

Государственное бюджетное профессиональное образовательное учреждение
Псковской области «Опочецкий индустриально-педагогический колледж»
Центр цифрового образования детей «IT – куб»

СОГЛАСОВАНО

Педагогическим советом
Протокол № 2 от 09.02.2024 г.

УТВЕРЖДЕНО

приказом
исполняющего обязанности
директора колледжа
И.А. Гайдовской
от 23.04.2024г № 207

Дополнительная общеобразовательная общеразвивающая
программа технической направленности
«Основы алгоритмики и логики»

Направленность: техническая
Уровень программы: стартовый
Срок реализации: 1 год (144 часа)
Возраст детей: 10-12 лет

Разработчики:

Иванова С.П., зам. директора по
учебно-методической работе;
Иванова С.Т., методист

г. Опочка
2024 г.

СОДЕРЖАНИЕ

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА.....	3
УЧЕБНЫЙ ПЛАН	13
СОДЕРЖАНИЕ УЧЕБНО-ТЕМАТИЧЕСКОГО ПЛАНА	15
КОМПЛЕКС ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИХ УСЛОВИЙ.....	21
СПИСОК ЛИТЕРАТУРЫ	24
Приложение 1	25

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

1. Нормативно-правовая база разработки и реализации программы.

Программа разработана на основе следующих нормативных документов:

- Федерального закона от 29.12.2012 № 273-ФЗ (ред. от 31.07.2020) «Об образовании в Российской Федерации» (с изм. и доп., вступ. в силу с 01.09.2020);
- Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам, утвержденного приказом Министерства просвещения Российской Федерации от 27.07.2022 г. № 629;
- Постановления Главного государственного санитарного врача Российской Федерации от 28.09.2020 № 28 «Об утверждении санитарных правил СП 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи»;
- Постановления Главного государственного санитарного врача Российской Федерации от 28.01.2021 № 2 «Об утверждении санитарных правил и норм СанПиН 1.2.3685-21 Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания»;
- Приказа Минобрнауки России №882, Минпросвещения России №391 от 05.08.2020 «Об организации и осуществлении образовательной деятельности при сетевой форме реализации образовательных программ» (вместе с «Порядком организации и осуществления образовательной деятельности при сетевой форме реализации образовательных программ»);
- Методических рекомендаций по созданию и функционированию центров цифрового образования «IT-куб» (письмо Министерства просвещения Российской Федерации от 10 ноября 2021 г. № ТВ-1984/04);
- Стратегии развития воспитания в Российской Федерации на период до 2025 года (Утверждена распоряжением Правительства РФ от 29.05.2015 № 996-р «Об утверждении Стратегии развития воспитания в Российской Федерации на период до 2025 года»);

– Концепции развития дополнительного образования детей до 2030 года, утвержденной распоряжением Правительства Российской Федерации от 31 марта 2022 г. № 678-р.

2. Новизна, актуальность, педагогическая целесообразность реализации программы.

Актуальность данной образовательной программы состоит в том, что мультимедийная среда Scratch позволяет сформировать у детей интерес к программированию, отвечает всем современным требованиям объективно-ориентированного программирования. Изучение языка значительно облегчает последующий переход к изучению других языков программирования. Также стоит отметить, что большое количество времени уделяется творческим заданиям, выполнение которых благоприятно скажется на развитии творческого потенциала учащихся. Данная программа способствует развитию мотивации к получению новых знаний, возникновению интереса к программированию.

Новизна программы состоит в раскрытии особенно важных элементов в обучении программированию, формировании у обучающихся умения владения компьютером как средством решения практических задач, связанных с алгоритмизацией, программированием, и работой с информацией, подготовке учеников в активной полноценной жизни и работе в условиях технологически развитого общества.

Педагогическая целесообразность программы определяется образовательным замыслом. При изучении программирования в среде Scratch у обучающихся не только развивается логическое мышление, но и формируются навыки работы с мультимедиа. Освоение программы происходит в основном в процессе практической творческой деятельности. Знания и умения, приобретенные в результате освоения курса, являются базой для обучения программированию.

Направленность программы: техническая.

3. Адресат программы.

Адресатом программы являются дети в возрасте от 10 до 12 лет.

