполнителем с помощью клавиатуры.

Команды «спросить» и «думать».

2.11 Практика: Упражнение с анимацией сцены (фона).

Синхронная анимация сцены и исполнителя.

2.12 Практика: Упражнение с анимацией с помощью вращения.

Дублирование исполнителей.

При копировании исполнителей копируются и их скрипты. Использование одинаковых программ, но разных костюмов у исполнителей.

2.13 Практика: Упражнение «Дублирование исполнителей».

Синхронное выполнение скриптов.

При параллельном выполнении скриптов результат работы всех исполнителей получается гораздо быстрее, чем в тех задачах, когда работал один исполнитель.

2.14 Практика: Кейс 8. Мини-проект «Смена времени суток».

Создайте скрипт, в котором будет четыре сцены: «утро», «день», «вечер», «ночь». Используйте «таймер» для своевременного изменения сцен, своевременного запуска скриптов разных исполнителей.

Планирование действий разных исполнителей во времени для решения общей задачи.

Сенсор «таймер»

Сброс «таймера»

Команда «ждать до...»

2.15 Практика: Кейс 9. Мини-проект «Часы с кукушкой».

Создайте сцену в виде часов, и три исполнителя: часовую стрелку, минутную стрелку и кукушку. Стрелки должны вращаться на циферблате. При наступлении каждого часа (т.е. в _ часов: 00 минут) должна появляться кукушка.

Координаты в Scratch. Команда «идти в х: _ у: _» (переместиться в точку с координатами (х,у))

Команда «плыть _ секунд в точку х: _ у: _» (плавное движение в точку с координатами (х,у))

Команды «спрятаться», «показаться».

2.16 Теория:

Еще из сказок мы знаем, что на камне было написано: «Налево пойдешь - коня потеряешь, направо пойдешь - себя не найдешь, прямо пойдешь - в сказку попадешь»

Блок-схема алгоритма с ветвлением.

Полная форма ветвления. Команда «если _ или»

Неполная форма ветвления. Команда «если _»

Сенсор «касается» (касания края, или другого спрайта, или указателя мыши)

Практика: Упражнение с блок-схемой алгоритма с ветвлением.

2.17 Практика: Кейс 10. Мини-проект «Шарики в лабиринте»

Создайте сцену в виде бильярдного стола, на котором три шарика будут двигаться с разными скоростями и отталкиваться от бортиков.

Сенсор «касается цвета»

Программируем отскок шариков от стен

2.18 Практика: Упражнение «Программируем поведение исполнителя в зависимости от цвета фона».

2.19 Теория: Команда «Выдать случайное число от _ до _». Случайные перемещения исполнителя в координатной плоскости.

Случайное количество шагов.

Случайные координаты.

Поворот на случайный угол.

Практика: Упражнение с использованием команды случайных чисел.

2.20 Теория: Команда «Перейти в верхний слой»

Команда «Перейти назад на _ слоев»

Эффект «Призрак»

Практика: Упражнение с использованием переходов между слоями.

2.21 Практика: Кейс 10. Мини-проект «Дорога».

Создайте сцену, на которой изображена дорога с двумя полосами, по обочинам дороги расположите кусты и деревья. По дороге должны двигаться автомобили. Правильно расположите автомобили, деревья и кусты по слоям.

2.22 Теория: Исполнители касаются друг друга.

Команда «Касается _»

Практика: Упражнение «Поведение исполнителей при столкновении».

2.23 Теория: Команда «Передать»

Команда «Когда я получу»

Практика: Упражнение «Связи между программами разных исполнителей».

2.24 Теория: Взаимодействие пользователя с программой.

Клавиши управления перемещением исполнителя Команда «Изменить значение x на _»

Команда «Изменить значение y на _»

Практика: Кейс 11. Игра «Лабиринт»

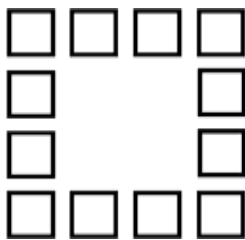
Создайте сцену в виде лабиринта. С помощью клавиш со стрелками управляйте движением маленькой мышки. Мышка должна добраться до выхода, не задевая стенок лабиринта. В случае касания стенок мышка возвращается в исходную точку.

2.25 Практика: Передача сообщений между исполнителями и фоном.

Запуск программ после получения сообщения.

2.26 Практика: Кейс 12. Исследование времени выполнения программ.

Создайте одного исполнителя, рисующего квадраты, как показано на рисунке. Затем создайте четыре исполнителя, которые будут параллельно рисовать по три квадрата. Сравните время работы в этих двух случаях.



