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 «ПРОГРАММИРОВАНИЕ НА PYTHON»

Срок освоения: 36 недель (72 академических часа, 2 часа в неделю)
Возраст обучающихся: 13–17 лет

Разработчик:
Терехова Светлана Николаевна,
Силин Андрей Григорьевич,
педагоги дополнительного образования

ГБОУ лицей № 144
Калининского района
Санкт-Петербурга
Подписано электронной подписью
01.09.2026 16:23
директор
Федорова Лолита Анатольевна
7804140160-15-1793021473-20261026-298-1-1631-13

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Направленность программы – техническая.

Адресат программы: обучающиеся 13–17 лет.

Актуальность программы.

Актуальность программы обусловлена широким применением программирования в образовании, науке, технологиях и повседневной жизни. Язык Python отличается понятным синтаксисом и позволяет постепенно переходить от простых алгоритмов к созданию полноценных программ. Освоение Python помогает обучающимся развивать логическое и алгоритмическое мышление, учиться анализировать задачи, проектировать решения, работать с данными и создавать собственные цифровые продукты.

Программа рассчитана на полный учебный год и построена по принципу «от простого к сложному»: от знакомства со средой разработки и базовыми конструкциями Python до использования циклов, строк, списков, функций и создания итогового проекта. Практические задания связывают изучаемый материал с реальными и учебными задачами.

Уровень освоения программы – общекультурный.

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: формирование базовых и развивающихся навыков программирования на языке Python, развитие алгоритмического мышления и способности самостоятельно разрабатывать, тестировать и улучшать программы.

Задачи

Обучающие:

- закрепить правила безопасной работы за компьютером и организации файлов программного проекта;
- познакомить с языком Python, средой разработки и принципами выполнения программ;
- освоить переменные, основные типы данных, ввод и вывод, арифметические операции;
- научить использовать условные конструкции, логические выражения и циклы;
- освоить базовые приёмы обработки строк, списков и последовательностей данных;
- познакомить с функциями и научить писать, тестировать, отлаживать и документировать небольшие программы;

Развивающие:

- развитие алгоритмического, логического и вычислительного мышления;
- развитие умения разбивать сложную задачу на подзадачи и выстраивать последовательность действий;
- развитие точности при анализе условий, вычислениях и проверке результатов программы;
- развитие умения находить ошибки в алгоритмах и программном коде и выбирать подходящий способ решения.

Воспитательные:

- воспитание усидчивости, аккуратности и ответственности при написании программ;
- формирование интереса к информатике, программированию и цифровым технологиям;
- формирование культуры использования компьютера как инструмента решения практических и учебных задач;
- формирование навыков объяснения алгоритма и представления результатов своей работы.

ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ

Язык реализации программы: русский

Форма обучения: очная

Условия приема на обучение: принимаются все желающие в возрасте 13–17 лет.

Количество обучающихся в группе: не менее 15 человек.

Формы проведения занятий

Основная форма – учебное занятие. Используются объяснение и демонстрация программного кода, индивидуальная практическая работа за компьютером, решение задач, работа в парах, взаимопроверка и проектная деятельность.

Материально-техническое оснащение программы

Оснащение	**Кол-во**
Рабочий кабинет (компьютерный класс)	1
Персональный компьютер / ноутбук с установленной средой разработки Python	10–15
Проектор / интерактивная доска	1
Проектор / интерактивная доска	1
Программное обеспечение: Python 3 и среда разработки (IDLE, Thonny или аналогичная)	10–15

Планируемые результаты освоения программы.

Личностные:

- проявление усидчивости и аккуратности при написании и проверке программ;
- устойчивый интерес к информатике и программированию;
- понимание роли алгоритмов и программирования в решении современных задач;
- готовность объяснить ход решения и представить результат своей работы.

Метапредметные:

- умение планировать алгоритм решения задачи и проверять его корректность;
- умение анализировать условие задачи, выделять исходные данные и требуемый результат;
- умение находить и исправлять синтаксические и логические ошибки в программе;
- умение составлять, трассировать и реализовывать алгоритмы на языке Python.

