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 (углубленная)»

Срок освоения: 36 недель (144 академических часа, 4 часа в неделю)
Возраст обучающихся: 14–16 лет

Разработчик:
Терехова Светлана Николаевна,
педагог дополнительного образования

ГБОУ лицей № 144
Калининского района
Санкт-Петербурга
Подписано электронной подписью
01.09.2026 16:36
директор
Федорова Лолита Анатольевна
7804140160-15-1793021832-20261026-298-1-1637-12

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Направленность программы – техническая.

Адресат программы: обучающиеся 14–16 лет, заинтересованные в углублённом изучении программирования.

Актуальность программы.

Актуальность программы связана с возрастающей ролью программирования в цифровой экономике, науке и инженерных технологиях. Python позволяет осваивать современные подходы к разработке программ и постепенно переходить от базовых конструкций к более сложным алгоритмам, обработке данных и проектированию приложений. Углублённый курс направлен на развитие алгоритмического мышления, самостоятельности, навыков анализа и решения задач разного уровня сложности.

Программа рассчитана на полный учебный год и предполагает расширенную практическую подготовку. Обучающиеся изучают сложные комбинации условий и циклов, строки и коллекции данных, функции, рекурсию, алгоритмы поиска и сортировки, работу с файлами и обработку исключений. Значительная часть занятий отводится решению алгоритмических задач, тестированию, анализу эффективности и разработке итогового проекта.

Уровень освоения программы – общекультурный.

Объем и срок освоения программы.

Программа рассчитана на 144 академических часа, 36 недель. Режим занятий: 2 раза в неделю по 2 академических часа (с перерывом 10–15 минут).

Цель: углублённое освоение языка Python и алгоритмических методов решения задач, развитие навыков проектирования, реализации, тестирования и оптимизации программ.

Задачи

Обучающие:

- закрепить правила безопасной работы за компьютером и культуру организации программных проектов;
- углубить знания синтаксиса Python, типов данных и принципов выполнения программ;
- научить применять сложные условия, вложенные циклы и составные алгоритмы;
- освоить строки, списки, кортежи, множества и словари для обработки данных;
- развить навыки использования функций, рекурсии, алгоритмов поиска и сортировки;
- научить работать с файлами, обрабатывать исключения, тестировать и оптимизировать 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 обучающиеся в возрасте 14–16 лет; желательно владение базовыми конструкциями Python.

Количество обучающихся в группе: не менее 15 человек.

Формы проведения занятий

Основная форма – учебное занятие. Используются объяснение и разбор алгоритмов, демонстрация кода, индивидуальная практическая работа, решение задач повышенной сложности, взаимопроверка, проектная деятельность и презентация результатов.

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, тестировать, отлаживать и улучшать программы на Python, применять алгоритмические методы и объяснять выбранное решение.

Учебный план

№ п/п	Название раздела, темы	Количество часов			Формы контроля/ аттестации
		Всего	Теория	Практика	
1	Углублённый синтаксис Python. Типы данных и сложные выражения	12	3	9	Практические задания
2	Условия, логические выражения и вложенные циклы	16	3	13	Практические задания
3	Циклы, перебор вариантов и алгоритмические задачи	20	4	16	Самостоятельная работа
4	Строки и коллекции: списки, кортежи, множества, словари	20	4	16	Практические задания
5	Функции, рекурсия и декомпозиция задач	20	4	16	Практические задания
6	Алгоритмы поиска и сортировки. Оценка эффективности	20	5	15	Решение задач
7	Файлы, исключения, модули и структурирование программ	16	4	12	Практическая работа
8	Итоговый проект: разработка, тестирование и презентация	20	5	15	Защита проекта
	Итого	144	32	112	

СОДЕРЖАНИЕ

Тема 1. Углублённый синтаксис Python. Типы данных и сложные выражения (12 ч)

Теория. Повторение базового синтаксиса, преобразование типов, составные выражения, приоритет операций, особенности целочисленного деления и остатка, форматирование вывода. Читаемость и стиль кода.
Практика. Решение вычислительных задач, преобразование данных, анализ выражений и написание программ с несколькими этапами вычислений.
Форма контроля. Практическое задание.

Тема 2. Условия, логические выражения и вложенные циклы (16 ч)

Теория. Составные логические выражения, вложенные условия, проверка нескольких ограничений, вложенные циклы и типичные ошибки в логике программы.
Практика. Задачи на диапазоны, делимость, классификацию данных, построение таблиц и обработку двумерных структур с помощью вложенных циклов.
Форма контроля. Практическое задание.

Тема 3. Циклы, перебор вариантов и алгоритмические задачи (20 ч)

Теория. Циклы for и while, вложенные циклы, накопители, флаги, раннее завершение, перебор диапазонов и вариантов, поиск закономерностей и базовая оценка количества операций.
Практика. Задачи на перебор, подсчёт, поиск подходящих чисел, обработку последовательностей, построение простых числовых и символьных закономерностей.
Форма контроля. Самостоятельная работа.