Содержание и объем стартовых знаний, необходимых для начального этапа освоения программы: умение читать и писать, решать арифметические задачи, иметь базовые навыки пользования ПК.

4. Срок реализации программы.

Срок реализации программы составляет 1 год.

Уровень программы: стартовый.

Программа предполагает использование и реализацию общедоступных и универсальных форм организации материала, минимальную сложность предлагаемого для освоения содержания программы.

5. Форма реализации программы.

Форма обучения – очная.

Образовательные технологии: информационные технологии, проектная технология, здоровьесберегающие технологии, технология проблемного обучения.

Форма организации содержания и процесса педагогической деятельности – комплексная.

Тип организации работы учеников: групповая работа, индивидуальная, коллективная.

Виды занятий: лекции и практические занятия.

Наполняемость группы: от 12 до 15 человек.

6. Объем программы и режим работы

Объем программы: 144 часа.

Режим занятий: 2 раза в неделю по 2 академических часа.

Продолжительность часа – 40 минут.

7. Цель программы:

Целью программы является создание условий для освоения обучающимися базовых навыков для изучения языков программирования в процессе проектной деятельности.

8. Задачи программы.

Программа направлена на решение следующих задач:

- познакомить с правилами безопасного использования цифровыми инструментами и компьютерным оборудованием, организации рабочего места;
- сформировать умения построения различных видов алгоритмов (линейных, разветвляющихся, циклических) для решения поставленных задач;
- сформировать умения использовать инструменты среды Scratch для решения поставленных задач;
- развитие алгоритмического, логического и образного мышления;
- развивать умение доводить решение задачи от идеи до работающего проекта;
- сформировать навык представления результатов своей работы окружающим, аргументировать свою позицию;
- развивать умение постановки технической задачи, собирать и изучать нужную информацию, находить конкретное решение задачи и осуществлять свой творческий замысел.
- воспитать умение работать в команде, развитие коммуникативных навыков;
- воспитать трудолюбие и уважительные отношения к интеллектуальному труду.

9. Планируемые результаты освоения программы

Знать:

- правила безопасного использования цифровыми инструментами и компьютерным оборудованием, организации рабочего места;
- виды алгоритмов и способы их реализации.

Уметь:

- строить различные виды алгоритмов (линейных, разветвляющихся, циклических) для решения поставленных задач;
- строить различные алгоритмы в среде Scratch для решения поставленных задач.

Владеть:

- навыком безопасного использования цифровых инструментов и компьютерного оборудования;
- навыком использования инструментов среды Scratch для решения поставленных задач.

Метапредметные результаты:

- развивать умение доводить решение задачи от идеи до работающего проекта;
- развивать умение постановки технической задачи, собирать и изучать нужную информацию, находить конкретное решение задачи и осуществлять свой творческий замысел;
- формирование умения представлять результаты своей работы окружающим, аргументировать свою позицию.

Личностные результаты:

- формирование уважительного отношения к интеллектуальному труду;
- развитие логического, алгоритмического и образного мышления;
- умение работать в команде, развитие коммуникативных навыков.

10. Формы представления результатов

Формы аттестации: наблюдение, опрос, демонстрация решения, проверочная работа, защита проекта. Наблюдение предполагает просмотр работ учащихся на занятии. Опрос предполагает разговор с учениками в конце занятия.

Демонстрация решения предполагает выход обучающегося к доске и показ выполненного проекта.

Входной контроль - имеет диагностические задачи и осуществляется в начале учебного года в виде наблюдения. Цель предварительной диагностики – зафиксировать начальный уровень подготовки обучающихся, имеющиеся знания, умения и навыки, связанные с предстоящей деятельностью.

Промежуточная аттестация проводится на основании диагностики теоретических знаний и практических умений и навыков по итогам освоения модуля. Промежуточная аттестация проводится в форме проверочной работы, состоящей из теста с выбором варианта ответа и практического задания.

Итоговый контроль проводится с целью определения степени достижения результатов обучения и получения сведений для совершенствования программы и методов обучения – представляет из себя защиту проекта.

11. Оценочные материалы, формирующие систему оценивания

Промежуточный контроль представляет из себя проверочную работу, включающую тест с выбором варианта ответа и практическое задание на разработку приложения в Scratch (приложение 1). Максимальный балл за проверочную работу: 50 баллов.

