Подписано цифровой подписью: Natalia Leshuk
Дата: 2026.09.06 00:11:21 +03'00'

ЧАСТНОЕ УЧРЕЖДЕНИЕ
ОБЩЕОБРАЗОВАТЕЛЬНАЯ ОРГАНИЗАЦИЯ
«ПЕТРОВСКАЯ ШКОЛА»

Принято
на заседании педагогического совета
протокол №1 от «28» августа 2026 г.

Утверждаю
Генеральный директор
Вяземская Е.К.

Дополнительная общеобразовательная общеразвивающая программа

физкультурно – спортивной направленности

ознакомительного уровня

«Школа программирования KIBERone»

Возраст обучающихся: 10-15 лет

Срок реализации: 1 год

Автор-составитель:
Великий Кирилл Сергеевич,
педагог дополнительного образования

Москва, 2026 г.

Раздел №1. «Комплекс основных характеристик программа»

1.1 Пояснительная записка

Направленность программы – техническая.

Уровень освоения программы – ознакомительный.

Программа ориентирована на формирование у обучающихся первичных навыков алгоритмического мышления и практического программирования через создание компьютерных игр и веб-сайтов.

Актуальность программы

Обусловлена необходимостью раннего формирования цифровой грамотности, развития логического и алгоритмического мышления. Программа позволяет изучать технологии, творчески развиваться через создание игр и приложений.

Адресат программы

Программа рассчитана на обучающихся в возрасте 10-15 лет, без предварительного отбора.

Объем и срок освоения программы

Программа рассчитана на 36 часов в год (1 час в неделю) и реализуется в течение 36 учебных недель

Форма обучения

Форма обучения – очная. Наполняемость учебной группы до 10 человек.

Режим занятий

Занятия проводятся 1 раз в неделю, продолжительность занятий 60 минут.

1.2. Цель и задачи программы

Цель программы - сформировать у обучающихся базовое понимание логики программирования через практическое создание игр и веб-страниц.

Задачи:

Обучающие:

1. Познакомить с понятиями: алгоритм, программа, событие, переменная, цикл, условие, тег, атрибут, свойство CSS.

2. Научить создавать события в визуальной среде Construct 3 (условие → действие).
3. Сформировать навык создания 2D-игры-раннера с препятствиями, счётом и экранами старта/проигрыша.
4. Обучить разработке 2D-платформера в GDevelop: анимация персонажа, враги, монеты, переход между уровнями.
5. Научить писать HTML-код веб-страницы: заголовки, абзацы, списки, изображения, ссылки.
6. Сформировать базовые навыки CSS: цвет, фон, шрифт, отступы, расположение блоков (flexbox).
7. Познакомить с переменными и типами данных в Python (int, str, float).
8. Научить использовать условный оператор if-else и цикл while в консольных программах.
9. Сформировать умение создавать консольные мини-игры («Угадай число», «Камень, ножницы, бумага», текстовый квест).

Развивающие:

1. Развивать алгоритмическое мышление (умение разбивать задачу на последовательные шаги).
2. Развивать навык отладки (поиск и исправление синтаксических и логических ошибок).
3. Развивать пространственное воображение при размещении объектов на игровом поле и блоков на веб-странице.
4. Развивать внимание к деталям (закрытие тегов, правильные отступы, точный синтаксис).
5. Развивать навык работы с инструкцией и справочными материалами (чек-листы, карточки кода).
6. Развивать умение публичной презентации продукта (защита проекта).
7. Развивать коммуникативные навыки (работа в парах, взаимопомощь при отладке).

Воспитательные:

1. Воспитывать ответственное отношение к результатам своего труда (доведение проекта до конца).
2. Воспитывать уважение к интеллектуальной собственности (указание источников спрайтов, звуков, кода).
3. Воспитывать цифровую культуру: соблюдение правил работы за компьютером (осанка, зрительная гимнастика), безопасное поведение в сети.
4. Воспитывать умение конструктивно воспринимать ошибки и критику.
5. Воспитывать взаимопомощь и умение работать в паре/команде.
6. Воспитывать гордость за созданный цифровой продукт и положительную самооценку через публичное признание.

