Муниципальное бюджетное общеобразовательное учреждение
городского округа Тольятти
«Школа №13 имени Бориса Борисовича Левицкого»
(МБУ «Школа № 13»)

РАССМОТРЕНА	СОГЛАСОВАНА	УТВЕРЖДЕНА
Методическим объединением	Зам. директора	Приказом № 136 -ОД
Руководитель МО	по ВР	от 01.09.2026
Шмелёва К.О._____	Никитина И.Н.	Директор
	_____	МБУ «Школа № 13»
Протокол № 1 от 28.08.2026	28.08.2026	

OU=Директор, O=МБУ Школа№13, CN=Торопова Наталья Юрьевна, E=school13@edu.tgl.ru 009c2ec9e72df13c26 2026-09-01 10:06:59

Рабочая программа курса внеурочной деятельности «Алгоритмика»

Направление: техническая (основы алгоритмического мышления) направленность
Возрастная группа: 6 класс
Срок реализации: 1 год

Составитель:
Сторублевцева Т.В.
Учитель информатики

2026

1. Пояснительная записка

Рабочая программа курса внеурочной деятельности «Алгоритмика» для 5–6 классов разработана в соответствии с требованиями актуального Федерального государственного образовательного стандарта основного общего образования (ФГОС ООО), Федеральной образовательной программы (ФОП ООО), а также с учетом рабочей программы воспитания образовательной организации.

Цель курса: развитие алгоритмического, логического и системного мышления учащихся, формирование базовых навыков программирования (визуального и текстового), а также развитие метапредметных компетенций через проектно-исследовательскую деятельность.

Основные задачи:

- Сформировать представление об алгоритмах, их свойствах и исполнителях.
- Освоить базовые алгоритмические конструкции в визуальной среде (Scratch) и текстовой среде (Python).
- Развить навыки декомпозиции задач, поиска и исправления ошибок в коде (отладки).
- Обучить основам командного взаимодействия и публичной презентации ИТ-продуктов.

2. Содержание курса внеурочной деятельности

5 класс. Визуальное программирование в среде Scratch (34 часа)

Раздел 1. Мир алгоритмов и исполнителей (4 ч). Алгоритмы в повседневной жизни. Свойства алгоритмов (дискретность, понятность, определенность). Понятие исполнителя, система команд исполнителя (СКИ). Способы записи: словесный, блок-схемы.

Раздел 2. Знакомство со средой Scratch (6 ч). Интерфейс Scratch. Понятие спрайта, сцены, костюма. Координатная сетка экрана. Управление движением, анимация персонажей через смену костюмов. Звуковое оформление. Блоки событий. Создание интерактивных диалогов.

Раздел 3. Конструирование скриптов и логика (14 ч). Понятие цикла. Безусловные циклы и циклы со счетчиком. Расширение «Перо», геометрия и рисование орнаментов. Понятие ветвления. Условия. Одинарное и полное ветвление. Датчики и сенсоры: программирование физического взаимодействия объектов. Логические операторы (и, или, не). Циклы с условием. Параллельные процессы и передача сообщений между спрайтами.

Раздел 4. Работа с данными и переменными (6 ч). Переменная как ячейка памяти. Операции с переменными. Создание игрового счета. Генерация случайных чисел и их применение в игровых механиках. Понятие о списках (массивах) данных.

Раздел 5. Творческие мини-проекты (4 ч). Этапы разработки программного продукта: от идеи до реализации. Проектирование, программирование, отладка и тестирование. Публичная защита интерактивных игр и анимаций.

6 класс. Текстовое программирование и логика (34 часа)

Раздел 6. Формальные исполнители в КуМир (6 ч).

Введение в синтаксис среды «КуМир». Управление формальным исполнителем Робот. Изучение СКИ Робота, движение по лабиринтам. Использование циклов раз и циклов пока с проверкой датчиков стен.

Раздел 7. Синтаксис текстового языка Python (8 ч).

Переход от блоков к тексту. Среда разработки. Вывод данных (print()), синтаксис строк. Переменные в Python, динамическая типизация. Ввод данных (input()). Арифметические операции в коде, приоритеты вычислений. Типы данных int и float. Операции деления без остатка (//) и взятия остатка (%). Разработка консольного калькулятора.

