

**муниципальное образовательное учреждение дополнительного образования
«Центр дополнительного образования детей Дзержинского района»
(МОУ ДО «Центр дополнительного образования детей
Дзержинского района»)**

СОГЛАСОВАНО
Методический совет
от «18» августа 2026 г.
Протокол № 1

УТВЕРЖДЕНА
Директор  М.В. Мирошникова
Приказ № 01-07/524 от 25.08.2026 г.
Принята на заседании Педагогического совета
Протокол № 1 от «25» августа 2026 г.



**Дополнительная общеобразовательная общеразвивающая программа
«Код будущего»**

Направленность программы – техническая

Срок реализации: 1 год
Возраст обучающихся: 10 – 18 лет

Автор-составитель:
педагог дополнительного образования
Тюленев Сергей Алексеевич

Ярославль, 2026 г.

Оглавление

1. Пояснительная записка	3
2. Учебно-тематический план	5
3. Содержание программы	9
4. Воспитательный модуль	16
5. Ожидаемые результаты	18
6. Обеспечение программы	20
7. Контрольно-измерительные материалы	22
8. Список информационных источников	26

1. Пояснительная записка

Актуальность программы

В эпоху цифровой трансформации информационные технологии проникают во все сферы жизни — от образования и медицины до производства и искусства. Согласно концепции технологического развития Российской Федерации до 2030 года, подготовка квалифицированных ИТ-кадров является одним из приоритетных направлений государственной политики. Спрос на специалистов в области программирования, веб-разработки, мобильной разработки и искусственного интеллекта растет с каждым годом, при этом дефицит кадров в ИТ-сфере продолжает увеличиваться.

Раннее знакомство с основами программирования позволяет ребенку не только освоить востребованную профессию в будущем, но и развивает алгоритмическое мышление, умение решать сложные задачи, работать с информацией и принимать нестандартные решения. Эти навыки становятся универсальными и применимыми в любой сфере деятельности, независимо от выбранной профессии.

Особенность современного этапа развития образования заключается в необходимости не просто дать ребенку знания, но и сформировать у него метапредметные компетенции: умение учиться, анализировать, проектировать, работать в команде и создавать законченные продукты. Программирование — это идеальная среда для формирования этих качеств, поскольку оно соединяет логику, творчество и практическую реализацию идей.

Дополнительная общеобразовательная общеразвивающая программа «Программирование для начинающих: от Scratch до профессиональной разработки» ориентирована на детей и подростков 10-18 лет, интересующихся информационными технологиями, и направлена на постепенное погружение в мир программирования через игровые и проектные формы работы. Программа построена по принципу «от простого к сложному», позволяя каждому ученику двигаться в комфортном для него темпе и находить свою траекторию развития в мире ИТ.

Программа соответствует требованиям нормативных документов:

1. Концепция развития дополнительного образования детей до 2030 года, утв. распоряжением Правительства РФ от 31 марта 2022 года № 678-р.
2. Федеральный закон «Об образовании в Российской Федерации» № 273-ФЗ от 29.12.2012 года.
3. Приказ Минпросвещения России от 27.07.2022 № 629 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам» (Зарегистрировано в Минюсте России 26.09.2022 № 70226).

Направленность программы – техническая, вид деятельности – программирование.

Возраст обучающихся: 10-18 лет.

Срок реализации программы: программа рассчитана на 1 год обучения (9 месяцев), объем 144 часа в год.

Форма обучения: очная.

Режим занятий: занятия проводятся 2 раза в неделю, продолжительность одного занятия – 2 академических часа, длительность одного часа 40 минут, перерыв между занятиями 10 минут.

Наполняемость учебных групп: оптимальная численность группы — 10-12 человек, возможны индивидуальные и групповые занятия (5-6 человек) при особых условиях обучения.

Цель программы – формирование у обучающихся устойчивых навыков программирования, алгоритмического мышления и проектной культуры через последовательное освоение современных языков и технологий разработки, а также профессиональное самоопределение в сфере информационных технологий.

Обучающие задачи:

- Познакомить обучающихся с основами алгоритмизации и структурного программирования через визуальные среды (Scratch, MIT App Inventor, Construct 3).
- Сформировать базовые навыки написания кода на языке Python: работа с переменными, условиями, циклами, функциями, списками и библиотеками.
- Научить создавать интерактивные веб-страницы с использованием HTML, CSS и JavaScript.
- Познакомить с основами мобильной разработки на языке Kotlin и визуальной среде MIT App Inventor.
- Сформировать навыки создания VK-ботов с использованием API и работы с внешними данными.
- Научить работать с системами контроля версий (Git) и публиковать проекты на GitHub.
- Сформировать представление о современных IT-профессиях и требованиях к специалистам в области разработки программного обеспечения.

Развивающие задачи:

- Развивать алгоритмическое, логическое и критическое мышление, умение разбивать сложную задачу на подзадачи.
- Развивать творческие способности через создание собственных проектов: игр, приложений, сайтов и ботов.
- Развивать навыки проектной деятельности: планирование, постановка целей, управление временем, презентация результатов.
- Формировать способность к самообучению и самостоятельному поиску информации в технической документации и интернет-источниках.
- Развивать коммуникативные навыки через работу в парах и групповые проекты, умение объяснять свои решения и вести конструктивный диалог.

Воспитательные задачи:

- Воспитывать ответственное отношение к результатам своего труда, аккуратность и внимательность при написании кода.
- Формировать нормы информационной безопасности, цифровой гигиены и этики поведения в сети Интернет.
- Воспитывать уважение к интеллектуальной собственности и понимание важности лицензионной чистоты программного обеспечения.
- Формировать бережное отношение к компьютерной технике, соблюдение правил работы в компьютерном классе.
- Содействовать профессиональному самоопределению обучающихся через знакомство с реальными IT-профессиями и практиками разработки.