Оценивание проверочной работы осуществляется по следующим уровням:

- высокий уровень – учащийся набрал не менее 80% от максимально возможного количества баллов (от 40 баллов).
- средний уровень – учащийся набрал не менее 50% от максимально возможного количества баллов (от 25 баллов).
- низкий уровень – учащийся набрал менее 50% от максимально возможного количества баллов (менее 25 баллов).

Итоговый контроль представляет из себя защиту проекта, выполненного в среде Scratch.

Критерии оценки проекта

№	Название критерия	Максимальный балл
1.	Актуальность и проработанность проблемы	До 5 баллов
2.	Четкость формулировки целей и задач	До 5 баллов
3.	Технологическая сложность проекта	До 10 баллов
4.	Новизна и оригинальность решения	До 5 баллов
5.	Качество разработанного продукта	До 5 баллов
6.	Защита проекта: <ul style="list-style-type: none">• качество презентации;• четкость и ясность изложения, умение взаимодействовать с аудиторией, отвечать на вопросы	До 5 баллов
7.	Наличие самооценки и перспектив дальнейшей разработки проекта	До 5 баллов
Итого		40 баллов

Оценивание защиты проекта осуществляется по следующим уровням:

- высокий уровень – учащийся набрал не менее 32 баллов по итогам защиты проекта.
- средний уровень – учащийся набрал от 20 до 31 балла по итогам защиты проекта
- низкий уровень – учащийся набрал менее 20 баллов по итогам защиты проекта.

Общими критериями оценки результативности обучения являются:

- оценка уровня теоретических знаний: свобода восприятия теоретической информации, осмысленность и свобода использования специальной терминологии, свобода ориентации в теоретическом материале;
- оценка уровня практической подготовки: соответствие развития уровня

практических умений и навыков программным требованиям, свобода владения специальным оборудованием и оснащением;

- оценка уровня достижения личностных результатов: культура организации самостоятельной деятельности, культура работы с информацией, аккуратность и ответственность при работе.

Оценка итоговых результатов освоения программы осуществляется по трем уровням:

Высокий уровень – достижение 80 - 100% показателей освоения программы.

Средний уровень – достижение 50 - 79% показателей освоения программы.

Низкий уровень – достижение менее чем 50% показателей освоения программы.

Достигнутые обучающимся знания, умения и навыки заносятся в сводную таблицу результатов обучения.

Оценка уровней освоения программы

Уровни	Параметры	Показатели
Высокий уровень (80-100%)	Теоретические знания	Обучающийся освоил материал в полном объеме. Знает и понимает значение терминов, употребляет их осознанно и в полном соответствии с содержанием. Самостоятельно ориентируется в содержании материала по темам.
	Практические умения и навыки	Обучающийся овладел 80-100% умений и навыков, предусмотренных программой за конкретный период. Умет работать самостоятельно, применяя практические умения и навыки. Правильно и по назначению применяет инструменты. Умеет выполнять основные логические действия (анализ, синтез, установление причинно-следственных связей). Способен планировать и регулировать свою деятельность по реализации проекта. Умеет осуществлять поиск информации, в том числе в сети Интернет; выслушивать собеседника и вести диалог; выражать свои мысли в соответствии с задачами и условиями коммуникации.
	Личностные результаты	Обучающийся обладает внутренней мотивацией. Способен самостоятельно организовать собственную деятельность. Сформирована культура работы с информацией. Работу выполняет аккуратно, доводит до конца. Может оценить результаты выполнения своего задания и

