ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБЩЕОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ЛИЦЕЙ №144
КАЛИНИНСКОГО РАЙОНА САНКТ-ПЕТЕРБУРГА

ПРИНЯТА
Педагогическим советом
ГБОУ лицея №144
Калининского района Санкт-Петербурга
Протокол от 23.06.2026 г. № 11

УТВЕРЖДЕНА
Приказом от 23.06. 2026 г. №40
директор ГБОУ лицея №144
Калининского района Санкт-Петербурга
_____ / Федорова Л. А. /

Дополнительная общеразвивающая программа
«АЙТИЗНАЙКА»

Срок освоения: 36 недель (72 академических часа, 2 часа в неделю)
Возраст обучающихся: 8–10 лет

Разработчик:
Крутиков Антон Сергеевич,
педагог дополнительного образования

ГБОУ лицей № 144
Калининского района
Санкт-Петербурга
Подписано электронной подписью
01.09.2026 16:23
директор
Федорова Лолита Анатольевна
7804140160-15-1793021045-20261026-298-1-1624-05

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Направленность программы – ***техническая.***

Адресат программы: обучающиеся 8–10 лет, делающие первые шаги в изучении информационных технологий.

Актуальность программы.

Дополнительная общеобразовательная программа «Айтизнайка» имеет техническую направленность и адресована обучающимся, делающим первые шаги в изучении информационных технологий. Курс направлен на развитие алгоритмического мышления и формирование базовых представлений о программировании. Знакомство с Python в доступной игровой форме помогает детям научиться последовательно мыслить, находить закономерности, составлять понятные инструкции и видеть результат собственных действий. Практические задания адаптированы к возрасту обучающихся и постепенно усложняются: от простых команд и вычислений до создания небольших интерактивных программ.

Программа рассчитана на полный учебный год. Последовательное освоение базовых конструкций Python, решение игровых и прикладных задач и выполнение итогового мини-проекта позволяют закрепить полученные знания и создать фундамент для дальнейшего изучения информатики и программирования.

Уровень освоения программы – стартовый.

Объем и срок освоения программы.

Программа рассчитана на 72 академических часа, 36 недель. Режим занятий: 1 раз в неделю по 2 академических часа (с перерывом 10–15 минут).

Цель: развитие алгоритмического мышления и формирование первоначальных навыков программирования на языке Python через решение игровых и практико-ориентированных задач.

Задачи

Обучающие:

- научить правилам безопасной и ответственной работы за компьютером;
- познакомить с понятием алгоритма, исполнителя и языком программирования Python;
- научить вводить, запускать и сохранять простые программы;
- освоить базовые конструкции Python: вывод данных, переменные, ввод, арифметические операции и условия;
- познакомить с циклами, строками, списками и простыми функциями на доступных примерах;
- научить находить и исправлять простые ошибки, разбивать задачу на последовательные шаги.

Развивающие:

- развитие логического, алгоритмического и творческого мышления;
- развитие умения разбивать задачу на подзадачи и выбирать последовательность действий;
- развитие внимательности, самостоятельности и навыков проверки результата;
- развитие способности находить разные способы решения игровой или прикладной задачи.

Воспитательные:

- воспитание усидчивости, аккуратности и ответственности при работе с компьютером;
- формирование интереса к информационным технологиям и программированию;
- формирование культуры безопасного поведения в цифровой среде;
- развитие умения представлять свою работу и доброжелательно обсуждать результаты других.

ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ

Язык реализации программы: русский

Форма обучения: очная

Условия приема на обучение: принимаются все желающие в возрасте 8–10 лет.

Количество обучающихся в группе: не менее 15 человек.

Формы проведения занятий

Основная форма – учебное занятие. Используются объяснение и демонстрация на экране, игровые упражнения, выполнение пошаговых заданий, индивидуальная практика за компьютером, работа в парах и создание небольших программ по собственному замыслу. Материал подаётся от простого к сложному с регулярным повторением и поддержкой педагога.

Материально-техническое оснащение программы

Оснащение	Кол-во
Учебный кабинет (компьютерный класс)	1
Персональный компьютер / ноутбук для каждого обучающегося	10–15
Проектор / интерактивная доска	1
Экран для проектора	1
Среда программирования Python 3 и редактор кода (установленные или доступные онлайн)	10–15 рабочих мест

Планируемые результаты освоения программы.

Личностные:

- соблюдение правил безопасной работы за компьютером и ответственное отношение к цифровым инструментам;
- проявление интереса к программированию, логическим задачам и созданию собственных программ;
- понимание того, что программа выполняет команды в определённой последовательности;
- готовность продемонстрировать программу и объяснить её назначение.