1.3 Содержание программы

Учебный план

№	Название раздела, темы	Количество часов			Формы контроля
		Всего	Теория	Практика	
1.	**Разработка игр на Construct 3** (Клон Geometry Dash)	9	3	6	Опрос
2.	**Продвинутый геймдев в GDevelop** (2D+3D)	**9**	**3**	**6**	Текущий контроль
3.	**Веб-разработка: HTML и CSS** (Интернет-магазин)	**7**	**1**	**6**	Текущий контроль
4.	Основы программирования на Python	**9**	**1**	**8**	Текущий контрол
5	Итоговый контроль	2	-	2	Презентация работ обучающихся
	Итого:	**36**	**8**	**28**	

Содержание учебного плана

Разработка игр на Construct 3

Цель: изучить основы логики игр, работая над клоном Geometry Dash.

Теория (3 ч): Знакомство с интерфейсом Construct 3. Понятие спрайтов, «поведений» (behaviors) и слоев. Логика событий: условие -> действие.

Практика (6 ч): Создание персонажа и препятствий. Реализация постоянного движения. Настройка прыжков. Работа с камерой. Создание уровней, механика «смерти» и перезапуска игры.

Продвинутый геймдев в GDevelop

Цель: Углубление навыков программирования через создание 2D и 3D игр.

Теория (3 ч): Продвинутые события в GDevelop. Переменные (глобальные, объектные). Основы 3D-проекции в движке.

Практика (6 ч):

Часть 1 (2D): Разработка платформера (враги, сбор бонусов, платформы).

Часть 2 (3D): Создание 3D-игры с видом сверху (Top-down shooter или бродилка).

Сайт на HTML и CSS

Цель: освоить структуру сайтов (HTML) и их оформление (CSS).

Теория (3 ч): Структура HTML-документа (теги, заголовки, параграфы, картинки). Понятие стилей CSS (классы, ID, цвета, шрифты). Понятие «блочной модели».

Практика (6 ч): Верстка макета интернет-магазина. Создание структуры каталога товаров. Применение CSS для стилизации товаров, меню и оформления заказа.

Основы программирования на Python

Цель: изучить синтаксис Python через создание консольных приложений.

Теория (3 ч): Знакомство с Python. Переменные и типы данных. Условные конструкции (if/else). Циклы (for/while).

Практика (6 ч): Написание консольных игр (например: «Угадай число», «Виселица»). Обработка ввода пользователя, работа с логикой условий.

1.4 Планируемые результаты

По окончанию освоения программы должны быть достигнуты предметные, метапредметные, личностные результаты.

Предметные результаты (знания и навыки):

В области визуального программирования (Construct 3, GDevelop):

Знание интерфейса игровых движков, понятий «спрайт», «событие», «действие», «поведение» и разницы между 2D и 3D-пространством.

Умение настраивать физику объектов, анимацию, управление персонажем, логику столкновений и механику начисления очков.

В области веб-разработки (HTML/CSS):

Знание структуры веб-страницы, основных тегов разметки и базовых свойств стилизации (цвет, шрифты, отступы).

Умение верстать структурированные страницы по макету и связывать файлы HTML и CSS между собой.

В области текстового программирования (Python):

Знание базового синтаксиса языка Python, типов данных, структуры условий (if-else) и циклов (for, while).

Умение писать, читать и отлаживать консольный код для реализации алгоритмов мини-игр.

Метапредметные результаты:

Алгоритмическое мышление — умение разбивать сложную задачу на последовательность простых шагов (инструкций для компьютера).

Навыки поиска и исправления ошибок (Debugging) — способность анализировать причины сбоев в коде и самостоятельно находить решения.

Проектное мышление — умение планировать этапы создания цифрового продукта от идеи до готовой работающей версии.

Информационная грамотность — навык работы с технической документацией, интерфейсами программ и веб-ресурсами.

Личностные результаты:

Целеустремленность и усидчивость — готовность доводить до конца создание цифрового продукта, преодолевая технические трудности.

Креативность — проявление индивидуальности при создании собственного дизайна уровней игр и интерфейса интернет-магазина.

Профессиональное самоопределение — формирование первичного интереса к ИТ-профессиям (геймдизайнер, веб-разработчик, программист).

Критическое отношение к информации — понимание того, как устроены цифровые продукты изнутри, и переход от пассивного потребления контента к его созданию.

Раздел № 2. Комплекс организационно – педагогических условий»

2.1 Календарный учебный график

Календарный учебный график Календарный учебный график – это составная часть образовательной программы, является приложением к дополнительной общеобразовательной программе.