		дать оценку работы своего товарища.
Средний уровень (50-79%)	Теоретические знания	Учащийся освоил базовые знания, ориентируется в содержании материала по темам, иногда обращается за помощью к педагогу. Использует специальную терминологию, однако сочетает её с бытовой
	Практические умения и навыки	Владеет базовыми навыками и умениями, но не всегда может выполнить задание самостоятельно, просит помощи педагога. В основном выполняет задания на основе образца. Способен разрабатывать алгоритм решения при помощи преподавателя. Делает ошибки в работе, но может устранить их после наводящих вопросов или самостоятельно Испытывает незначительные затруднения при выполнении основных логических действий (анализ, синтез, установление причинно-следственных связей). Способен планировать и регулировать свою деятельность по реализации проекта с помощью педагога. Испытывает незначительные сложности в осуществлении коммуникации с педагогом и сверстниками.
	Личностные результаты	Внутренняя мотивация к обучению сочетается с внешней. В работе допускает небрежность. Работу не всегда выполняет аккуратно и/или доводит до конца. Оценить результаты своей деятельности может с подсказкой педагога.
Низкий уровень (меньше 50%)	Теоретические знания	Владеет минимальными знаниями, ориентируется в содержании материала по темам только с помощью педагога. Избегает употреблять специальные термины.
	Практические умения и навыки	Владеет минимальными начальными навыками и умениями. Учащийся способен выполнять каждую операцию только с подсказкой педагога или товарищей. В работе допускает грубые ошибки, не может их найти даже после указания преподавателя. В состоянии выполнять лишь простейшие практические задания педагога. Испытывает существенные затруднения при выполнении основных логических действий (анализ, синтез, установление причинно-следственных связей). Не способен планировать и регулировать свою деятельность по реализации проекта. Испытывает значительные сложности в осуществлении коммуникации с педагогом и сверстниками.
	Личностные результаты	Преобладает внешняя мотивация к обучению. Работу часто выполняет неаккуратно и/или не доводит до конца. Не способен самостоятельно и объективно оценить результаты своей работы.

Сводная таблица результатов обучения по программе

№ п/п	Фамилия, имя обучающегося	Оценка теоретических знаний	Оценка практических умений и навыков (предметных и метапредметных)	Личностные результаты	Итоговая оценка
----------	------------------------------	-----------------------------------	--	--------------------------	--------------------

УЧЕБНЫЙ ПЛАН

№ п/п	Название разделов, тем	Количество часов			Форма аттестации/ контроля
		Всего	Теория	Практика	
	1. Основы алгоритмизации и программирования	6	4	2	
1.1	Вводное занятие	2	2	0	Наблюдение
1.2	Введение. Понятие алгоритма. Решение задач.	4	2	2	Наблюдение
	2. Работа со средой Scratch	70	23	47	
2.1	Знакомство со средой Scratch. Навигация в среде Scratch.	2	1	2	Наблюдение
2.2	Блоки движения и событий	2	1	2	Наблюдение
2.3	Знакомство с линейными алгоритмами в Scratch.	2	1	1	Демонстрация решения
2.4	Знакомство с блоками «Внешний вид». Анимация персонажей с использованием костюмов, фонов из библиотеки.	5	1	4	Демонстрация решения
2.5	Знакомство с графическим редактором Scratch. Создание первого спрайта и фона.	5	2	3	Наблюдение
2.6	Работа со звуком Scratch	4	2	2	Демонстрация решения
2.7	Знакомство с пером	4	1	3	Наблюдение
2.8	Знакомство с отрицательными числами и координатами.	2	1	1	Наблюдение
2.9	Условные алгоритмы	6	2	4	Наблюдение
2.10	Понятие переменная. Работа с переменными.	4	2	2	Наблюдение
2.11	Знакомство с клонами	2	1	3	Демонстрация решения
2.12	Циклические алгоритмы. Блоки «Управления».	6	2	4	Наблюдение
2.13	Создание проекта с вычислениями. Случайные числа.	5	1	4	Демонстрация решения
2.14	Создание проекта с подсчетом очков	5	1	4	Демонстрация решения
2.15	Работа со списками	6	2	4	Демонстрация решения
2.16	Создание подпрограмм	6	2	4	Демонстрация решения
	3. Знакомство со средой программирования Spike App	8	1	7	
3.1	Знакомство с конструктором Lego Spike Prime.	2	1	1	Наблюдение