Рисование геометрических фигур на координатной плоскости. Последовательное выполнение команд одним исполнителем. Параллельное выполнение команд несколькими исполнителями.

2.27 Практика: Рисование геометрических фигур на координатной плоскости с использованием переменных.

Построение перпендикуляров к координатным осям. Блок «Переменные»

2.28 Теория: Название списка.

Элементы списка.

Длина списка.

Команда «Создать список».

Выбор элемента списка.

Практика: Кейс 13. Мини-проект «Викторина».

Создайте проект «Викторина» в котором будет задано 10 вопросов. Правильный ответ надо выбрать из списка. В случае правильного ответа очки увеличиваются на единицу. В конце выставляется оценка: «отлично» за 9 или 10, «хорошо» за 7 или 8, «удовлетворительно» за 5 или 6 правильных ответов.

2.29 Практика: Упражнение «Как сделать программу структурированной и более понятной».

Команда «Передать _ и ждать»

Команда «Играть звук _»

Команда «Ноту _ играть _ тактов»

2.30 Теория: Scratch 2.0. Регистрация на сайте. Личный кабинет.

Практика: Публикация проектов.

2.31 Практика: Отладка программ с ошибками.

<http://scratch.mit.edu/projects/10437040>

<http://scratch.mit.edu/projects/10437249>

<http://scratch.mit.edu/projects/10437366>

<http://scratch.mit.edu/projects/10437439>

<http://scratch.mit.edu/projects/10437476>

2.32-2.33 Практика: Кейс 14. Итоговый проект.

Учащиеся самостоятельно или в парах выполняют индивидуальный проект, согласованный с учителем. Как итог, проводится презентация проектов.

2.4. Планируемые результаты

В результате изучения курса получают дальнейшее развитие метапредметные, личностные, регулятивные, коммуникативные и познавательные универсальные учебные действия, учебная (общая и предметная) и общепользовательская ИКТ-компетентность обучающихся.

Метапредметные результаты:

- умение самостоятельно определять цели своего обучения, ставить и формулировать для себя новые задачи в учёбе и познавательной деятельности, развивать мотивы и интересы своей познавательной деятельности;
- умение самостоятельно планировать пути достижения целей, в том числе альтернативные, осознанно выбирать наиболее эффективные способы решения учебных и познавательных задач;
- умение соотносить свои действия с планируемыми результатами, осуществлять контроль своей деятельности в процессе достижения результата, определять способы действий в рамках предложенных условий и требований, корректировать свои действия в соответствии с изменяющейся ситуацией;
- умение оценивать правильность выполнения учебной задачи, собственные возможности её решения;
- владение основами самоконтроля, самооценки, принятия решений и осуществления осознанного выбора в учебной и познавательной деятельности;
- умение определять понятия, создавать обобщения, устанавливать аналогии, классифицировать, самостоятельно выбирать основания и критерии для классификации, устанавливать причинно-следственные связи, строить логическое рассуждение, умозаключение (индуктивное, дедуктивное и по аналогии) и делать выводы;
- умение создавать, применять и преобразовывать знаки и символы, модели и схемы для решения учебных и познавательных задач;
- умение организовывать учебное сотрудничество и совместную деятельность с учителем и сверстниками; работать индивидуально и в группе: находить общее решение и разрешать конфликты на основе согласования позиций и учёта интересов; формулировать, аргументировать и отстаивать своё мнение;
- формирование и развитие компетентности в области использования информационно-коммуникационных технологий.

Личностные результаты:

- формирование ответственного отношения к учению, готовности и способности обучающихся к саморазвитию и самообразованию на основе мотивации к обучению и познанию, осознанному выбору и построению дальнейшей индивидуальной траектории образования на базе ориентировки в мире профессий и профессиональных предпочтений, с учётом устойчивых познавательных интересов;
- формирование целостного мировоззрения, соответствующего современному уровню развития науки и общественной практики, учитывающего социальное, культурное, языковое, духовное многообразие современного мира;
- формирование коммуникативной компетентности в общении и сотрудничестве со сверстниками, детьми старшего и младшего возраста, взрослыми в процессе образовательной, общественно полезной, учебно-исследовательской, творческой и других видов деятельности.