2. Учебно-тематический план

№ темы	Название темы	Кол-во учебных часов		
		Теория	Практика	Всего
Модуль 1. Блочное программирование				
1	Техника безопасности при работе на ПК. Знакомство со Scratch. Интерфейс и первая анимация	1	1	2
2	Циклы и клонирование. Создание дождя из падающих объектов	1	1	2
3	Условия (if-else). Счетчик очков. Создание игры-ловушки	1	1	2
4	Таймеры и прогрессивная сложность в играх	1	1	2
5	Враги и столкновения. Простейший искусственный интеллект	1	1	2
6	Дизайн и интерфейс. Создание меню и переключение экранов	1	1	2
7	Тестирование и отладка проектов. Код-ревью	0,5	1,5	2
8	Защита проектов. Презентация игр	0,5	1,5	2
Итого по модулю 1:		7	9	16
Модуль 2. Основы Python				
9	Установка Python и среды разработки. Переменные и print()	1	1	2
10	Ввод данных (input). Создание интерактивной программы	1	1	2
11	Арифметические операции. Создание консольного калькулятора	1	1	2
12	Условные конструкции (if-elif-else). Проверка пароля	1	1	2
13	Циклы (for и while). Таблица умножения	1	1	2
14	Списки (list). Операции добавления, удаления, поиска	1	1	2
15	Функции (def). Создание собственных функций	1	1	2
16	Итоговый проект — Викторина. Защита проекта	0,5	1,5	2
Итого по модулю 2:		7,5	8,5	16
Модуль 3. VK-боты				
17	Знакомство с API и библиотекой vk_api	1	1	2
18	Регистрация бота. Эхо-бот	1	1	2
19	Обработка команд и состояния	1	1	2
20	Бот-калькулятор	1	1	2
21	Клавиатуры и меню	1	1	2
22	Бот-викторина с хранением состояния	1	1	2
23	Отправка картинок и медиафайлов	0,5	1,5	2

24	Деплой (запуск 24/7)	0,5	1,5	2
Итого по модулю 3:		7	9	16
Модуль 4. Веб-разработка (HTML + CSS)				
25	Введение в веб-разработку. Структура HTML-документа	1	1	2
26	Ссылки и изображения. Создание галереи и навигации	1	1	2
27	CSS — цвета, шрифты, оформление. Подключение стилей	1	1	2
28	Блочная модель (Box Model). Создание карточек с рамками и тенями	1	1	2
29	Flexbox — гибкое расположение элементов. Создание сеток	1	1	2
30	Навигация и hover-эффекты. Создание меню сайта	1	1	2
31	Формы и элементы ввода. Создание формы обратной связи	1	1	2
32	Публикация на GitHub Pages. Получение живой ссылки	0,5	1,5	2
Итого по модулю 4:		7,5	8,5	16
Модуль 5. JavaScript и интерактивность				
33	Подключение JavaScript. Alert, Prompt, Console	1	1	2
34	События (onclick). Счетчик кликов. Изменение контента	1	1	2
35	Работа с DOM. Изменение стилей. Переключение тем	1	1	2
36	Условные конструкции (if) в JS. Проверка пароля	1	1	2
37	Циклы в JS. Динамическое создание элементов списка	1	1	2
38	Таймеры (setInterval). Создание секундомера	1	1	2
39	Массивы объектов. Создание интерактивной галереи	1	1	2
40	Итоговый проект — Интерактивная страница	0,5	1,5	2
Итого по модулю 5:		7,5	8,5	16
Модуль 6. Canvas и браузерные игры				
41	Canvas — рисование фигур. Система координат	1	1	2
42	Анимация (requestAnimationFrame). Движение объектов	1	1	2
43	Управление клавишами. Управление игровым персонажем	1	1	2
44	Столкновения (Hitbox). Сбор предметов и подсчет очков	1	1	2

45	Игра «Змейка» — разработка логики	1	1	2
46	Игра «Змейка» — финал, счет, рекорд, дизайн	0,5	1,5	2
47	Игра «Арканоид» — создание кирпичей и разрушение	1	1	2
48	Презентация игр. Защита проектов	0,5	1,5	2
Итого по модулю 6:		7	9	16
Модуль 7. Мобильная разработка				
49	Введение в MIT App Inventor. Создание первого приложения	1	1	2
50	Переключение экранов. Передача данных между экранами	1	1	2
51	Списки (ListView). Добавление и удаление элементов	1	1	2
52	Хранение данных (TinyDB). Сохранение настроек	1	1	2
53	Сенсоры (акселерометр). Приложение, реагирующее на тряску	1	1	2
54	То-до лист — разработка приложения для задач	1	1	2
55	Сборка APK. Установка приложения на телефон	0,5	1,5	2
56	Презентация мобильных приложений	0,5	1,5	2
Итого по модулю 7:		7	9	16
Модуль 8. Хакатон и командная разработка				
57	Введение в Git и GitHub. Регистрация и создание репозитория	1	1	2
58	Работа с ветками (branches). Совместная разработка	1	1	2
59	Выбор финального проекта. Составление технического задания	1	1	2
60	Разработка проекта (день 1). Реализация базовой функциональности	0,5	1,5	2
61	Разработка проекта (день 2). Добавление функций и дизайна	0,5	1,5	2
62	Тестирование и отладка. Поиск и исправление ошибок	0,5	1,5	2
63	Оформление проекта. Создание ReadMe.md	0,5	1,5	2
64	Предзащита проектов. Репетиция выступления	0,5	1,5	2
Итого по модулю 8:		5,5	10,5	16
Модуль 9. Финальный проект и Demo Day				
65	Полировка проектов. Исправление последних ошибок	0,5	1,5	2
66	Подготовка портфолио. Сбор всех проектов за год	0,5	1,5	2

67	Профориентация. Знакомство с IT-профессиями	1,5	0,5	2
68	Генеральная репетиция Demo Day	0,5	1,5	2
69	Demo Day. Часть 1. Защита проектов первой группы	0,5	1,5	2
70	Demo Day. Часть 2. Защита проектов второй группы	0,5	1,5	2
71	Вручение сертификатов. Награждение в номинациях	1	1	2
72	Итоговое занятие. Подведение итогов года. Чаепитие	0,5	1,5	2
Итого по модулю 9:		5,5	10,5	16
ВСЕГО ПО ПРОГРАММЕ:		61	83	144

Календарный учебный график

Год обучения	Дата освоения программы		Количество учебных недель	Количество учебных часов	Режим занятий в неделю
	начало	окончание			
1	первая неделя сентября	последняя неделя мая	36	144	2 раза по 2 часа

3. Содержание программы

Сентябрь. Модуль 1: Блочное программирование (Scratch)

Занятие 1. Знакомство со Scratch. Движение спрайта.

Регистрируемся на scratch.mit.edu. Изучаем интерфейс: сцена, спрайты, палитра блоков. Собираем первую программу — кот движется по квадрату. Используем блоки «Когда зеленая нажата», «Идти 100 шагов», «Повернуть на 90 градусов» в цикле «Повторить 4 раза». Запускаем анимацию.

Занятие 2. Циклы и клонирование. Дождь.

Создаем спрайт капли. Учимся использовать бесконечный цикл и клоны. Код: каждая капля — клон, который падает вниз и исчезает при касании края. Добавляем случайную скорость для каждой капли. Понимание экономии памяти через клоны.

Занятие 3. Условия и счетчик очков.

Добавляем корзинку внизу. Создаем переменную «Очки». Пишем условие: если капля касается корзинки, то увеличиваем счет на 1 и удаляем каплю. Если капля касается пола — просто удаляем. Первая полноценная механика игры.

Занятие 4. Таймер и прогрессивная сложность.