3.2	Сборка робота	2	-	2	Наблюдение
3.3	Программирование робота	4	-	4	Демонстрация решения
	4. Создание мультфильмов и игр.	60	16	44	
4.1	Создание мультфильма. Разработка темы и персонажей	4	1	3	Демонстрация решения
4.2	Создание мультфильма. Программирование	4	1	3	Демонстрация решения
4.3	Создание игры. Разработка темы и персонажей	4	1	3	Демонстрация решения
4.4	Создание игры. Программирование	4	1	3	Демонстрация решения
4.5	Создание игры с несколькими уровнями. Разработка темы и персонажей	4	1	3	Демонстрация решения
4.6	Программирование игры с несколькими уровнями.	4	1	3	Демонстрация решения
4.7	Добавление анимации.	4	1	3	Демонстрация решения
4.8	Добавления подсчета очков.	4	1	3	Демонстрация решения
4.9	Создание собственного проекта. Тема. Цели, задачи, изучение предметной области.	6	3	3	Демонстрация решения
4.10	Создание собственного проекта. Программирование	4	1	3	Демонстрация решения
4.11	Создание собственного проекта. Добавление музыки и анимации	4	1	3	Демонстрация решения
4.12	Создание собственного проекта. Разработка презентации	4	1	3	Демонстрация решения
4.13	Создание собственного проекта. Подготовка к защите	4	1	3	Демонстрация решения
4.14	Создание собственного проекта. Репетиция защиты	4	1	3	Демонстрация решения
4.15	Защита итогового проекта	2	-	2	Защита проекта.
	Итого	144	44	100	

СОДЕРЖАНИЕ УЧЕБНО-ТЕМАТИЧЕСКОГО ПЛАНА

1. Основы алгоритмизации и программирования. (*теория – 4 часа, практика – 2 часа*)

1.1 Вводное занятие

Теория (2 часа). Правила работы в компьютерном кабинете. Инструктаж по технике безопасности. Введение в курс программирования на Scratch.

1.2 Введение. Понятие алгоритма. Решение задач

Теория (2 часа). Понятие алгоритма и исполнителя алгоритмов. Понятие отладки программы. Основные алгоритмические конструкции: линейный алгоритм, циклический алгоритм, алгоритм ветвления. Способы представления алгоритмов (словесный, графический, программный). Определяемые допустимые действия.

Практика (2 часа). Решения задач с использованием алгоритмов.

2. Работа со средой Scratch. (*теория – 23 часа, практика – 47 часов*)

2.1 Знакомство со средой Scratch.

Теория (1 час). Изучение основных элементов интерфейса Scratch

Практика (2 часа). Изучение пользовательского интерфейса 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. Основной персонаж как исполнитель программ. Система команд исполнителя (СКИ). Блочная структура программы. Непосредственное управление исполнителем.

2.2 Блоки движения и событий

Теория (1 час). Изучение блоков движения и событий. Запуск проектов с помощью различных событий.

Практика (2 часа). Обучение спрайта различным движением. Запуск кода с помощью разных событий.

2.3 Знакомство с линейными алгоритмами Scratch

Теория (1 час). Изучение блоков, позволяющих создать линейный алгоритм.

Практика (1 час). Создание простейшего приложения с использованием линейных алгоритмов.

2.4 Знакомство с блоками «Внешний вид». Анимация персонажей с использованием костюмов, фонов из библиотеки

Теория (1 час). Изучение блоков «Внешний вид». Добавление нескольких персонажей и фонов.

Практика (4 часа). Создание анимированного персонажа с использованием готовых костюмов.

2.5 Знакомство с графическим редактором Scratch. Создание первого спрайта и фона

Теория (2 часа). Графический редактор Scratch. Растровый и векторный режимы. Редактирование спрайта в векторном графическом редакторе. Слои изображения. Группировка фигур. Сохранение отредактированного спрайта в отдельный файл. Создание нового спрайта с несколькими костюмами в редакторах Scratch. Сохранение нового спрайта в отдельный файл. Создание собственного фона.

Практика (3 часа). Изменение готового персонажа из библиотеки. Сохранение нового спрайта. Создание собственного спрайта и фона.

2.6 Работа со звуком Scratch

Теория (2 часа). Изучение назначения блоков «Звук». Запись собственного звука. Добавление звука в проект с рабочего стола.

Практика (2 часа). Работа со звуками.

2.7 Знакомство с пером

Теория (1 час). Изучение блоков «Перо».

Практика (3 часа). Создание программы для рисования линий.

2.8 Знакомство с отрицательными числами. Знакомство с координатами

Теория (1 час). Изучение понятия «отрицательное число». Применение отрицательных чисел в различных блоках. Изучение координатной плоскости. Перемещение спрайта по сцене с помощью координат. Изучение понятия «нулевые координаты».