Предметные:

- знание правил безопасной работы за компьютером и основных правил оформления программного кода;
- знание базового синтаксиса Python, операторов ввода и вывода, основных типов данных;
- умение использовать переменные, арифметические выражения и условные операторы;
- умение применять циклы и логические условия при решении задач;
- умение обрабатывать строки и списки в простых алгоритмических задачах;
- умение самостоятельно разрабатывать, запускать, тестировать и отлаживать программы на Python для решения учебных и практических задач.

Учебный план

№ п/п	Название раздела, темы	Количество часов			Формы контроля/ аттестации
		Всего	Теория	Практика	
1	Вводное занятие. Техника безопасности. Среда Python	2	1	1	Беседа, вводное задание
2	Основы Python. Переменные, типы данных, ввод и вывод	8	2	6	Практическое задание
3	Арифметические выражения. Условия и логические операции	10	2	8	Практическое задание
4	Циклы и алгоритмы с повторением	12	3	9	Практическое задание
5	Трассировка алгоритмов. Тестирование и отладка	8	2	6	Практическое задание
6	Строки и списки. Обработка данных	12	3	9	Практическое задание
7	Функции и создание собственных программ	12	2	10	Практическое задание
8	Итоговый проект. Разработка и презентация программы	8	1	7	Защита проекта
	Итого	72	16	56	

СОДЕРЖАНИЕ

Тема 1. Вводное занятие. Инструктаж по ТБ. Среда Python (2 ч)

Теория. Правила безопасной работы за компьютером. Назначение языка Python, области применения программирования. Знакомство со средой разработки, запуск программы и сохранение файла.
Практика. Запуск первой программы, вывод сообщения на экран, сохранение и повторный запуск файла.
Форма контроля. Беседа, выполнение вводного задания.

Тема 2. Основы языка Python. Переменные, типы данных, ввод и вывод (8 ч)

Теория. Синтаксис программы, переменные, целые и вещественные числа, строки, функции input() и print(), преобразование типов, арифметические операции и целочисленное деление.

Практика. Программы для вычисления выражений, перевода единиц, обработки введённых чисел и решения простых прикладных задач.
Форма контроля. Практическое задание.

Тема 3. Арифметические выражения. Условия и логические операции (10 ч)

Теория. Операторы if, elif, else; сравнения; логические операции and, or, not; вложенные условия; анализ граничных случаев.
Практика. Решение задач на сравнение чисел, определение принадлежности диапазону, выбор по условию, проверку делимости и анализ логических выражений.
Форма контроля. Практическое задание.

- Определение наибольшего и наименьшего из нескольких чисел;
- Проверка делимости и принадлежности числовому промежутку;
- Построение программы с несколькими ветвями условия;
- Решение задач с логическими выражениями и проверка граничных случаев.
  Форма контроля. Практическое задание, самостоятельная работа.

Тема 4. Циклы. Алгоритмы с повторением (12 ч)

Теория. Циклы for и while, функция range(), счётчик цикла, накопление суммы и произведения, условие завершения цикла, типичные ошибки.
Практика. Решение задач на перебор чисел, подсчёт количества элементов, суммирование, поиск значений и обработку последовательности вводимых данных.
Форма контроля. Практическое задание.

- Вычисление суммы и произведения последовательности;
- Подсчёт чисел, удовлетворяющих заданному условию;
- Перебор диапазона, поиск подходящих значений и отладка цикла.
  Форма контроля. Практическое задание.

Тема 5. Алгоритмы и исполнение программ. Трассировка и отладка (8 ч)

Теория. Понятие алгоритма, последовательность выполнения команд, значения переменных на каждом шаге, таблица трассировки, тестирование программы, синтаксические и логические ошибки.
Практика. Пошаговое выполнение коротких фрагментов кода, заполнение таблиц значений, поиск ошибок и исправление алгоритмов.
Форма контроля. Практическое задание.

- Определение значений переменных после выполнения фрагмента программы;
- Заполнение таблицы трассировки алгоритма;
- Поиск и исправление ошибок в коде.
  Форма контроля. Практическое задание.

Тема 6. Строки и списки. Обработка последовательностей (12 ч)

Теория. Индексация строк, длина строки, срезы и основные методы; создание списков, обращение к элементам, перебор, поиск минимума и максимума, подсчёт и фильтрация элементов.
Практика. Решение задач на обработку символов и слов, подсчёт элементов, поиск экстремальных значений и преобразование последовательностей данных.
Форма контроля. Практическое задание.