Предметные результаты:

- формирование представления об основных изучаемых понятиях курса;
- формирование информационной и алгоритмической культуры; формирование представления о компьютере как универсальном устройстве обработки информации; развитие основных навыков и умений использования компьютерных устройств;
- развитие алгоритмического мышления, необходимого для профессиональной деятельности в современном обществе; развитие умений составить и записать алгоритм для решения

конкретной задачи;

- формирование умений формализации и структурирования информации, умения выбирать способ представления данных в соответствии с поставленной задачей, с использованием соответствующих программных средств обработки данных;
- формирование представления о том, что значит “программировать” на примере языка Scratch, формирование умения составлять сценарии проектов среды Scratch;
- знакомство с основными алгоритмическими структурами — линейной, условной и циклической;
- формирование умения тестировать и оптимизировать алгоритмы исполнителей;
- формирование умения формализации и структурирования информации;
- использование готовых прикладных компьютерных программ по выбранной специализации;
- развитие умений применять изученные понятия, результаты, методы для решения задач практического характера и задач из смежных дисциплин с использованием при необходимости справочных материалов, компьютера;
- формирование навыков и умений безопасного и целесообразного поведения при работе с компьютерными программами и в Интернете, умения соблюдать нормы информационной этики и права.

3. Комплекс организационно-педагогических условий

3.1. Условия реализации программы

Материально-техническое обеспечение.

Для реализации настоящей программы требуется:

- наличие компьютерного класса, с оборудованием, соответствующим санитарным нормам;
- 10 ученических мест;
- каждое учебное место должно быть оборудовано 1 компьютером с установленным программным обеспечением, соответствующим следующим характеристикам:
- ПО Scratch
- принтер, сканер;
- маркерная доска;
- видеопроектор.

Информационное обеспечение.

Аудио-, видео-, фото-, интернет источники.

1. <https://scratch.mit.edu/> – web сайт Scratch
2. <http://robot.edu54.ru/> - Портал «Образовательная робототехника»
3. <http://www.robogeek.ru/> - РобоГик, сайт, посвященный робототехнике
4. <http://wroboto.ru/> - Сайт, посвященный международным состязаниям роботов
5. <http://ligarobotov.ru/> - сайт проекта «Лига роботов»

Кадровое обеспечение.

Преподаватель математики и информатики Семенова Т. П. - учитель первой категории.

Методические материалы.

Методическое обеспечение – это методы и технологии, используемые для реализации образовательного процесса.

В процессе реализации программы используются следующие методы организации занятий:

- ✓ словесные методы (лекция, объяснение);
- ✓ демонстративно-наглядные (демонстрация работы в программе, схем, скриптов, таблиц);
- ✓ исследовательские методы;
- ✓ работа в парах;
- ✓ работа в малых группах;
- ✓ проектные методы (разработка проекта по спирали творчества, моделирование, планирование деятельности);
- ✓ работа с Интернет-сообществом (публикация проектов в Интернет-сообществе скретчеров).

Практическая часть работы – работа в среде программирования со скриптами и проектирование

информационных продуктов. Для наилучшего усвоения материала практические задания рекомендуется выполнять каждому за компьютером. При выполнении глобальных проектов рекомендуется объединять школьников в пары. Текущий контроль уровня усвоения материала осуществляется по результатам выполнения рефлексивных упражнений и практических заданий. Итоговый контроль осуществляется по результатам разработки проектов. Формы подведения итогов: презентация проекта, испытание квеста, игры.

Для успешной реализации программы используются следующие методические материалы:

- ✓ учебно-тематический план;
- ✓ календарно-тематический план;
- ✓ теоретический материал по изучаемым темам;
- ✓ инструкции по технике безопасности и правилам поведения в учреждении;
- ✓ справочники и переводчики в электронном виде;
- ✓ методическая литература для педагогов дополнительного образования.

Дидактическое обеспечение программы включает в себя следующие материалы:

- ✓ учебные презентации по темам;
- ✓ материалы для практических и самостоятельных заданий;
- ✓ материалы для проведения конкурсных мероприятий.

3.2. Формы аттестации/контроля и оценочные материалы.

Форма аттестации обучающихся по данной программе итоговая проектная работа.

Для отслеживания результатов обучения по программе используется метод педагогического наблюдения, беседа с учащимися, педагогический анализ проводимых отчетных мероприятий.

Мониторинг освоения Общеобразовательной (общеразвивающей) программы дополнительного образования.

Цель: отслеживание динамики развития личностных качеств и исполнительских навыков обучающихся.

Сроки проведения:

Входной контроль	Дата проведения: Сентябрь
Промежуточный контроль	Дата проведения: Декабрь
Итоговый контроль	Дата проведения: Май

С целью проверки эффективности развития личностных качеств и исполнительских навыков обучающихся были разработаны ее критерии и уровни.

- Универсальные учебные действия;
- Подготовка по предмету.