Добавляем таймер на 60 секунд. Привязываем скорость падения к оставшемуся времени: чем меньше времени, тем быстрее капли. Добавляем победу при 30 очках и поражение при окончании времени. Игра становится динамичной.

Занятие 5. Враги и столкновения.

Создаем спрайт призрака. Он хаотично двигается по экрану. Добавляем условие: если призрак касается корзинки — Game Over. Усложняем: призрак преследует корзинку (простейший AI через блок «Идти к»).

Занятие 6. Дизайн и меню.

Создаем стартовый экран с кнопкой «Старт». Делаем экран Game Over с кнопкой «Начать сначала». Учимся переключать фоны. Настраиваем слои, чтобы кнопки были поверх всего. Добавляем звуки и улучшаем внешний вид.

Занятие 7. Тестирование и отладка.

Меняемся ноутбуками. Задание: найти 3 ошибки в игре соседа (баги). Исправляем их. Проводим код-ревью: оцениваем читаемость кода, наличие комментариев. Обсуждаем типичные ошибки.

Занятие 8. Презентация проектов.

Публикуем игры на сайте Scratch. Каждый ученик выходит к проектору, показывает свою игру и объясняет ключевые моменты кода. Защита проекта, вопросы от группы.

Октябрь. Модуль 2: Основы Python

Занятие 9. Переменные и print().

Устанавливаем Python и среду разработки Thonny. Выводим первую надпись «Hello World!». Учимся создавать переменные: имя, возраст, город. Выводим их на экран. Понимание разницы между текстом и числом.

Занятие 10. Ввод данных (input).

Программа спрашивает у пользователя имя, возраст и отвечает приветствием. Работаем с конкатенацией строк. Понимаем, что input всегда возвращает строку. Первый интерактивный диалог.

Занятие 11. Калькулятор (арифметика).

Учимся преобразовывать строки в числа через `int()`. Пишем простой калькулятор сложения двух чисел. Добавляем вычитание, умножение и деление. Обрабатываем ошибку деления на ноль.

Занятие 12. Условия (if-elif-else).

Проверка пароля: если ввел 1234 — доступ разрешен, иначе — запрещен. Добавляем три попытки. Пишем программу, которая определяет возрастную категорию (ребенок, подросток, взрослый).

Занятие 13. Циклы (for и while).

Выводим таблицу умножения через цикл `for`. Считаем сумму чисел от 1 до 100 через `while`. Понимаем разницу: `for` — когда знаем количество повторений, `while` — когда не знаем.

Занятие 14. Списки (list).

Создаем список любимых игр. Учимся добавлять новые элементы (`append`), удалять (`pop`), выводить по индексу. Пишем программу, которая ищет элемент в списке.

Занятие 15. Функции (def).

Пишем свою первую функцию приветствия. Создаем функцию, которая принимает два числа и возвращает их сумму. Понимаем, зачем нужны функции — чтобы не писать один код дважды.

Занятие 16. Итоговый проект — Викторина.

Пишем консольную викторину: 5 вопросов с вариантами ответов. Используем списки для хранения вопросов и ответов. Подсчет правильных ответов. Вывод результата в конце. Защита проекта.

Ноябрь. Модуль 3: VK-боты

Занятие 17. Знакомство с API и библиотекой vk_api.

Объясняем, что такое API (Application Programming Interface) и как программы общаются с внешними сервисами. Рассматриваем социальную сеть ВКонтакте как платформу для разработки ботов. Устанавливаем библиотеку `vk_api` через `pip`. Изучаем документацию. Создаем токен доступа сообщества ВКонтакте (раздел «Управление» → «Работа с API» → «Создать ключ»). Пишем простой скрипт для подключения к API и вывода информации о последних сообщениях.

Занятие 18. Регистрация бота. Эхо-бот.

Создаем сообщество ВКонтакте, настраиваем Long Poll API (включаем «Входящие сообщения», выбираем версию API). Записываем токен сообщества в `config.py`. Пишем бота, который слушает Long Poll и отвечает на любое сообщение тем же текстом. Настраиваем обработку события `message_new`. Запускаем и проверяем в диалоге с сообществом.

Занятие 19. Обработка команд и состояния.

Добавляем обработку команд (например, "привет", "помощь", "пока"). Используем условную логику: если текст сообщения равен "привет", бот отвечает приветствием. Изучаем, как хранить состояние пользователя в словаре Python для многошаговых сценариев. Добавляем простую клавиатуру с кнопками через `Keyboard` из библиотеки `vkbot` или стандартную клавиатуру ВКонтакте.

Занятие 20. Бот-калькулятор.

Создаем бота, который принимает текст сообщения (например, «2+2»), вычисляет результат и отправляет ответ. Используем библиотеку `vk_api`, получаем текст сообщения, обрабатываем его через `eval()` или парсинг строки. Добавляем обработку ошибок, чтобы бот не падал при вводе букв. Реализуем с помощью условной логики и функций.

Занятие 21. Клавиатуры и меню.

Создаем меню с кнопками. Изучаем параметры клавиатур: `one_time` (одноразовая), `inline` (встроенная в сообщение). Добавляем кнопки с текстом, ссылками и `callback-`запросами. Обрабатываем нажатия на кнопки. Реализуем интерактивный выбор: бот спрашивает, какую информацию показать (погода, курс валют), и по нажатию кнопки отвечает.

Занятие 22. Бот-викторина с хранением состояния.

Создаем бота, который задает вопросы по очереди. Используем словарь для хранения состояния пользователя: текущий вопрос, количество правильных ответов. При каждом ответе проверяем, сохраняем баллы и переходим к следующему вопросу. В конце выводим результат. Обрабатываем команды запуска викторины и сброса прогресса.

Занятие 23. Отправка картинок и медиафайлов.

Изучаем загрузку медиафайлов через `VkUpload` и отправку их в сообщениях. Бот по команде `/theme` присылает случайную картинку из папки с проектом. Используем `photo_messages` для загрузки фотографий в сообщения.

Занятие 24. Деплой (запуск 24/7).

Регистрируемся на `PythonAnywhere`. Загружаем код бота через `GitHub` или загрузчик файлов. Устанавливаем библиотеки `vk_api`, `requests` и другие. Настраиваем `Always-on task` для круглосуточной работы. Бот работает даже при выключенном компьютере. Проверяем, что он отвечает в диалоге с сообществом.

Декабрь. Модуль 4: Веб-разработка (HTML + CSS)

Занятие 25. HTML — скелет страницы.

Создаем `index.html`. Изучаем основные теги: `html`, `head`, `body`, `h1`, `p`, `div`. Открываем страницу в браузере. Понимаем структуру любого сайта. Сохраняем первую страницу.

Занятие 26. Ссылки и изображения.

Вставляем картинку через тег `img`. Делаем ссылки через тег `a`. Учимся делать картинку-ссылку. Атрибуты `src`, `href`, `alt`. Создаем страницу с галереей и переходами на другие сайты.