Раздел 8. Реализация базовой логики в коде (10 ч).

Логический тип данных bool. Операторы сравнения. Условный оператор if, значимость отступов в Python. Полное ветвление if-else. Множественный выбор if-elif-else. Сложные условия с операторами and, or, not. Создание текстовых квестов. Цикл с условием while, валидация данных, защита от зацикливания. Цикл со счетчиком for, функция range. Вычисление математических последовательностей.

Раздел 9. Вспомогательные алгоритмы (6 ч).
Понятие декомпозиции. Объявление функций через ключевое слово def. Вызов функций. Функции с входными параметрами (аргументами). Возврат значений через return. Импорт встроенных модулей (библиотеки math, random).
Раздел 10. Итоговый командный хакатон (4 ч).
Командная разработка комплексного консольного приложения. Распределение ИТ-ролей (архитектор, программист, тестировщик). Разработка технического задания, написание кода, финальное тестирование и презентация готового цифрового продукта.

3. Планируемые результаты освоения курса

Личностные результаты:

- Готовность к саморазвитию и непрерывному обучению в цифровой среде.
- Сформированность навыков продуктивного сотрудничества, распределения ролей и ответственности в команде при реализации групповых ИТ-проектов.
- Критическое мышление при анализе информации и оценке корректности работы программных систем.

Метапредметные результаты:

- **Познавательные:** умение выделять существенные признаки, классифицировать, строить логические цепочки рассуждений, осуществлять декомпозицию сложных задач на подзадачи, использовать модели и схемы для решения алгоритмических задач.
- **Регулятивные:** способность самостоятельно ставить цели, планировать этапы деятельности, осуществлять отладку (поиск и исправление ошибок) программного кода, соотносить полученный результат с ожидаемым.
- **Коммуникативные:** умение развернуто и аргументированно обосновывать свои технические решения, публично представлять проектные результаты, бесконфликтно взаимодействовать в процессе групповой разработки.

Предметные результаты:

- Владение понятиями «алгоритм», «исполнитель», «система команд исполнителя», «переменная», «функция».
- Умение читать, модифицировать и составлять линейные, циклические и разветвляющиеся алгоритмы.
- Умение работать в визуальной среде (Scratch) и текстовой среде (Python), понимание базового синтаксиса языка.
- Опыт создания интерактивных цифровых продуктов (игр, симуляторов, анимаций, калькуляторов) от стадии проектирования до отладки.

4. Календарно-тематическое планирование

№ урока	Название модуля / Тема урока	Кол-во часов	Форма занятия	Основные виды деятельности учащихся
5 КЛАСС. ВИЗУАЛЬНОЕ ПРОГРАММИРОВАНИЕ (SCRATCH), 34 час				
Модуль 1	**Мир алгоритмов и исполнителей**	**4**		
1	Что такое алгоритм? Роль в жизни человека.	1	Лекция-дискуссия	Выделение алгоритмов в быту, изучение их свойств (дискретность).
2	Понятие исполнителя. СКИ.	1	Практикум	Определение формальных и неформальных исполнителей и их СКИ.
3	Способы записи алгоритмов. Блок-схемы.	1	Работа по образцу	Анализ графических блоков схем (действие, условие), построение схем.