Четверть	Учебное время	Количество учебных недель	Количество занятий
1	сентябрь - октябрь	8	8
2	ноябрь - декабрь	8	8
3	январь-март	11	11
4	апрель-май	9	9
Всего	Учебных месяцев - 9	36	36

2.2 Условия реализации программы

Материально – техническое обеспечение программы

Учебный кабинет, соответствующий санитарно-эпидемиологическим правилам и нормативам (СанПиН).

Рабочее место преподавателя: персональный компьютер (ПК) с выходом в интернет, проектор или интерактивная панель (демонстрационный экран) для показа кодинга и разбора интерфейсов программ.

Рабочие места обучающихся: персональные компьютеры (или ноутбуки) из расчета 1 компьютер — 1 ученик.

Минимальные системные требования к компьютерам:

Операционная система: Windows 10/11 (64-bit) или macOS 10.15+.

Процессор: Intel Core i3 / AMD Ryzen 3 и выше.

Оперативная память (RAM): не менее 8 ГБ (необходимо для стабильной работы 3D-графики в GDevelop и локальных сред разработки).

Видеокарта: встроенная или дискретная с поддержкой WebGL 2.0.

Периферия: мышь (обязательно для работы в игровых движках) и клавиатура.

Информационное обеспечение программы

для подготовки материалов для занятий используются аудио, видео, фото, интернет-источники.

Кадровое обеспечение

Программу реализует педагог дополнительного образования, имеющий высшее или среднее специальное педагогическое образование, прошедший дополнительное образование в форме курсов повышения квалификации в области дополнительного образования детей физкультурно-спортивной направленности.

2.3 Формы контроля

Контроль знаний и умений учащихся строится на принципах безоценочного (в баллах) или критериального оценивания. Главный акцент делается на создании работающего цифрового продукта (программы, игры, сайта).

Текущий контроль осуществляется в течение каждого раздела,

Проводится на каждом занятии для оценки усвоения текущего материала и своевременной помощи учащимся.

Практический экспресс-контроль: Выполнение пошаговых мини-заданий вслед за преподавателем (написание тега, добавление условия в событие, создание переменной).

Самостоятельные пятиминутки (Кодинг-спринт): Короткие задания на закрепление без помощи учителя.

Пример: «Добавьте на страницу интернет-магазина еще один заголовок <h2> синего цвета» или «Измените скорость кубика в Construct 3 при нажатии на клавишу пробел».

Фронтальный устный опрос (Блиц): Проверка ИТ-терминологии в начале занятия.

Вопросы для блица: что такое спрайт? Чем отличается тег <ul> от ? Для чего нужна переменная? Что произойдет, если в цикле while условие всегда будет верным?

Метод «Найди баг» (Отладка): Преподаватель намеренно выводит на экран код с ошибкой (например, незакрытый тег в HTML или синтаксическая ошибка в if-else на Python), а учащиеся должны найти и исправить её.

Итоговый контроль по разделу (промежуточный контроль)

Проводится в конце каждого из 4-х тематических блоков в форме защиты мини-проекта.

Раздел 1. Контрольный проект «Мой Geometry Dash» (Construct 3)

Форма: Демонстрация уникального уровня игры.

Требования: Персонаж стабильно двигается и прыгает; препятствия (шипы) корректно перезапускают уровень; на экране отображается счетчик попыток; игра имеет финишную черту.

Раздел 2. Контрольный проект «2D-платформер или 3D-лабиринт» (GDevelop)

Форма: Презентация игры на выбор учащегося.

Требования для 2D: Персонаж имеет анимацию бега, собирает монеты (очки меняются на экране), на уровне есть минимум один движущийся враг.

Требования для 3D: Построен замкнутый лабиринт, стены имеют коллизию (персонаж не проходит сквозь них), реализована триггерная зона победы при выходе.

Раздел 3. Контрольный проект «Сайт интернет-магазина» (HTML/CSS)

Форма: Защита веб-страницы.

Требования: Корректная структура HTML (наличие шапки, каталога, формы); подключен внешний файл CSS; оформлена сетка минимум из 4 карточек товаров (фото, цена, кнопка); ссылки в меню меняют цвет при наведении (:hover).

Раздел 4. Контрольный проект «Консольная игра на Python»

Форма: Демонстрация работы программы в терминале.

Требования: Программа использует случайные числа (random); циклы (while или for) ограничивают количество попыток пользователя; добавлены условия проверки (if-elif-else); программа не «вылетает» при вводе некорректных данных.