Занятие 27. CSS — цвета и шрифты.

Добавляем внутренние стили в тег `style`. Меняем цвет текста, размер шрифта, тип шрифта. Подключаем шрифты из `Google Fonts`. Учимся делать фон страницы.

Занятие 28. CSS — рамки и отступы.

Изучаем блочную модель (`Box Model`). Понятия `margin` (внешний отступ), `padding` (внутренний), `border` (рамка). Делаем карточку товара с красивой рамкой и тенями. Понимаем, как браузер вычисляет размеры.

Занятие 29. Flexbox — выравнивание.

Учимся располагать элементы в ряд через `display: flex`. Свойства `justify-content`, `align-items`.

Создаем сетку из трех карточек. Добавляем адаптивность — при сужении экрана карточки переносятся.

Занятие 30. Навигация и hover.

Создаем меню сайта. Добавляем эффекты при наведении мыши через псевдокласс :hover. Меняем цвет фона, добавляем плавные переходы (transition). Меню становится интерактивным.

Занятие 31. Формы (input).

Создаем форму обратной связи: поле для имени, email, сообщения. Добавляем кнопку отправки. Используем атрибуты placeholder, required. Понимаем разницу между GET и POST.

Занятие 32. Публикация на GitHub Pages.

Регистрируемся на GitHub. Создаем репозиторий username.github.io. Загружаем наш сайт. Включаем GitHub Pages в настройках. Получаем живую ссылку на сайт в интернете. Радует первому опубликованному сайту.

Январь. Модуль 5: JavaScript (интерактивность)

Занятие 33. Подключение JS. Alert и Prompt.

Вставляем тег script в HTML. Пишем всплывающее окно alert(). Спрашиваем имя через prompt(). Выводим ответ в консоль (console.log). Учимся открывать консоль браузера для отладки.

Занятие 34. События (onclick).

Делаем кнопку, которая реагирует на клик. Создаем счетчик кликов. Изменяем текст на странице через innerHTML. Понимаем, как JS взаимодействует с HTML-элементами.

Занятие 35. Работа с DOM. Изменение стилей.

Учимся находить элементы по id через getElementById. Меняем стили через свойство style. Делаем кнопку переключения темной/светлой темы (изменяем фон и цвет текста).

Занятие 36. Условия (if) в JS.

Проверка пароля в браузере: поле ввода и кнопка. Если пароль правильный — доступ открыт, иначе — ошибка. Синтаксис JS: фигурные скобки, круглые скобки, точка с запятой.

Занятие 37. Циклы в JS.

Динамически создаем список элементов на странице через цикл for. Добавляем новые элементы через createElement и appendChild. Учимся работать с массивами в JS.

Занятие 38. Таймеры (setInterval).

Делаем секундомер на странице. Каждую секунду увеличиваем счетчик. Добавляем кнопки Старт, Стоп, Сброс (clearInterval). Понимаем, как работают асинхронные события.

Занятие 39. Массивы объектов. Галерея.

Создаем массив объектов с картинками. Каждый объект содержит src и название. Делаем галерею с кнопками «Вперед» и «Назад». Добавляем автопрокрутку через setInterval.

Занятие 40. Итоговый проект — Интерактивная страница.

Собираем все знания: страница с выбором цвета фона, секундомером, галереей и To-Do списком. Каждый элемент должен быть интерактивным. Защита проекта.

Февраль. Модуль 6: Canvas и игры

Занятие 41. Canvas — рисование фигур.

Создаем тег canvas в HTML. Учимся рисовать прямоугольники, круги, линии через контекст 2d. Рисуем домик: квадрат и треугольник. Понимаем систему координат (0,0 — левый верхний угол).

Занятие 42. Анимация (requestAnimationFrame).

Мяч летает по экрану и отскакивает от стен. Используем requestAnimationFrame для плавной анимации. Управляем скоростью через переменные dx, dy. Мяч меняет цвет при касании стены.

Занятие 43. Управление клавишами.

Ловим нажатия клавиш через событие keydown. Управляем квадратом стрелками. Добавляем ограничение — квадрат не выходит за границы. Ускорение при зажатой клавише Shift.

Занятие 44. Столкновения (Hitbox).

Реализуем проверку пересечения двух прямоугольников. Игрок собирает яблоки, счет увеличивается. При столкновении с врагом — теряем жизнь. Понимаем геометрию столкновений.

Занятие 45. Игра «Змейка» (логика).

Создаем массив для хранения сегментов змейки. Голова двигается, хвост подтягивается. Реализуем движение через setInterval. Еда появляется в случайном месте.

Занятие 46. «Змейка» — финал.

Добавляем рост змейки при поедании еды. Проверка столкновения с собственным хвостом — Game Over. Добавляем счет, рекорд (localStorage), кнопку рестарта. Полноценная работающая игра.

Занятие 47. Игра «Арканоид» (кирпичи).

Создаем ракетку, мяч и массив кирпичей. Мяч отскакивает от ракетки и стен. При касании кирпича — он разрушается, счет растет. Кирпичи имеют разную прочность и цвет.

Занятие 48. Презентация игр.

Каждый показывает свою игру (Змейка или Арканоид). Демонстрируют код, объясняют сложные моменты. Играем в игры друг друга. Защита проекта.

Март. Модуль 7: Мобильная разработка**Занятие 49. Введение в MIT App Inventor.**

Заходим на ai2.appinventor.mit.edu. Изучаем интерфейс: палитра компонентов, дизайнер, блоки. Создаем приложение «Привет мир» — кнопка меняет текст. Устанавливаем AI2 Companion на телефон и подключаем.

Занятие 50. Переключение экранов.

Создаем два экрана (Screen1 и Screen2). Кнопка «Далее» открывает второй экран. Передаем текст из одного экрана в другой через глобальные переменные. Понимание навигации в мобильных приложениях.

Занятие 51. Списки (ListView).

Выводим список имен на экран. Добавляем поле ввода и кнопку, чтобы добавлять новые элементы в список. Учимся работать с динамическими данными в мобильном приложении.

Занятие 52. Хранение данных (TinyDB).

Приложение запоминает имя пользователя после перезапуска. Используем TinyDB — локальное хранилище ключ-значение. Сохраняем настройки (тему, язык). Приложение «запоминает» пользователя.

Занятие 53. Сенсоры (акселерометр).

При тряске телефона меняется цвет экрана. Используем компонент Accelerometer. Делаем «уровень» — пузырек в центре экрана, который двигается при наклоне телефона. Работа с физическими датчиками.

Занятие 54. То-до лист (задачи).

Создаем полноценное приложение для списка дел. Добавляем задачи, отмечаем выполненные, удаляем свайпом. Используем TinyDB для сохранения списка. Приложение готово к использованию.