4	Интерактивная игра «Пойми исполнителя».	1	Игра-квест	Составление и тестирование линейных команд в игровых ситуациях.
Модуль 2	**Знакомство со средой Scratch**	**6**		
5	Интерфейс Scratch. Спрайты и сцены.	1	Лаб. работа	Навигация по палитре блоков, настройка рабочей области проекта.
6	Координатная сетка. Движение спрайта.	1	Практикум	Расчет перемещения персонажа по осям X и Y, использование шагов.
7	Костюмы персонажей. Создание анимации.	1	Творческая работа	Разработка цикличной анимации ходьбы через смену костюмов.
8	Звуковое оформление в проектах.	1	Практикум	Импорт и обрезка звуковых дорожек, привязка звука к действию.
9	Первое интерактивное приложение.	1	Практикум	Настройка запуска скриптов по клику на флажок или клавишу.
10	Разработка диалога персонажей.	1	Проектный практикум	Синхронизация реплик персонажей с помощью пауз (ждать ... сек).
Модуль 3	**Конструирование скриптов и логика**	**14**		
11	Понятие цикла. Безусловный цикл.	1	Практикум	Применение блока «всегда» для организации фонового движения.
12	Цикл с заданным числом повторений.	1	Лаб. работа	Оптимизация длинных скриптов с помощью блока «повторить».
13	Рисование фигур. Расширение «Перо».	1	Практикум	Управление виртуальным пером, настройка цвета и толщины линий.
14	Программирование орнаментов и узоров.	1	Практикум	Математический расчет углов поворота для рисования многоугольников.
15	Понятие ветвления. Блок «если... то».	1	Комбинированный	Конструирование условных выражений, проверка истинности.
16	Полное ветвление «если... то... иначе».	1	Практикум	Программирование двух вариантов поведения объекта при проверке.
17	Взаимодействие объектов. Сенсоры.	1	Лаб. работа	Работа с блоками касания цвета, краев экрана и других спрайтов.
18	Разработка игры «Лабиринт».	1	Проектный практикум	Программирование отскока персонажа от стен лабиринта.
19	Логические операторы «и», «или», «не».	1	Семинар	Объединение нескольких условий в один логический блок.
20	Сложные ветвления в игровых механиках.	1	Практикум	Ограничение перемещений спрайта границами игрового поля.
21	Цикл с условием «повторять пока не...».	1	Лаб. работа	Организация циклов, завершающихся при достижении цели.
22	Параллельные процессы и сообщения.	1	Практикум	Изучение механики «передать сообщение» для связи скриптов.
23	Симуляция работы светофора.	1	Практикум	Синхронизация движения автомобилей по сигналам таймера.
24	Комплексный практикум «Открытка».	1	Творческая работа	Разработка многосценарного интерактивного поздравления.
Модуль 4	**Работа с данными и переменными**	**6**		
25	Что такое переменная? Создание в Scratch.	1	Комбинированный	Изучение понятия переменной как контейнера для хранения данных.
26	Изменение значений переменных.	1	Практикум	Перезапись и инкрементирование значений числовых переменных.

27	Программирование игрового счета.	1	Лаб. работа	Разработка системы начисления игровых очков и штрафных баллов.
28	Случайные числа в алгоритмах.	1	Практикум	Использование случайных чисел для имитации непредсказуемости.
29	Разработка игры «Пойми предмет».	1	Проектный практикум	Программирование падения предметов со случайных координат X.
30	Понятие о списках (массивах) данных.	1	Практикум	Создание динамических списков, добавление строк пользователем.
Модуль 5	**Творческие мини-проекты**	**4**		
31	Проектирование индивидуального проекта.	1	Мозговой штурм	Составление дизайн-документа проекта, отрисовка блок-схем.
32	Практический этап разработки проекта.	1	Разработка за ПК	Самостоятельное кодирование логики игры, импорт ассетов.
33	Финальное тестирование и оптимизация.	1	Тестирование	Поиск багов методом «черного ящика», исправление ошибок отладки.
34	Защита творческих проектов.	1	Выставка-презентация	Демонстрация работы игры, ответы на вопросы экспертов-одноклассников.
6 КЛАСС. ТЕКСТОВОЕ ПРОГРАММИРОВАНИЕ И ЛОГИКА. 34 час				
Модуль 6	**Формальные исполнители в КуМир**	**6**		
35	Введение в среду «КуМир». Структура кода.	1	Комбинированный	Изучение служебных слов алг, нач, кон. Синтаксический разбор.
36	Исполнитель Робот и его СКИ.	1	Практикум	Написание команд перемещения и закрашивания клеток поля.
37	Составление линейных алгоритмов.	1	Лаб. работа	Трассировка линейного алгоритма движения по карте, поиск тупиков.
38	Использование циклов в КуМир.	1	Практикум	Оптимизация кода Робота через конструкции фиксированных повторений.
39	Цикл с условием нц пока ... кц.	1	Лаб. работа	Написание алгоритмов обработки стен неизвестной протяженности.
40	Соревнование «Мастер Робот».	1	Турнир	Программирование Робота для очистки полей на скорость и точность.
Модуль 7	**Синтаксис текстового языка Python**	**8**		
41	Переход к Python. Среда разработки.	1	Комбинированный	Сопоставление визуальных блоков и текстового кода в IDE Thonny.
42	Вывод информации. Функция print().	1	Практикум	Вывод форматированных строк с использованием разделителей.
43	Переменные в Python. Типы данных.	1	Практикум	Объявление переменных, присвоение значений, правила имен.
44	Ввод данных. Функция input().	1	Практикум	Чтение строк из консоли, сохранение текстовых ответов.
45	Арифметика в коде. Базовые операции.	1	Решение задач	Программирование математических выражений, учет приоритетов.
46	Работа с числами int и float.	1	Практикум	Преобразование типов данных с использованием явного приведения.
47	Целочисленное деление и остаток.	1	Решение задач	Выделение разрядов многозначных чисел через `//` и `%`.