Итоговый контроль

Форма проведения: Выставка цифровых проектов (Презентация ИТ-портфолио).

Суть: Каждый учащийся собирает свои лучшие проекты за год (2 игры, 1 сайт, 1 программу на Python) в общую папку-портфолио. На итоговом занятии проводится мини-фестиваль, где ученики демонстрируют свои работы группе, преподавателю и приглашенным родителям, запуская проекты на ПК.

2.4 Оценочные материалы

Критерий	Высокий уровень	Базовый уровень	Пониженный уровень
Работоспособность (Техничность)	Проект полностью работает. Ошибок (багов) в коде и логике нет	Проект работает, но изредка встречаются мелкие сбои в логике	Проект запускается, но ключевые функции не работают (персонаж

			не прыгает, сайт «разваливается»)
Самостоятельность	В проект внесены уникальные изменения, выходящие за рамки уроков	Проект сделан строго по инструкции и шаблону преподавателя	Проект сделан частично, код скопирован без понимания сути
Оформление и дизайн	Интерфейс аккуратный, графика подобрана со вкусом, код снабжен отступами	Базовое оформление присутствует, цвета и элементы читаемы	Оформление отсутствует, текст и элементы накладываются друг на друга
Презентация	Ученик уверенно объясняет, как устроен его код и какие логические блоки он использовал	Ученик может запустить проект, но затрудняется объяснить отдельные строки кода	Ученик не может объяснить, как работает его программа

2.5 Методические материалы

Эффективность реализации программы обеспечивается сочетанием современных педагогических технологий, интерактивных методов обучения и практико-ориентированных методических материалов.

Методы обучения

Для развития самостоятельности учащихся и формирования навыка осознанного кодинга (чтобы каждый ученик мог уверенно объяснить структуру своего кода) используются следующие методы:

Практический метод (Метод проектов): Основной метод программы. Обучение строится не вокруг сухой теории, а вокруг создания законченного продукта (игры, сайта, программы). Каждая новая тема — это добавление конкретного функционала в текущий проект.

Словесно-иллюстративный метод (Живой кодинг / Live Coding): Преподаватель не просто показывает слайды, а пишет код и собирает логику игры на экране в реальном времени, комментируя каждый шаг, допуская типичные ошибки и показывая, как их исправлять.

Частично-поисковый и проблемный метод: Преподаватель намеренно создает «проблемную ситуацию» в коде (баг) или предлагает задачу без готового алгоритма (например, *«Как сделать так, чтобы кубик прыгал выше, если долго удерживать кнопку?»*). Это стимулирует учащихся рассуждать и искать логическое решение.

Метод парного программирования и взаимообучения (Peer-to-Peer): Учащиеся с высоким темпом работы выступают в роли «менторов» для тех, у кого возникают трудности. Это развивает метапредметные навыки и заставляет сильных учеников проговаривать логику кода вслух, закрепляя материал.

Метод демонстраций и самопрезентации: Регулярный разбор кода учеников у доски/экрана. Ученик сам комментирует свои логические блоки, отвечая на вопросы: *«За что отвечает этот цикл?»*, *«Почему здесь выбран именно этот тег?»*.

Методические материалы для педагога

Для обеспечения учебного процесса преподаватель использует следующий комплекс материалов:

Поурочные карты-руководства (Скрипты занятий): Подробные конспекты каждого урока с описанием ключевых понятий, скриншотами правильной логики событий (в Construct/GDevelop) и эталонным кодом (HTML/CSS/Python).

Демонстрационная база проектов («Ожидаемый результат»): Коллекция полностью готовых и работающих финальных версий игр, сайтов и программ для демонстрации учащимся на вводных занятиях модулей (в качестве визуальной цели).

Карточки «Архив багов»: Набор дидактических материалов с примерами «сломанного» кода для проведения пятиминуток по отладке (Debugging).

Критериальные матрицы (Рубрикаторы): Таблицы оценки проектов, которые раздаются ученикам *до* начала работы над финальным заданием, чтобы они четко понимали технические требования.