Занятие 55. Сборка APK.

Собираем приложение в установочный файл APK. Скачиваем на телефон. Устанавливаем и тестируем. Добавляем иконку приложения. Приложение готово к публикации.

Занятие 56. Презентация мобильных приложений.

Каждый показывает свое приложение на телефоне. Демонстрация функционала. Обсуждение сложностей. Защита мобильного проекта.

Апрель. Модуль 8: Хакатон и командная разработка

Занятие 57. Введение в Git и GitHub.

Регистрируемся на GitHub. Создаем первый репозиторий. Учимся делать коммиты и пушить. Скачиваем GitHub Desktop для удобства. Понимаем систему контроля версий.

Занятие 58. Работа с ветками (branches).

Создаем ветку для нового функционала. Делаем изменения и сливаем ветку с основной (merge). Работаем в парах — один пишет код, второй делает код-ревью. Понимаем командную разработку.

Занятие 59. Выбор финального проекта.

Каждый выбирает проект на основе изученного: веб-сайт, VK-бот, браузерная игра или мобильное приложение. Пишем техническое задание: что будет делать проект, какие технологии используем. Старшие помогают младшим с выбором.

Занятие 60. Разработка (день 1).

Пишем код проекта. Младшие работают над базовой функциональностью. Старшие берут сложные задачи — базы данных, авторизация, работа с API. Преподаватель консультирует каждого. Старшие помогают младшим.

Занятие 61. Разработка (день 2).

Продолжаем писать код. Добавляем дизайн и интерфейс. Тестируем промежуточные версии. Исправляем баги. Приводим код в порядок (рефакторинг). Готовимся к финальной сдаче.

Занятие 62. Тестирование и отладка.

Меняемся проектами с соседом. Каждый ищет баги в чужом проекте. Составляем список найденных ошибок. Возвращаем проекты и исправляем их. Проводим финальное тестирование.

Занятие 63. Оформление проекта и ReadMe.

Создаем файл ReadMe.md на GitHub. Описываем проект: что делает, как установить, как использовать. Добавляем скриншоты. Оформляем портфолио проекта. Готовим презентацию.

Занятие 64. Предзащита проектов.

Репетируем выступление перед группой. Каждый рассказывает о своем проекте 5 минут. Получаем критику и рекомендации. Дорабатываем презентации. Готовимся к родительскому дню.

Май. Модуль 9: Финал и Demo Day

Занятие 65. Полировка проектов.

Исправляем последние ошибки. Улучшаем внешний вид. Добавляем финальные штрихи —

анимации, переходы, звуки. Проверяем работу на разных устройствах. Все проекты готовы к показу.

Занятие 66. Подготовка портфолио.

Собираем ссылки на все проекты: игры на Scratch, боты в VK, сайты на GitHub, приложения на Android. Создаем единую страницу-портфолио со ссылками. Старшие составляют IT-резюме.

Занятие 67. Профориентация.

Рассказываем о профессиях в IT: фронтенд, бэкенд, мобильный разработчик, QA-тестировщик, аналитик, DevOps. Обсуждаем, какие навыки нужны для каждой специальности. Помогаем определиться с направлением.

Занятие 68. Генеральная репетиция.

Полная репетиция Demo Day. Каждый выступает с проектом 5–7 минут. Отрабатываем ответы на вопросы. Улучшаем подачу материала. Убираем волнение перед выступлением.

Занятие 69. Demo Day. Часть 1.

Приглашаем родителей и администрацию. Первая группа показывает проекты. Каждый демонстрирует код и готовый продукт. Ответы на вопросы. Вручение сертификатов за модули.

Занятие 70. Demo Day. Часть 2.

Вторая группа показывает проекты. Финальное выступление. Подведение итогов. Награждение лучших в номинациях: «Самый сложный код», «Лучший дизайн», «Самый полезный проект».

Занятие 71. Вручение сертификатов.

Торжественное вручение сертификатов об окончании курса. В сертификате указаны все освоенные технологии: Scratch, Python, VK Bot, HTML/CSS/JS, Canvas, MIT App Inventor, Git. Фотосессия.

Занятие 72. Итоговое занятие. Чаепитие.

Обсуждаем прошедший год: что было самым сложным, что самым интересным. Строим планы на следующий год. Старшие получают рекомендательные письма. Завершение курса.

4. Воспитательный модуль

Дополнительное образование сегодня – это необходимое звено в воспитании многогранной личности, органично и естественно сочетающее в себе процессы обучения ребенка конкретной прикладной деятельности и разностороннего воспитания. Организации дополнительного образования обладают существенным воспитательным потенциалом и благоприятными условиями для поддержки творческих устремлений детей в самопознании, самоопределении, самореализации и самоутверждении.

Общей целью воспитания в МОУ ДО «Центр дополнительного образования детей Дзержинского района» является приобщение обучающихся к российским традиционным духовно-нравственным ценностям, правилам и нормам поведения в российском обществе, а также создание условий для гармоничного вхождения обучающихся в социальную и профессиональную среды.

Достижению поставленной общей цели воспитания будут служить следующие задачи:

- формировать у обучающихся духовно-нравственные, гражданско-правовые ценности, чувство причастности и уважительного отношения к историко-культурному и природному наследию России и малой родины;
- формировать у обучающихся внутреннюю позицию личности по отношению к окружающей социальной действительности;
- формировать мотивацию к профессиональному самоопределению обучающихся, приобщению к социально-значимой деятельности для осмысленного выбора профессии;
- воспитывать у обучающихся ответственное отношение к результатам своего труда, аккуратность и внимательность при решении поставленных задач;
- формировать навыки работы в команде, умение конструктивно взаимодействовать с другими участниками образовательного процесса, уважать чужое мнение и находить компромиссы.

Воспитательная работа в творческом объединении «Программирование для начинающих: от Scratch до профессиональной разработки» осуществляется по нескольким направлениям деятельности (духовно-нравственное, гражданско-патриотическое, экологическое, профориентационное), позволяющим охватить и развить все аспекты личности обучающихся.

Воспитательная деятельность, органично вплетенная в процесс обучения, позволяет суммировать полученные знания, умения, навыки и ориентировать личность ребенка на творческое саморазвитие и нравственное самосовершенствование.

Система общих воспитательных дел и мероприятий включает в себя:

- **Массовые мероприятия учебного характера:** итоговые и отчетные занятия, открытые уроки, защиты проектов, участие в конкурсах, хакатонах и олимпиадах различного уровня.
- **Массовые мероприятия воспитательно-развивающего характера:** тематические праздники, календарные мероприятия, дни рождения творческого объединения, посвящение в программисты, новогодние встречи.
- **Социальные акции и проекты:** создание социально значимых цифровых продуктов (приложения для школы, сайты для общественных организаций, цифровые поздравительные открытки).