48	Практическая работа «Калькулятор».	1	Самостоятельная	Создание консольного интерфейса вычисления арифметических задач.
Модуль 8	**Реализация базовой логики в коде**	**10**		
49	Логический тип bool. Операторы сравнения.	1	Комбинированный	Оценка истинности выражений, применение знаков `==`, `!=`.
50	Условный оператор if. Роль отступов.	1	Практикум	Форматирование кода блоками отступов, синтаксис двоеточий.
51	Полное ветвление: конструкция if - else.	1	Практикум	Написание бинарных развилок для обработки ложного условия.
52	Множественный выбор if - elif - else.	1	Лаб. работа	Каскадное ветвление условий для многоуровневого выбора.
53	Логические операторы and, or, not.	1	Решение задач	Написание комплексных условий для валидации диапазонов чисел.
54	Разработка игры «Текстовый квест».	1	Творческий практикум	Создание интерактивной новеллы с ветвлением сюжета в консоли.
55	Цикл с условием while. Зацикливание.	1	Практикум	Программирование циклов с изменяемым флагом выхода.
56	Валидация данных. Игра «Угадай число».	1	Лаб. работа	Написание игрового цикла с проверкой больше/меньше и счетчиком.
57	Цикл со счетчиком for и функция range().	1	Практикум	Итерация по числовым рядам с настройкой старта, стопа и шага.
58	Решение алгоритмических задач на циклы.	1	Практикум	Написание программ расчета факториала, суммирования чисел.
Модуль 9	**Вспомогательные алгоритмы**	**6**		
59	Понятие декомпозиции. Зачем нужны функции.	1	Лекция-семинар	Анализ преимуществ модульной структуры кода перед монолитной.
60	Создание собственных функций. Ключевое слово def.	1	Практикум	Определение сигнатуры функции, вызов функции по её имени.
61	Функции с параметрами (аргументами).	1	Практикум	Реализация передачи локальных переменных внутрь области функции.
62	Возврат значения. Оператор return.	1	Лаб. работа	Практический разбор перехвата вычисленных данных из функций.
63	Разработка модуля геометрических утилит.	1	Самостоятельная	Создание библиотеки кастомных функций площади и периметра.
64	Знакомство с модулями. Библиотека random.	1	Практикум	Подключение внешних библиотек, вызов метода `randint()`.
Модуль 10	**Итоговый командный хакатон**	**4**		
65	Постановка задачи хакатона. Разработка ТЗ.	1	Командная работа	Распределение ролей, декомпозиция проекта на задачи (Trello/доска).
66	Командное написание программного кода.	1	Хакатон за ПК	Сборка программных модулей в единое приложение (база, текстовая RPG).
67	Тестирование продукта и исправление багов.	1	Тестирование	Моделирование некорректного пользовательского ввода, отладка крашей.
68	Финальный демонстрационный турнир.	1	Презентация проектов	Публичная защита кода, ревью архитектуры, подведение итогов.
ВСЕГО	За 2 года обучения	**68**		

5. Учебно-методическое и материально-техническое обеспечение курса

Материально-техническая база:

1. Учебный кабинет, оборудованный персональными компьютерами для каждого обучающегося.
2. Интерактивная доска или мультимедийный проектор для демонстрации кода и блок-схем.
3. Локальная сеть с выходом в Интернет (с установленной системой контентной фильтрации).