Практика (1 час). Добавление отрицательных чисел в ранее созданные проекты. Создание приложения, в котором спрайт перемещается по сцене по разным координатам.

2.9 Условные алгоритмы

Теория (2 часа). Изучение понятия «условный алгоритм». Использование сенсоров касания и команды «если ..., то...». Управление движением персонажа с помощью мыши и клавиатуры.

Практика (4 часа). Создание приложения с использованием условных алгоритмов.

2.10 Понятие переменная. Работа с переменными

Теория (2 часа). Изучение понятия «переменная», для чего нужны переменные. Изучение блоков «Переменные».

Практика (2 часа). Создание приложения с использованием переменных.

2.11 Знакомство с клонами

Теория (1 час). Изучение блоков «Клоны». Изучение возможностей клонов.

Практика (3 часа). Создание проекта с клонами.

2.12 Циклические алгоритмы. Блоки «Управления»

Теория (2 часа). Изучение блоков «Управления». Изучение понятия «цикл».

Практика (4 часа). Создание приложения с использованием циклических алгоритмов.

2.13 Создание проекта с вычислениями. Случайные числа

Теория (1 час). Изучение блоков «Операторы». Работа с арифметическими операциями, изучение понятия «случайное число».

Практика (4 часа). Создание приложения со случайными числами количества костюмов спрайта.

2.14 Создание проекта с подсчетом очков

Теория (1 час). Создание переменной для подсчета очков. Разработка темы игры.

Практика (4 часа). Создание игры с подсчетом очков.

2.15 Работа со списками

Теория (2 часа). Изучение понятия «Список». Создание списка. Изучение блоков для работы со списками.

Практика (4 часа). Добавление в ранее созданный проект списка.

2.16 Создание подпрограмм

Теория (2 часа). Последовательность и параллельность выполнения скриптов. Управление через обмен сообщениями. Блоки «Передать сообщение» и «Когда я получу сообщение».

Практика (4 часа). Создание проекта с подпрограммами.

3. Знакомство со средой программирования Spike App (*теория – 1 час, практика – 7 часов*)

3.1 Знакомство с конструктором Lego Spike Prime

Теория (1 час). Изучение деталей конструктора. Просмотр обучающих видео.

Практика (1 час). Сборка простого робота.

3.2 Сборка робота

Практика (2 часа). Сборка выбранной модели робота.

3.3 Программирование робота

Практика (4 часа). Программирование выбранного робота.

4. Создание мультфильмов и игр (*теория – 18 часов, практика – 42 часа*)

4.1 Создание мультфильма. Разработка темы и персонажей

Теория (1 час). Обсуждение темы мультфильма.

Практика (3 часа). Создание собственного мультфильма. Прорисовка персонажей и фона. Разработка сценария. Озвучивание и анимация персонажей.

4.2 Создание мультфильма. Программирование

Теория (1 час). Обсуждение программирование мультфильма.

Практика (3 часа). Программирование мультфильма.

4.3 Создание игры. Разработка темы и персонажей

Теория (1 час). Обсуждение темы и персонажей проекта.

Практика (3 часа). Создание собственной игры. Создание собственного игрового интерфейса. Реализации игры с одним уровнем.

4.4 Создание игры. Программирование

Теория (1 час). Обсуждение программирование игры.

Практика (3 часа). Программирование игры.

4.5 Создание игры с несколькими уровнями. Разработка темы и персонажей

Теория (1 час). Обсуждение темы игры. Обдумывание персонажей.

Практика (3 часа). Создание игры с несколькими уровнями и подсчётом очков.

4.6 Программирование игры с несколькими уровнями

Теория (1 час). Обсуждение программирования игры.

Практика (3 часа). Программирование игры.

4.7 Добавление анимации

Теория (1 час). Обсуждение анимации.

Практика (3 часа). Добавление анимации к игре.

4.8 Добавление подсчёта очков

Теория (1 час). Обсуждение добавления переменной.

Практика (3 часа). Добавление переменной «очки». Программирование добавления очков в игре.

4.9 Создание собственного проекта. Цели, задачи, изучение предметной области

Теория (3 часа). Обсуждение темы проекта.