- **Профориентационные мероприятия:** дни открытых дверей в вузах и колледжах, встречи с выпускниками, работающими в сфере ИТ, мастер-классы от приглашенных специалистов, участие в профориентационных фестивалях.

Календарный план воспитательной работы

№ п/п	Название мероприятия	Дата проведения	Место проведения
1	Инструктажи о правилах поведения и технике безопасности	сентябрь, январь	МОУ ДО «Центр дополнительного образования детей Дзержинского района»
2	Участие в массовых мероприятиях разного уровня	согласно графику	МОУ ДО «Центр дополнительного образования детей Дзержинского района»
3	Участие в создании поздравлений к праздникам (День учителя, День матери, Новый год, 23 февраля, 8 марта, 9 мая и т.д.)	согласно календарю	МОУ ДО «Центр дополнительного образования детей Дзержинского района»
4	Проведение в творческом объединении воспитательных мероприятий (День матери, 8 марта, 23 февраля, Новогодние мероприятия и т.д.)	согласно календарю	МОУ ДО «Центр дополнительного образования детей Дзержинского района»
5	Организация открытых занятий обучающихся	май	МОУ ДО «Центр дополнительного образования детей Дзержинского района»

5. Ожидаемые результаты

В результате обучения по программе «Программирование для начинающих: от Scratch до профессиональной разработки» обучающиеся:

будут знать:

1. правила безопасной работы на персональном компьютере и требования к организации рабочего места;
2. основные понятия алгоритмизации: алгоритм, цикл, условие, переменная, функция, массив;
3. базовый синтаксис языка Python: типы данных, условные конструкции, циклы, функции, списки, работа с файлами;
4. основы работы с библиотеками и API: установка библиотек через pip, получение данных из интернета, работа с JSON;
5. принципы создания VK-ботов: регистрация токена, обработка команд и сообщений, работа с клавиатурами и callback-запросами;
6. структуру веб-страницы: HTML-теги, CSS-стили, основы JavaScript;
7. основы работы с системой контроля версий Git и платформой GitHub;
8. принципы работы с графическим интерфейсом в среде Scratch: спрайты, сцены, блоки, переменные, клоны;
9. основы мобильной разработки в среде MIT App Inventor: интерфейс, экраны, компоненты, TinyDB;
10. форматы файлов и способы экспорта проектов для различных платформ;
11. современные IT-профессии и требования к специалистам в области разработки программного обеспечения;
12. основы командной разработки: распределение ролей, планирование задач, совместное кодирование.

будут уметь:

1. использовать персональный компьютер как инструмент для обучения и создания программных продуктов;
2. создавать игровые проекты и анимации в среде Scratch с использованием циклов, условий, переменных и клонов;
3. писать консольные программы на Python: ввод и вывод данных, арифметические операции, условные конструкции, циклы, функции;
4. разрабатывать VK-ботов с интерактивным меню, обработкой команд, отправкой медиафайлов и хранением состояния пользователя;
5. создавать веб-страницы с использованием HTML и CSS: заголовки, абзацы, ссылки, изображения, формы, Flexbox-верстка;
6. добавлять интерактивность на веб-страницы с помощью JavaScript: события, изменение стилей, таймеры, анимация;
7. разрабатывать браузерные игры с использованием Canvas: рисование фигур, анимация, управление клавишами, обработка столкновений;
8. создавать простые мобильные приложения для Android в среде MIT App Inventor с навигацией между экранами и хранением данных;
9. работать с системой контроля версий Git: создание репозитория, коммиты, ветки, публикация на GitHub;

10. строить различные алгоритмические конструкции и применять их для решения практических задач;
11. создавать комплексные проекты, объединяющие несколько технологий (например, бот + сайт, игра + приложение);
12. работать в команде, распределять роли и задачи, презентовать свои проекты, защищать и отстаивать свои идеи, аргументированно вести диалог с педагогом и сверстниками.

6. Обеспечение программы

6.1 Методическое обеспечение

Методы и приёмы обучения:

1. **Словесный метод обучения** (лекция, рассказ, объяснение, проведение бесед и бесед-объяснений, инструктаж, дискуссия, обсуждение проектов);
2. **Использование наглядных средств** (демонстрация презентаций, видеофильмов, изображений и схем, работа с готовыми программными кодами и проектами, демонстрация работы готовых приложений и игр);
3. **Выполнение устных и практических упражнений** (упражнение, практикум, проектирование, планирование и проведение эксперимента, решение поисковых задач, написание кода, отладка программ);
4. **Метод проектов** (создание индивидуальных и групповых проектов: игры, боты, сайты, приложения, презентация и защита проектов);
5. **Игровой метод** (использование игровых сред Scratch, Construct 3, игровых механик для обучения, соревновательные элементы в виде хакатонов);
6. **Метод «перевернутого класса»** (предварительное ознакомление с темой, затем практическое закрепление на занятии);
7. **Парное обучение и наставничество** (старшие помогают младшим, сильные помогают слабым).

Структура занятия:

Занятие с детьми строится по следующему принципу:

- **В начале занятия** обсуждается самостоятельная работа дома, обсуждение трудностей, которые возникли при выполнении домашнего задания, проработка основных ошибок, которые могли возникнуть при выполнении работы, мотивация обучающегося на выполнение домашних заданий и самостоятельных работ;
- **Затем сообщается новая тема занятия**, объясняется связь с предыдущими темами, формулируется цель занятия с привлечением обучающихся, обсуждаются методы, приёмы и инструменты;
- **Объясняется новая тема** с учётом мотивации и поддержки интереса к предмету и разъясняется задание, которое будет выполняться на занятии. Теоретический материал подается блоками по 10-15 минут с обязательной сменой деятельности;
- **Обязательно используются на занятии демонстрационные материалы**, презентации, работа с готовыми программными кодами, демонстрация работы проектов на проекторе;
- **После объяснения учащиеся приступают к выполнению заданий**, которые выполняют совместно с педагогом (код-сессия) или самостоятельно. Задания даются от простого к сложному. Осуществляется дифференцированный подход: младшие выполняют базовый уровень, старшие — Pro-уровень с дополнительными задачами;
- **Педагог активно помогает учащимся**, которые столкнулись с трудностями в практической работе, отслеживает правильность выполнения заданий, проводит индивидуальные консультации. Старшие обучающиеся привлекаются в качестве ассистентов для помощи младшим;
- **В конце занятия подводятся итоги**, обобщается изученный на занятии материал и находится его связь с ранее полученными знаниями. Проводится рефлексия: что получилось, что было сложным;

- **Объясняется домашнее задание**, раскрывается его суть, как сделать, приводятся примеры и аналоги с текущего занятия. Домашнее задание носит закрепляющий характер и занимает не более 15-20 минут;
- **Рефлексия учащихся по итогам занятия**, выставляются объективные оценки, аргументируются педагогом. Оценка выставляется за активность, качество выполнения заданий, творческий подход и самостоятельность.