Тема 7. Функции и создание собственных программ (12 ч)

Теория. Назначение функций, параметры и возвращаемые значения, разбиение программы на логические части, повторное использование кода, базовые принципы читаемости и структурирования программ.
Практика. Создание собственных функций и объединение изученных конструкций Python в небольших прикладных программах и интерактивных проектах.
Форма контроля. Практическое задание.

- Разработка функций для вычислений и преобразования данных;
- Разбиение программы на отдельные логические блоки;
- Создание интерактивной программы с использованием функций;
- Проверка работы программы на разных наборах данных.
  Форма контроля. Практическое задание, наблюдение педагога.

Тема 8. Итоговое занятие. Комплексная задача и представление решения (8 ч)

Практика. Обучающийся выбирает тему небольшого проекта (например, викторина, текстовая игра, калькулятор или программа для обработки данных), формулирует задачу, составляет алгоритм, пишет программу, тестирует её и исправляет ошибки. Завершает работу краткая презентация проекта и объяснение использованных конструкций Python.
Форма контроля. Защита проекта.

МЕТОДИЧЕСКИЕ и ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ

Используемые технологии и методы:

1. ***Объяснительно-иллюстративный метод (демонстрация синтаксиса Python и пошагового решения задач).***
2. ***Репродуктивный метод (выполнение упражнений по образцу и алгоритму).***
3. ***Проблемно-поисковый метод (самостоятельный выбор алгоритма для решения задачи).***
4. ***Технология практико-ориентированного обучения (решение задач и разработка собственных программ).***
5. ***Здоровьесберегающая технология (соблюдение режима работы за компьютером, физкультминутки для глаз).***

Дидактические средства: презентации по темам, карточки с условиями задач, памятки по синтаксису Python, наборы тестовых данных, примеры программ и задания для самостоятельной работы.

Информационные источники:

Список литературы для педагога:

1. *Лутц М. Изучаем Python. – М.: Символ-Плюс. (справочное издание для педагога).*
2. *Официальная документация Python 3: https://docs.python.org/3/*

Список литературы для обучающихся:

1. *Учебные материалы по основам Python, алгоритмам и созданию небольших программ, рекомендованные педагогом.*

Интернет-источники:

1. *Официальная документация Python: https://docs.python.org/3/*

Оценочные материалы

Входной контроль.

На первом занятии проводится беседа/опрос для выявления начального уровня навыков работы с компьютером и представлений об алгоритмах и программировании. Результаты фиксируются в карте обучающегося.

Текущий контроль.

После каждой темы обучающиеся выполняют практические задания. Проверяются понимание изученных конструкций Python, корректность алгоритма, работоспособность программы, качество тестирования и умение объяснить решение. Результаты фиксируются в карте учёта.

Итоговый контроль.

На итоговом занятии обучающийся представляет собственный проект. Оценивается:

- *соответствие программы поставленной задаче;*
- *корректность алгоритма и применение изученных конструкций Python;*
- *качество тестирования, отладки и оформления кода;*
- *самостоятельность и умение объяснить работу программы.*

Параметры и критерии оценки

Критерии оценки	**Параметр: уровни освоения программы**		
	Низкий (1 балл)	Средний (2 балла)	Высокий (3 балла)
Понимание задачи	Не выделяет цель и данные задачи без постоянной помощи	Понимает условие с отдельными подсказками	Самостоятельно определяет цель, данные и ожидаемый результат
Алгоритм решения	Алгоритм неполный или приводит к неверному результату	Алгоритм в целом верный, но требует доработки	Самостоятельно составляет последовательный и корректный алгоритм
Использование Python	Использует конструкции с постоянной	Применяет основные конструкции с	Корректно использует изученные конструкции

	помощью	отдельными ошибками	Python
Тестирование и отладка	Не проверяет результат или не находит ошибки	Проверяет программу на простых примерах с подсказками	Подбирает тесты, находит ошибки и исправляет их
Самостоятельность	Работает только по пошаговой инструкции	Выполняет типовые задания с отдельными подсказками	Самостоятельно выбирает подход и доводит программу до результата
Представление проекта	Не может объяснить назначение и работу программы	Объясняет основные функции программы с помощью педагога	Последовательно объясняет алгоритм, функции программы и результаты тестирования

Уровень освоения программы:

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