Метапредметные:

- умение планировать последовательность действий для решения задачи;
- умение представлять задачу в виде понятного алгоритма и делить её на простые шаги;
- умение проверять результат выполнения программы и находить простые ошибки;
- умение работать по инструкции и постепенно переходить к самостоятельному составлению алгоритмов.

Предметные:

- знание базовых правил безопасной работы с компьютером;
- понимание назначения среды программирования и основных элементов программы на Python;
- умение использовать print(), input(), переменные и простые арифметические операции;
- умение применять условия if/else и циклы на простых задачах;
- умение работать с простыми строками и списками, использовать готовые и самостоятельно составленные функции на начальном уровне;
- умение запускать, проверять, исправлять и сохранять небольшие программы.

Учебный план

№ п/п	Название раздела, темы	Количество часов			Формы контроля/ / аттестации
		Всего	Теория	Практика	
1	Вводное занятие. Безопасность. Что такое алгоритм и программа	2	1	1	Беседа, игровое задание
2	Первые программы на Python. Вывод данных и простые команды	8	2	6	Практическое задание
3	Переменные, ввод данных и вычисления	10	3	7	Практическое задание
4	Условия и логика принятия решений	10	3	7	Практическое задание
5	Циклы: повторение действий	10	2	8	Практическое задание
6	Строки, списки и работа с наборами данных	10	3	7	Практическое задание
7	Функции, отладка и создание игровой программы	14	3	11	Практическая работа, наблюдение
8	Итоговый мини-проект и презентация	8	1	7	Защита мини-проекта
	Итого	72	18	54	

СОДЕРЖАНИЕ

Тема 1. Вводное занятие. Безопасность. Что такое алгоритм и программа (2 ч)

Теория. Правила безопасной работы за компьютером. Понятия «команда», «исполнитель», «алгоритм», «программа». Где встречается программирование.
Практика. Игровые алгоритмы: составление точных инструкций для знакомого действия, поиск ошибок в последовательности команд. Знакомство со средой Python и запуск первой программы.
Форма контроля. Беседа, выполнение простого задания.

Тема 2. Первые программы на Python. Вывод данных и простые команды (8 ч)

Теория. Назначение языка Python. Команда print(), текстовые строки, кавычки, комментарии. Порядок выполнения команд и типичные ошибки в записи.
Практика. Создание программ-приветствий, вывод рисунков из символов, оформление короткой истории или поздравления с помощью print().
Форма контроля. Практическое задание.

Тема 3. Переменные, ввод данных и вычисления (10 ч)

Теория. Понятие переменной. Имена переменных, числа и строки. Команда input(), преобразование введённых данных в числа, арифметические операции.
Практика. Программы «Знакомство», «Возраст в следующем году», «Счётчик очков», вычисление суммы и простых величин.
Форма контроля. Самостоятельное практическое задание.

- Создание программы с приветствием и именем пользователя;
- Ввод двух чисел и вычисление суммы;
- Расчёт простых игровых очков;
- Проверка программы на разных исходных данных.
  Форма контроля. Практическое задание, работа по карточке.

Тема 4. Условия и логика принятия решений (10 ч)

Теория. Сравнение чисел и строк. Условные конструкции if, else, elif. Операторы сравнения и логические выражения.
Практика. Программы «Угадай число», определение результата по условию, простая викторина с проверкой ответа.
Форма контроля. Практическое задание.

- Проверка, является ли число положительным или отрицательным;
- Мини-викторина с несколькими вариантами ответа;
- Игровая задача с выбором действия по условию.
  Форма контроля. Практическое задание, самостоятельная работа.

Тема 5. Циклы: повторение действий (10 ч)

Теория. Зачем нужны циклы. Циклы for и while на простых примерах. Счётчик, диапазон повторений и условие остановки.
Практика. Вывод последовательностей чисел, таблица умножения на выбранное число, повторяющиеся рисунки из символов, простые игровые счётчики.
Форма контроля. Практическое задание.

- Вывод чисел от 1 до заданного значения;
- Создание простого обратного отсчёта;
- Решение задач на повторение команд и подсчёт результата.
  Форма контроля. Практическое задание.

Тема 6. Строки, списки и работа с наборами данных (10 ч)

Теория. Строка как последовательность символов. Длина строки и обращение к отдельным элементам. Список как набор значений, получение элемента по индексу и простые операции со списком. Практика. Обработка имени пользователя, список любимых игр или животных, выбор случайного элемента из подготовленного набора (с поддержкой педагога). Форма контроля. Практическое задание.