Практика (3 часа). Выбор темы проекта. Разработка целей и задач, изучение предметной области.

4.10 Создание собственного проекта. Программирование

Теория (1 час). Обсуждение программирования.

Практика (3 часа). Программирование проекта.

4.11 Создание собственного проекта. Добавление музыки и анимации

Теория (1 час). Обсуждение анимации и музыки.

Практика (3 часа). Добавление в проект музыки и анимации.

4.12 Создание собственного проекта. Разработка презентации

Теория (1 час). Обсуждение создания презентации.

Практика (3 часа). Разработка презентации к защите проекта.

4.13 Создание собственного проекта. Подготовка к защите проекта

Теория (1 час). Обсуждение речи.

Практика (3 часа). Проверка проекта. Создание речи для защиты проекта.

4.14 Создание собственного проекта. Репетиция защиты

Теория (1 час). Создание презентации для защиты.

Практика (3 часа). Проверка проекта. Репетиция речи для защиты проекта.

4.15 Защита итогового проекта

Практика (2 часа). Защита итогового проекта.

КОМПЛЕКС ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИХ УСЛОВИЙ

Материально-техническое обеспечение

Комплекс условий реализации программы:

- кабинет, оснащенный компьютерной техникой, не менее 1 ПК на 1 ученика.
- образовательный набор Lego Spike Prime;
- ПО: Scratch 3, Lego Education Spike, любой браузер.

Аппаратное и техническое обеспечение:

Рабочее место учащегося:

- ноутбук с выходом в сеть Интернет.

Рабочее место наставника:

- ноутбук с выходом в сеть Интернет;
- технические средства обучения (ТСО) (мультимедийное устройство).

Методическое обеспечение программы

Методы обучения:

- объяснительно-иллюстративный (беседы, объяснения);
- репродуктивный (деятельность обучаемых носит алгоритмический характер, выполняется по инструкциям, предписаниям, правилам в аналогичных, сходных с показанным образцом ситуациях);
 - метод проблемного изложения;
 - эвристический (метод обучения заключается в организации активного поиска решения выдвинутых в обучении (или самостоятельно сформулированных) познавательных задач в ходе подготовки и реализации творческих проектов);
- исследовательский.

Педагогические технологии: информационные технологии, проектная технология, здоровьесберегающие технологии, технология проблемного обучения.

В процессе обучения учащиеся работают с текстовыми редакторами, редакторами презентаций, средой программирования Scratch.

Проектная технология дает возможность самостоятельно конструировать свои знания, ориентироваться в информационном пространстве, развивает критическое и творческое мышление, создаёт условия для формирования и развития внутренней мотивации учащихся к более качественному овладению знаниями, повышения мыслительной активности и приобретения навыков логического мышления.

Здоровьесберегающие технологии позволяют создать максимально возможные условия для сохранения, укрепления и развития эмоционального, интеллектуального и физического здоровья, в том числе в условиях работы с компьютерной техникой.

Проблемное обучение — это тип развивающего обучения, содержание которого представлено системой проблемных задач различного уровня сложности, в процессе решения которых учащиеся овладевают новыми знаниями и способами действия, а через это происходит формирование творческих способностей: продуктивного мышления и познавательной мотивации.

Учебно-методические средства обучения.

- специализированная литература по направлению, подборка журналов,
- образцы программ и систем, выполненные обучающимися и педагогом,
- учебно-методические пособия для педагога и обучающихся, включающие дидактический, информационный, справочный материалы на различных носителях, компьютерное и видео оборудование.

Применяемое на занятиях дидактическое и учебно-методическое обеспечение включает в себя электронные учебники, справочные материалы и системы используемых программ, интернет, рабочие тетради обучающихся.

Кадровое обеспечение

Программу реализуют педагоги дополнительного образования «IT-куб».

Информационное обеспечение

Для реализации программы планируется использование следующих информационных ресурсов:

- набор карточек с инструкциями «Карточки Scratch» (<https://aovchin67.files.wordpress.com/2019/12/scratch-cards-all.pdf>);
- курс «Scratch. Новичок» (<https://stepik.org/course/121602/syllabus>);
- курс «Мир Scratch» (<https://stepik.org/course/90478/syllabus>).