Программное обеспечение:

- Операционная система (Windows / Linux).
- Браузер с доступом к онлайн-версии Scratch или установленная офлайн-версия Scratch Desktop.
- Учебная алгоритмическая среда «КуМир» (актуальная версия).
- Среда разработки Python (IDLE, Thonny или аналогичная легковесная IDE).

Электронные образовательные ресурсы (ЭОР):

- Российская электронная школа (РЭШ) — модули по информатике и логике.
- Официальное сообщество Scratch (scratch.mit.edu) — галерея проектов, руководства.
- Образовательная платформа Stepik — бесплатные базовые интерактивные курсы по Python для начинающих.
- Платформа «Сириус.Курсы» — материалы по введению в алгоритмы.

6. Критерии и система оценивания

В соответствии со спецификой внеурочной деятельности применяется **безоценочная система** (зачет/незачет), ориентированная на индивидуальный прогресс учащегося.

Оценка индивидуальных достижений учащихся в рамках курса внеурочной деятельности «Алгоритмика» носит **комплексный, безоценочный (зачетный) характер** и базируется на сочетании текущего контроля, проектной деятельности и экспертного наблюдения.

Система текущего контроля

Текущий контроль осуществляется на каждом занятии по результатам выполнения индивидуальных практических и лабораторных работ за компьютером.

- **«Зачтено» (Базовый уровень):**
  - Программа успешно компилируется/запускается в изучаемой среде (Scratch, КуМир, Python).
  - Алгоритм решает поставленную базовую задачу (например, исполнитель доходит до цели, калькулятор выполняет сложение).
  - Ученик понимает структуру собственного кода и может объяснить назначение основных блоков или команд.
- **«Зачтено» (Повышенный уровень):**
  - Код программы оптимизирован (вместо дублирования строк используются циклы и ветвления).
  - Интерфейс программы интуитивно понятен, предусмотрена защита от простейших ошибок ввода (валидация данных).
  - Ученик может самостоятельно локализовать и исправить логическую ошибку (провести отладку).
- **«Не зачтено»:**
  - Программа не запускается из-за критических синтаксических ошибок.
  - Алгоритм полностью не соответствует условию задачи.
  - Отказ от выполнения практической работы или её компиляции.

Критерии оценивания итогового проекта / хакатона

Итоговый контроль проводится в конце каждого учебного года (34-й и 68-й уроки) в форме защиты группового или индивидуального цифрового продукта. Оценка выставляется на основе коллегиального решения (самооценка, оценка одноклассников и экспертное заключение учителя) по 10-

балльной шкале, где **7–10 баллов** соответствуют повышенному уровню, **4–6 баллов** — базовому, **менее 4 баллов** — неудовлетворительному.

Критерий	Описание	Баллы
Техническая сложность	Использование продвинутых конструкций: в 5 классе — списки, параллельные сообщения, сложные сенсоры; в 6 классе — собственные функции с параметрами, обработка исключений, внешние библиотеки (random).	0 – 3
Работоспособность и отладка	Отсутствие «зависаний» и аварийных завершений программы. Корректность работы алгоритма при различных сценариях использования и тестирования крайних случаев.	0 – 2
Оригинальность и дизайн	Уникальность идеи игрового сюжета или полезность утилиты. Качество проработки интерфейса (консольного меню в Python или графического оформления сцен в Scratch).	0 – 2
Командная работа и роли	Четкое распределение обязанностей в группе (для хакатонов). Вклад каждого участника в финальный репозиторий/код проекта, знание своей части программы.	0 – 2
Презентация продукта	Способность понятно рассказать об архитектуре программы, обосновать выбор алгоритмических структур, аргументированно ответить на вопросы аудитории.	0 – 1
МАКСИМАЛЬНЫЙ БАЛЛ	**За итоговый проектный модуль**	**10**

Итоговый «Зачет» за курс выставляется учащемуся при условии выполнения не менее 70% текущих практических работ и наборе не менее 4 баллов за итоговый проект.