6.2. Материально-техническое обеспечение

Техническое оснащение компьютерного класса:

1. IBM-совместимые персональные компьютеры (или ноутбуки) из расчета один ученик на один компьютер (оптимально 10-12 рабочих мест);
2. Компьютеры должны быть объединены в локальную сеть;
3. На компьютерах должна быть возможность выхода в Интернет с достаточной пропускной способностью (от 20 Мбит/с);
4. Проектор или интерактивная доска для демонстрации материала педагогом.

Программное обеспечение (лицензионное или свободно распространяемое):

1. Операционная система Windows 10/11;
2. PyCharm Community Edition — бесплатная версия;
3. Visual Studio Code — бесплатный редактор кода;
4. Git и GitHub Desktop — бесплатное ПО;
5. Браузер Google Chrome или Mozilla Firefox (бесплатные);
6. Онлайн-платформа Scratch (scratch.mit.edu) — доступ через браузер;
7. Онлайн-платформа MIT App Inventor (appinventor.mit.edu) — доступ через браузер;
8. Онлайн-платформа Construct 3 (бесплатный тариф) — доступ через браузер;
9. VK (для тестирования ботов) — бесплатное ПО;
10. PythonAnywhere (бесплатный тариф для деплоя ботов);
11. GitHub Pages (бесплатный хостинг для сайтов);
12. Офисный пакет (Microsoft Office или LibreOffice) для оформления документации.

7. Контрольно-измерительные материалы

Поскольку образовательная деятельность в системе дополнительного образования предполагает не только обучение детей определенным знаниям, умениям и навыкам, но и развитие многообразных личностных качеств обучающихся, постольку о ее результатах необходимо судить по двум группам показателей:

- **учебным** (фиксирующим предметные и учебные знания, умения, навыки, приобретенные ребенком в процессе освоения образовательной программы);
- **личностным** (выражающим изменения личностных качеств ребенка, развитие социально значимых качеств личности, уровень общего развития и уровень развития коммуникативных способностей под влиянием занятий в данном объединении).

Оценка качества освоения дополнительной образовательной программы является неотъемлемой частью системы дополнительного образования и имеет целью повышение его результативности, а также уровня профессионализма педагогических работников.

1. **Начальный контроль** проводится на вводном занятии курса с целью оценки имеющихся знаний и умений обучающихся перед прохождением данной программы. Формы: собеседование, выполнение простого диагностического задания, анкетирование для выявления уровня интереса к программированию.

2. **Текущий контроль** проводится в течение всего периода реализации программы в различных формах: оценка активности на занятии, выполнение домашней работы, проверка написанного кода, опрос, мини-тесты по пройденному материалу, проверка промежуточных этапов проектов.

3. **Промежуточный контроль** проводится по окончании изучения очередного модуля в форме беседы, защиты проекта, презентации созданного продукта, выполнения проверочной работы по пройденному материалу, участия в мини-хакатоне.

4. **Итоговый контроль** проводится посредством выполнения итоговой работы на выбранную самостоятельно тему, созданием презентации, защитой проекта перед другими учащимися и родителями на Demo Day.

Задачи мониторинга освоения дополнительной образовательной программы

- определение уровня образовательной подготовки обучающихся в области программирования;
- выявление степени сформированности умений и навыков детей в выбранном виде деятельности;
- анализ полноты реализации образовательной программы детского объединения;
- соотнесение прогнозируемых и реальных результатов освоения образовательной программы;
- создание условий для внесения необходимых корректив в ход и содержание образовательного процесса в детских объединениях.

В образовательном процессе детского объединения мониторинг освоения дополнительной образовательной программы выполняет ряд функций:

- **обучающую**, так как создает дополнительные условия для повышения уровня обобщения и осмысления ребенком полученных теоретических и практических знаний, умений и навыков;
- **воспитательную**, так как является условием расширения познавательных интересов и потребностей ребенка;

- **развивающую**, так как создает условия для осознания обучающимся их актуального развития и определения перспектив дальнейшего развития;

- **социально-психологическую**, так как создает условия для обучающихся пережить «ситуацию успеха».

Формы мониторинга освоения дополнительной образовательной программы

- текущие мероприятия городского, регионального, российского уровня: конкурсы по программированию, хакатоны, олимпиады по информатике, фестивали технического творчества;

- промежуточные итоговые мероприятия: презентации проектов, открытые занятия;

- итоговое занятие по модулям и за год: защита творческих проектов, Demo Day.

Критерии определения уровня освоения обучающимися дополнительной образовательной программы

Критерии оценки уровня теоретической подготовки обучающихся:

- соответствие уровня теоретических знаний программным требованиям;
- осмысленность и свобода использования специальной терминологии (алгоритм, переменная, цикл, условие, функция, массив, API, Git и др.);

- понимание принципов работы изучаемых технологий.

Критерии оценки уровня практической подготовки обучающихся:

- соответствие практических умений и навыков программным требованиям;
- отсутствие затруднений в использовании программного обеспечения и сред разработки;

- умение писать и отлаживать код, находить и исправлять ошибки;
- креативность в выполнении творческих заданий и проектов;
- качество и функциональность созданных программных продуктов.

Критерии оценки уровня личностного развития обучающихся:

- мотивация учебно-познавательной деятельности;
- сформированность интеллектуальных умений (анализ, синтез, логическое мышление);

- навыки учебного труда (самоорганизация, планирование времени);
- результативность индивидуальных занятий;
- уровень утомляемости;
- целеустремлённость;
- дисциплина и организованность;
- коммуникабельность, умение работать в команде;
- уровень этической культуры;
- исполнение обязанностей в детском объединении.

Технологическая карта оценки уровня освоения программы

Пример технологической карты по модулям:

Модуль	Критерии оценки	Низкий уровень	Средний уровень	Высокий уровень
Блочное программирование (Scratch)	Работа с интерфейсом, создание	Испытывает трудности при работе с	Уверенно работает с основными	Самостоятельно создает сложную игру с