СПИСОК ЛИТЕРАТУРЫ

Список литературы для педагога

1. Лебедева, Татьяна Николаевна, Информатика в начальной школе учебное пособие / Т. Н. Лебедева, С. С. Юнусова ; Южно-Уральский государственный гуманитарно-педагогический университет. — [Челябинск] ; Южно-Уральский научный центр РАО, 2021. — 180 с. : ил.
2. Винницкий Ю. А. Scratch и Arduino для юных программистов и конструкторов./ Винницкий Ю. А. — СПб.: БХВ-Петербург, 2018. — 176 с.
3. Поляков К. Ю. Информатика. 7 класс (в 2 частях): учебник. Ч. 1 / Поляков К. Ю., Еремин Е. А. — М.: БИНОМ. Лаборатория знаний, 2019. — 160 с.
4. Левитин А. Алгоритмические головоломки / А. Левитин, М., Левитина ; пер. с англ. Ж. А. Меркуловой, Н. А. Меркулова. — 2-е изд. — М. : Лаборатория знаний, 2019. — 325 с. : ил.
5. Пашковская Ю. В. Scratch 3.0: творческие проекты на вырост : рабочая тетрадь для 7—8 классов / Ю В. Пашковская ; под ред. В. В. Тарапаты. — М. : Лаборатория знаний, 2023. — 187 с. : ил. — (Школа юного программиста).

Список литературы для учащихся

1. Голиков Д. В. 42 проекта на Scratch 3 для юных программистов. / Голиков Д. В. — СПб.: БХВ-Петербург, 2019. — 185 с.
2. Сорокина Т. Е. Информатика. 5 — 6 классы. Практикум по программированию в среде Scratch. / Т. Е. Сорокина, А. Ю. Босова; под ред. Л. Л. Босовой. — М. : БИНОМ. Лаборатория знаний, 2021. — 146 с. : ил.
3. Пашковская Ю. В. Творческие задания в среде Scratch: рабочая тетрадь для 5—6 классов. / Пашковская Ю. В. — М., 2022. — 195 с.
4. Свейгарт Эл. Scratch 3. Изучайте язык программирования, делая крутые игры!/ Свейгарт Эл. — М.: Эксмо, 2023. — 226 с.

Примерные задания для промежуточной аттестации

1. Тестирование на знание Scratch. Выберите один правильный ответ.

1. Как называется графический объект, который выполняет различные алгоритмы, написанные пользователем в Scratch? (2 балла).

- а) скрипт
- б) спрайт
- в) котенок
- г) сцена

2. Сколько костюмов может быть у персонажа? (2 балла).

- а) 1
- б) 2
- в) любое количество
- г) не более 7

3. Какой блок позволяет изменить направление движения у края экрана? (2 балла).

- а) «если касается края, повернуться»
- б) «если касается края, отскочить»
- в) «если касается края, оттолкнуться»

4. Как называется алгоритм, составленный из блоков Scratch для какого-нибудь объекта? (2 балла).

- а) спрайт
- б) сцена
- в) код
- г) скрипт

5. Что является выделенной областью памяти компьютера, в которой хранятся данные? (2 балла).

- а) буква
- б) цифра
- в) блок

г) переменная

6. Какая группа блоков в Scratch содержит математические команды? (2 балла).

а) математика

б) сенсоры

в) операторы

г) другие блоки

7. Можно ли делать проект в Scratch, в котором нет сцены? (2 балла)?

а) нет

б) да

в) иногда можно

8. Какое расширение у приложений, созданных в Scratch? (2 балла).

а) .sb3

б) .py

в) .exe

г) .apk

9. Какая группа блоков хранит команды для работы с клонами? (2 балла)

а) сенсоры

б) события

в) операторы

г) управление

10. Координата ... отвечает за положение влево и вправо, координата ... - за вверх и вниз? Выберите вариант ответа, в котором координаты расположены в правильном порядке.

а) x, y

б) y, x

в) x, u

г) u, x

2. Создание игры в Scratch.

Нарисованы спрайты и фон – 10 баллов.

Использованы циклы – 5 баллов.

Использована анимация – 5 баллов.

Использованы условия – 5 баллов

Использованы переменные – 5 баллов.

Максимальное количество баллов за задание – 30 баллов.