Метапредметная матрица оценивания (навыки декомпозиции и логики)

Оценка универсальных учебных действий (УУД) фиксируется учителем дважды за год (в конце первого и второго полугодия):

Регулятивные УУД (Управление процессом кодинга): Способность разбить написание программы на шаги. Умение пользоваться пошаговой отладкой кода (Step-by-step). Реакция на сообщения интерпретатора об ошибках — самостоятельный поиск пропущенных двоеточий, скобок или неверных отступов.

Познавательные УУД (Моделирование): Умение сопоставить текстовую задачу с блок-схемой. Способность абстрагироваться от частной задачи к общему математическому правилу (например, перевод физической формулы в строку кода на Python с учетом приоритета операторов).

Критерии оценивания личностных результатов

Личностные результаты не подлежат стандартной числовой оценке (отметке) и фиксируются методом экспертного наблюдения на основе карты развития учащихся по следующим критериям:

Критерий оценки	Высокий уровень (зачтено)	Средний уровень (зачтено)	Низкий уровень (не зачтено)
Учебно-познавательный интерес	Проявляет высокую самостоятельность; ищет нестандартные алгоритмические решения; изучает дополнительные материалы.	Выполняет все базовые практические задания по образцу; проявляет интерес к новым темам среды.	Пассивен на занятиях; выполняет задания только под прямым контролем учителя; интерес угасает при первых трудностях.
Командное взаимодействие и сотрудничество	Легко распределяет ИТ-роли в хакатоне; помогает одноклассникам; конструктивно решает спорные моменты в коде.	Принимает условия работы в группе; бесконфликтно выполняет свою часть проектного задания.	Отказывается от командной работы; не принимает чужую точку зрения; перекладывает ответственность на других.

Критичность мышления и отладка	Спокойно воспринимает баги; системно тестирует свой код на крайние случаи; объективно оценивает чужие проекты.	Осознает наличие ошибок в программе; обращается за помощью к подсказкам или учителю для исправления кода.	Бросает работу при возникновении программной ошибки; проявляет негативные эмоции; не умеет проверять свой код.
Цифровая гигиена и этика	Соблюдает правила техники безопасности за ПК; уважает авторские права при использовании чужих спрайтов и ассетов.	Выполняет требования учителя по работе за компьютером; не нарушает правила поведения в сети.	Регулярно нарушает дисциплину и правила работы за ПК; использует чужой программный код без указания авторства.

7. Список литературы и интернет-ресурсов

Литература для педагогического работника

- Босова Л. Л., Босов А. Ю. Информатика. 5–6 классы: методическое пособие. — М.: Просвещение, 2023–2026.
- Поляков К. Ю., Еремин Е. А. Программирование. Python. Введение в алгоритмизацию. — М.: Лаборатория знаний, 2024.
- Денисова А. В. Визуальное программирование в среде Scratch для учителей и школьников. — СПб.: БХВ-Петербург, 2023.

Литература для обучающихся и родителей

- Вордерман К. Программирование для детей. Иллюстрированное руководство по языкам Scratch и Python. — М.: Манн, Иванов и Фербер, 2024.
- Мараско С. Scratch для начинающих. Создавай игры и анимацию. — М.: Эксмо, 2025.
- Кольцов Д. М. Python для школьников. Первые шаги в программировании. — СПб.: Наука и Техника, 2025.

Интернет-ресурсы и цифровые образовательные платформы

- Официальный сайт Scratch (scratch.mit.edu) — среда разработки, база учебных проектов, руководства.
- Система «КуМир» (niisi.ru/kumir) — страница загрузки официальной среды исполнителей.
- Платформа Stepik (stepik.org) — интерактивные онлайн-курсы по основам синтаксиса Python.
- Яндекс Учебник / Кодвардс (education.yandex.ru) — интерактивные тренажеры по логике и алгоритмизации.