	алгоритмов, использовани е циклов, условий, переменных, клонов, создание игры	интерфейсом, не может самостоятельно создать алгоритм	блоками, может создать простую игру с помощью педагога	несколькими механиками, использует продвинутое возможности
Основы Python	Работа с переменными, условиями, циклами, функциями, списками, написание консольных программ	Знает базовый синтаксис, но не может самостоятельно написать программу без помощи	Уверенно пишет программы, используя основные конструкции, допускает незначительные ошибки	Самостоятельно пишет сложные программы, использует продвинутое конструкции, оптимизирует код
VK-боты	Создание бота, обработка команд, работа с клавиатурами, API, деплой	Не может самостоятельно зарегистрировать бота и написать базовый код	Создает простого бота с базовыми командами, требует помощи при деплое	Самостоятельно создает и деплоит бота со сложным функционалом и взаимодействием с API
Веб-разработка (HTML+CSS)	Создание HTML-страниц, CSS-стилирование, верстка, Flexbox, публикация	Знает основные теги, но не может создать структурированную страницу	Самостоятельно верстает страницы с использованием CSS и Flexbox	Создает адаптивные сайты с продвинутой версткой и дизайном
JavaScript	Работа с событиями, DOM, таймерами, создание интерактивных элементов	Понимает базовые конструкции, но не может реализовать интерактивность	Уверенно добавляет интерактивность на страницу с помощью JS	Создает сложные интерактивные приложения с анимацией
Canvas и игры	Рисование на Canvas, анимация, управление клавишами, создание игр	Может нарисовать простые фигуры, но не создает игру	Самостоятельно создает игры с базовой логикой (Змейка)	Создает сложные игры с продуманной логикой и интерфейсом
Мобильная	Создание	Испытывает	Самостоятельно	Создает

разработка	приложений в MIT App Inventor, навигация, хранение данных	трудности с созданием интерфейса и логики	о создает простое приложение с навигацией	полноценное приложение с хранением данных и сенсорами
Хакатон и командная разработка	Работа в команде, использование Git, планирование, презентация	Не может работать в команде, конфликтует, не понимает Git	Работает в команде, выполняет свои задачи, использует Git на базовом уровне	Лидирует в команде, активно использует Git, качественно презентует результат

Оценка итоговых проектов

Итоговый проект оценивается по следующим критериям:

Критерий	Максимальный балл	Пояснение
Функциональность проекта	5	Программа работает без ошибок, реализованы все заявленные функции
Качество кода	5	Код читаемый, структурированный, есть комментарии, нет дублирования
Сложность проекта	5	Использованы продвинутое технологии, сложные алгоритмы, интеграция с внешними сервисами
Дизайн и интерфейс	5	Проект имеет удобный и эстетичный интерфейс, продумана логика взаимодействия
Презентация проекта	5	Уверенное выступление, четкое объяснение кода и механик, ответы на вопросы
Итого:	25	

Уровни освоения программы:

- **Высокий уровень** (90-100% от максимального балла): обучающийся самостоятельно создает сложные проекты, использует продвинутое технологии, активно участвует в командной работе.
- **Средний уровень** (60-89% от максимального балла): обучающийся выполняет базовые задания, создает проекты с помощью педагога, усваивает основной материал.
- **Низкий уровень** (менее 60% от максимального балла): обучающийся испытывает трудности при выполнении заданий, требуется постоянная помощь педагога.

8. Список информационных источников

Список нормативно-правовых документов:

1. Концепция развития дополнительного образования детей до 2030 года, утв. распоряжением Правительства РФ от 31 марта 2022 года № 678-р.
2. Концепция персонифицированного дополнительного образования детей в Ярославской области, утв. постановлением Правительства области от 17.07.2018 года № 527-п.
3. Письмо Минобрнауки России от 18.11.2015 №09-3242 «О направлении информации» (вместе с «Методическими рекомендациями по проектированию дополнительных общеразвивающих программ (включая разноуровневые программы)
4. Положение о персонифицированном дополнительном образовании детей в городе Ярославле, утв. постановлением мэрии города Ярославля от 11.04.2019 года № 428.
5. Постановление Главного государственного санитарного врача РФ от 28.09.2020 № 28 «Об утверждении СП 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи».
6. Постановление Главного государственного санитарного врача РФ 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».
7. Правила персонифицированного финансирования дополнительного образования детей в Ярославской области, утв. приказом департамента образования Ярославской области от 07.08.2018 года № 19-нп.
8. Приказ Минпросвещения России от 27.07.2022 № 629 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам» (Зарегистрировано в Минюсте России 26.09.2022 № 70226).
9. Приказ Министерства труда и социальной защиты РФ от 05 мая 2018 г. № 298н 2 «Об утверждении профессионального стандарта «Педагог дополнительного образования детей и взрослых».
10. Приказ Минпросвещения России от 17.03.2025 № 209 «Об утверждении федеральной программы воспитательной работы для организаций отдыха детей и их оздоровления и календарного плана воспитательной работы».
11. Приказ Минпросвещения РФ от 11.02.2023 № 75 «Об утверждении Порядка организации отдыха детей».
12. Сборник нормативно-правовых и информационно-методических материалов по организации внутреннего контроля образовательной деятельности в учреждениях дополнительного образования муниципальной системы образования г. Ярославля [Текст] /од редакцией Е.Г. Абрамовой, И.В. Лаврентьевой. – Ярославль: МОУ ДО Детский центр «Восхождение», 2017. – 44 с.
13. Указ Президента Российской Федерации от 21.07.2020 № 474 «О национальных целях развития Российской Федерации на период до 2030 года».
14. Указ Президента Российской Федерации от 02.07.2021 № 400 «О стратегии национальной безопасности Российской Федерации».
15. Указ Президента Российской Федерации от 09.11.2022 № 809 «Об утверждении Основ государственной политики по сохранению и укреплению традиционных российских духовно-нравственных ценностей».

16. Федеральный закон «Об образовании в Российской Федерации» № 273-ФЗ от 29.12.2012 года.
17. Федеральный закон от 24.07.1998 № 124-ФЗ «Об основных гарантиях прав ребёнка в Российской Федерации».
18. Федеральный закон от 28.12.2024 № 543-ФЗ «О внесении изменений в Федеральный закон «Об основных гарантиях прав ребёнка в Российской Федерации».

Список электронных ресурсов:

1. Скретч 3.0 | Кулибин — цифровая образовательная среда [Электронный ресурс] // URL: <https://scratch.kulibin.app> (дата обращения: 19.08.2026).
2. Справочник по HTML [Электронный ресурс] // URL: <https://htmlbook.ru/html> (дата обращения: 19.08.2026).
3. Справочник по CSS [Электронный ресурс] // URL: <https://htmlbook.ru/CSS> (дата обращения: 19.08.2026).
4. JavaScript - MDN Web Docs - Mozilla [Электронный ресурс] // URL: <https://developer.mozilla.org/ru/docs/Web/JavaScript> (дата обращения: 19.08.2026)
5. Kotlin Docs | Kotlin Documentation [Электронный ресурс] // URL: <https://kotlinlang.org/docs/home.html> (дата обращения: 19.08.2026)
6. The MIT App Inventor Library: Documentation & Support [Электронный ресурс] // URL: <https://appinventor.mit.edu/explore/library> (дата обращения: 19.08.2026)
7. Python 3.14.7 Documentation [Электронный ресурс] // URL: <https://docs.python.org/3/> (дата обращения: 19.08.2026)