Критерию «**универсальные учебные действия**» соответствуют следующие показатели:

- Мотивация, активная позиция обучающегося - интерес и потребность к данному виду деятельности, активность самоорганизации и стремление к занятиям, проявляемая активность при достижении целей, эмоциональное участие в процессе обучения, умение устанавливать личностный смысл деятельности, мотивировать ее внутренней или внешней необходимостью.
- Умение работать в команде - наличие коммуникативных навыков как фактора социализации обучающихся, создания благоприятного климата в детском коллективе для более легкого и успешного освоения программы.
- Умение самостоятельно находить способы решения поставленной задачи – осознание обучающимися уровня освоения планируемого результата деятельности, приводящее к пониманию своих проблем и тем самым созданию предпосылок для дальнейшего самосовершенствования.

Критерию «**Подготовка по предмету**» соответствуют следующие показатели:

- Знание комплекса артикуляционной гимнастики – уверенное выполнение 6-7 необходимых упражнений.
- Умение самостоятельно придумать и показать этюд на заданную тему – умение создать верное поведение на предложенные обстоятельства.

- Знание комплекса упражнений для развития гибкости и подвижности - уверенное выполнение необходимых упражнений.

- Знание основных эмоций - умение быстро переключаться с одной эмоции на другую.

Для проведения мониторинга определены три уровня развития определенных качеств: высокий, средний, низкий.

Высокому уровню (4-5 баллов) соответствуют: высокое и четкое проявление параметра, хорошо сформированный навык, глубокое, устойчивое знание предмета;

Средний уровень развития (2-3 балла) характеризуется: среднее проявление параметра, навык сформирован, присутствуют знания на среднем уровне, результат не стабильный;

Начальный уровень развития (0-1 балл): исследуемый параметр не развит, не выражен или проявляется на низком уровне, редко, навык не сформирован.

В ходе проведения мониторинга применялись следующие методы:

- наблюдение,
- опрос,
- беседа,
- диагностика,
- обобщение педагогического опыта,
- опытная работа.

Основной целью данной программы является формирование и развитие у детей навыков и умений конструирования и начального программирования, способность применять полученные знания при решении бытовых и учебных задач.

Успехи, достигнутые учащимися, демонстрируются во время проведения творческих мероприятий и оцениваются соучениками, родителями и педагогами.

Для этого используются такие формы:

- открытые занятия;
- обобщающие занятия;
- защита проектов.

4. Список литературы

Список литературы, используемой педагогом

1. Авторская программа курса по выбору «Творческие задания в среде программирования Scratch» Ю.В.Пашковской 5-6 классы, которая входит в сборник «Информатика. Программы для образовательных организаций: 2-11 классы» / составитель М.Н. Бородин. – М.: БИНОМ. Лаборатория знаний, 2015.
2. Иллюстрированное руководство по языкам Scratch и Python «Программирование для детей»/К. Вордерман, Дж.Вудкок, Ш.Макаманус и др.; пер. с англ.С.Ломакин. – М.:Манн, Иванов и Фербер, 2015.
3. Т.Е. Сорокина, поурочные разработки «Пропедевтика программирования со Scratch» для 5-го класса, 2015 г.
4. Учебно-методическое пособие. Проектная деятельность школьника в среде программирования Scratch. /В.Г. Рындак, В.О. Дженжер, Л.В. Денисова. - Оренбург - 2009
5. <http://scratch.mit.edu/pages/source> – страница разработчиков.
6. <http://scratch.mit.edu/> - официальный сайт проекта Scratch.
7. <http://setilab.ru/scratch/category/commun/>Сайт «Учитесь со Scratch»
8. http://minecraftnavideo.ru/play/vd20J2r5wUQ/scratch_lesson_01_znakomstvo_so_sredoj_programmirovaniya_scratch.html

Список литературы, рекомендованной обучающимся

1. [Первый шаг в робототехнику: практикум для 5–6 классов / Д. Г. Копосов / М.: БИНОМ. Лаборатория знаний, 2012](#)
2. <https://scratch.mit.edu/> – web сайт Scratch
3. <http://robot.edu54.ru/> - Портал «Образовательная робототехника»

Список литературы, рекомендованной родителям

1. Развитие ребенка в конструктивной деятельности. Справочное пособие / Н. В. Шайдурова / М.: Сфера, 2008
2. Робототехника для детей и их родителей / Ю. В. Рогов; под ред. В. Н. Халамова — Челябинск, 2012. — 72 с.: ил.<http://www.robogeek.ru/> - РобоГик, сайт, посвященный робототехнике
3. <http://wroboto.ru/> - Сайт, посвященный международным состязаниям роботов
4. <http://www.wedobots.com/> - Портал WeDo Bots