Тема 7. Функции, отладка и создание игровой программы (14 ч)

Теория. Понятие функции и её назначение. Повторное использование команд. Типичные ошибки в коде: синтаксические ошибки и неверная логика. Последовательность проверки и исправления программы. Практика. Создание программы-викторины, текстовой игры или тренажёра с вводом ответов, условиями и циклами. Планирование проекта и поэтапная разработка.

- «Викторина» с подсчётом правильных ответов;
- «Угадай число» с подсказками и ограничением попыток;
- «Текстовый помощник» или простая игра по выбору пользователя;
- Проверка программы на разных вариантах ввода, поиск ошибок и доработка по обратной связи. Форма контроля. Практическое задание, наблюдение педагога.

Тема 8. Итоговый мини-проект и презентация (8 ч)

Практика. Завершение собственной программы, тестирование и исправление ошибок. Подготовка короткого рассказа о назначении программы, используемых командах и возникших трудностях. Демонстрация результата группе. Форма контроля. Защита мини-проекта.

МЕТОДИЧЕСКИЕ и ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ

Используемые технологии и методы:

1. Объяснительно-иллюстративный метод (демонстрация коротких примеров кода и пошагового запуска программы).
2. Репродуктивный метод (выполнение заданий по образцу, алгоритму и карточке-подсказке).
3. Проблемно-поисковый метод (поиск ошибок, выбор последовательности действий и решение игровых задач).
4. Технология проектного обучения (создание собственной небольшой программы с постепенной доработкой).
5. Здоровьесберегающая технология (эргономичная организация рабочего места, перерывы и упражнения для глаз).

Дидактические средства: *презентации и демонстрационные примеры, карточки с пошаговыми алгоритмами, задания на поиск ошибок, схемы последовательности действий, игровые задачи и шаблоны начальных программ.*

Информационные источники:

Список литературы для педагога:

1. Материалы для педагога по обучению детей основам алгоритмизации и программирования на Python.
2. Методические рекомендации по организации практических занятий по программированию для младших школьников.

Список литературы для обучающихся:

1. Учебные карточки и практические задания по базовым конструкциям Python, алгоритмам и созданию простых игр.

Интернет-источники:

1. Официальная документация Python (справочные материалы для педагога): https://docs.python.org/3/

Оценочные материалы

Входной контроль.

На первом занятии проводится беседа и несколько игровых заданий для выявления начальных представлений об алгоритмах и опыта работы с компьютером. Предварительное знание Python не требуется. Результаты используются для выбора темпа и объёма поддержки.

Текущий контроль.

После каждой темы обучающиеся выполняют практическое задание. Педагог обращает внимание на правильность записи команд, понимание последовательности действий, использование изученных конструкций, умение проверить результат и исправить ошибку.

Итоговый контроль.

На последних занятиях проводится демонстрация и защита мини-проектов. Оценивается:

- соответствие программы поставленной задаче или игровому замыслу;
- корректность использования изученных команд и конструкций;
- работоспособность программы, проверка и исправление ошибок;
- умение кратко объяснить принцип работы и представить результат.

Параметры и критерии оценки

Критерии оценки /	Параметр: уровни освоения программы		
	Низкий (1 балл)	Средний (2 балла)	Высокий (3 балла)
Понимание алгоритма	Не понимает порядок действий даже с подсказкой	Следует алгоритму с отдельными подсказками	Самостоятельно составляет и объясняет простой алгоритм
Использование Python	Нужна постоянная помощь при записи и запуске кода	Использует изученные команды по образцу	Самостоятельно применяет изученные команды в задаче
Условия и циклы	Не может применить условие или повторение	Применяет конструкцию с подсказкой	Корректно использует условия и циклы в программе
Поиск ошибок	Не проверяет программу и затрудняется найти ошибку	Находит отдельные ошибки с помощью педагога	Проверяет результат и самостоятельно исправляет простые ошибки

Самостоятельность	Работает только при пошаговой помощи	Выполняет знакомые задания с отдельными подсказками	Самостоятельно планирует и выполняет основные этапы работы
Презентация проекта	Не может объяснить назначение программы	Объясняет программу с помощью вопросов педагога	Понятно описывает задачу, работу программы и результат

Уровень освоения программы:

- 6–10 баллов – низкий;
- 11–14 баллов – средний;
- 15–18 баллов – высокий.