СПИСОК ЛИТЕРАТУРЫ

1. Литература для учащихся

Гриффитс, Д. Изучаем HTML5. Экспресс-курс / Д. Гриффитс. — М.: Питер, 2024. — 144 с. *(Доступное изложение основ верстки в виде комиксов и практических задач).*

Дронов, В. А. HTML5 и CSS3 для начинающих / В. А. Дронов. — СПб.: БХВ-Петербург, 2024. — 192 с. *(Пошаговое руководство по созданию первых веб-страниц).*

Кольцов, Д. М. Python для детей и подростков / Д. М. Кольцов. — СПб.: Наука и Техника, 2024. — 288 с. *(Изучение синтаксиса Python на примере создания простых консольных и графических игр).*

Маруччи, Ф. Создание игр в GDevelop без программирования / Ф. Маруччи. — Электронное издание, 2024. — 120 с. *(Практическое руководство по логике событий движка).*

Элс, М. Разработка игр на Construct 3 / М. Элс. — М.: ДМК Пресс, 2024. — 210 с. *(Книга по созданию 2D-аркад и платформеров).*

Литература для педагогов

Босова, Л. Л. Развитие алгоритмического и логического мышления школьников на уроках информатики / Л. Л. Босова // Информатика в школе. — 2024. — № 4. — С. 12–19.

Васильев, А. Н. Программирование на Python для начинающих. Особенности языка и тактика кодинга / А. Н. Васильев. — М.: Эксмо, 2024. — 448 с.

(Методические подходы к объяснению сложных тем: циклы, обработка исключений).

Никсон, Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5 / Р. Никсон. — 6-е изд. — СПб.: Питер, 2023. — 816 с. *(Для углубленной подготовки педагога к вопросам веб-разработки).*

Шелл, Д. Геймдизайн. Как создать игру, в которую будут играть все / Д. Шелл. — М.: Альпина Паблишер, 2022. — 640 с. *(Настольная книга по теории игрового баланса, механике уровней и удержанию внимания игрока).*

Электронные ресурсы

Официальный сайт Construct 3 [Электронный ресурс]. — URL: construct.net — *(Документация движка, бесплатные ассеты и веб-редактор).*

Официальный сайт GDevelop [Электронный ресурс]. — URL: gdevelop.io — *(Учебные туториалы по 2D и 3D геймдеву, примеры готовых шаблонов).*

HTML Academy [Электронный ресурс]. — URL: htmlacademy.ru — *(Интерактивные онлайн-курсы по HTML и CSS для тренировки базовых навыков верстки).*

W3Schools на русском [Электронный ресурс]. — URL: w3schoolsu.org — *(Справочник по тегам HTML и свойствам CSS с возможностью тестирования кода в браузере).*

Официальный сайт Python [Электронный ресурс]. — URL: python.org — *(Скачивание интерпретатора, официальная документация).*

Питонтьютор [Электронный ресурс]. — URL: pythontutor.ru — *(Бесплатный интерактивный онлайн-курс по основам Python с автоматической проверкой задач).*

Нормативно – правовое обеспечение программы

1. Федеральный закон от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации»;

2. Приказ Министерства просвещения Российской Федерации от 27 июля 2022 г. N 629 «Об утверждении порядка организации и осуществления

образовательной деятельности по дополнительным общеобразовательным программам»;

3. Санитарно-эпидемиологические правила и нормативы СанПиН 2.4.2.4283-26 «Санитарно- эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи», утвержденные Постановлением Главного государственного санитарного врача Российской Федерации от 02.06.2026 № 19, действующие до 1 сентября 2032 г.;

4. Санитарные правила и нормы СанПиН 1.2.3685-21 «Гигиенические нормативы и требования к условиям обеспечения безопасности и (или) безвредности человека факторов среды обитания», утвержденные Постановлением Главного государственного санитарного врача Российской Федерации от 28.01.2021 № 2; (зарегистрировано Министерством юстиции Российской Федерации 29 января 2021 г., регистрационный N 62296), с изменениями, внесенными постановлением Главного государственного санитарного врача Российской Федерации от 30 декабря 2022 г. N 24 (зарегистрировано Министерством юстиции Российской Федерации 9 марта 2023 г., регистрационный N 72558), действующими до 1 марта 2027 г.;

5. Письмо Минобрнауки России от 18.11.2015 № 09-3242 «О направлении информации» (Методические рекомендации по проектированию дополнительных общеразвивающих программ (включая разноуровневые программы);

6. Распоряжение Правительства РФ от 31 марта 2022 г. N 678-р об утверждении Концепции развития дополнительного образования детей до 2030 г. и плана мероприятий по ее реализации»

7. Приказ Министерства труда и социальной защиты Российской Федерации от 22.09.2021 № 652н "Об утверждении профессионального стандарта "Педагог дополнительного образования детей и взрослых".