- Решение задач на подсчёт и накопление с несколькими условиями;
- Перебор диапазонов и поиск значений, удовлетворяющих ограничениям;
- Использование вложенных циклов для обработки таблиц и шаблонов;
- Сравнение алгоритмических подходов и проверка граничных случаев.
  Форма контроля. Практическое задание.

Тема 4. Строки и коллекции: списки, кортежи, множества, словари (20 ч)

Теория. Индексация и срезы строк, методы строк, списки и их изменение, кортежи, множества, словари, перебор элементов, подсчёт частот и выбор подходящей структуры данных.
Практика. Задачи на обработку текста, удаление повторов, подсчёт частоты символов и слов, поиск минимума и максимума, фильтрацию и группировку данных.
Форма контроля. Практическое задание.

- Обработка строк и поиск заданных символов или фрагментов;
- Обработка списков, фильтрация, подсчёт и преобразование элементов;
- Применение множеств и словарей для удаления повторов и подсчёта частот;

Тема 5. Функции, рекурсия и декомпозиция задач (20 ч)

Теория. Определение функций, параметры и возвращаемые значения, локальные переменные, разбиение задачи на подзадачи, повторное использование кода. Основы рекурсии и условия остановки.
Практика. Разработка набора функций для решения комплексных задач, реализация рекурсивных алгоритмов для небольших примеров, сравнение рекурсивного и итеративного подходов.
Форма контроля. Практическое задание.

- Создание функций для обработки чисел и последовательностей;
- Разбиение сложной задачи на функции и проверка каждой части;

- Реализация и трассировка простых рекурсивных алгоритмов. Форма контроля. Практическое задание.

Тема 6. Алгоритмы поиска и сортировки. Оценка эффективности (20 ч)

Теория. Линейный и бинарный поиск, простые алгоритмы сортировки, сортировка средствами Python, критерии выбора алгоритма, понятие времени выполнения и оценка количества операций на доступном уровне.
Практика. Реализация и сравнение алгоритмов поиска и сортировки, проверка результатов на разных наборах данных, решение задач на упорядочивание и выбор элементов. Форма контроля. Решение задач.

Тема 7. Файлы, исключения, модули и структурирование программ (16 ч)

Теория. Чтение и запись текстовых файлов, работа с контекстным менеджером with, обработка исключений try/except, подключение стандартных модулей, организация кода по функциям и файлам.
Практика. Создание программ для чтения, обработки и сохранения данных, обработка некорректного ввода, использование модулей и аккуратная организация файлов проекта. Форма контроля. Практическая работа.

- Чтение данных из текстового файла и обработка содержимого;
- Сохранение результатов работы программы в файл;
- Обработка ошибок ввода и исключительных ситуаций;
- Использование стандартных модулей и структурирование программы. Форма контроля. Практическое задание.

Тема 8. Итоговый проект: разработка, тестирование и презентация (20 ч)

Практика. Выбор задачи и формулировка требований, проектирование структуры программы, разработка алгоритма, реализация решения на Python, тестирование на различных данных, поиск и исправление ошибок. Подготовка краткой документации и презентация результата. Возможные проекты: текстовая игра, викторина с учётом результатов, анализатор текстовых данных, программа для обработки файлов или небольшая утилита. Форма контроля. Защита проекта.

МЕТОДИЧЕСКИЕ и ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ

Используемые технологии и методы:

1. ***Объяснительно-иллюстративный метод (разбор синтаксиса, структур данных и алгоритмов).***
2. ***Репродуктивный метод (выполнение упражнений с постепенным усложнением).***
3. ***Проблемно-поисковый метод (самостоятельный выбор и сравнение алгоритмов решения).***

4. ***Технология практико-ориентированного обучения (решение алгоритмических задач повышенной сложности и разработка проекта).***
5. ***Здоровьесберегающая технология (соблюдение режима работы за компьютером, физкультминутки для глаз).***

Дидактические средства: презентации, карточки с задачами, памятки по Python, наборы тестовых данных, примеры алгоритмов, задания повышенной сложности и материалы для проектной 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 уровня владения базовыми конструкциями Python. Результаты фиксируются в карте обучающегося.

Текущий контроль.

По завершении тематических разделов обучающиеся выполняют практические задания. Оцениваются корректность алгоритма, выбор структур данных, качество кода, тестирование и самостоятельность при отладке. Результаты фиксируются в карте учёта.

Итоговый контроль.

На итоговом занятии обучающийся представляет собственный программный проект. Оценивается:

- *соответствие программы поставленной задаче и корректность результата;*
- *обоснованность выбора алгоритмов и структур данных;*
- *качество тестирования, отладки и организации кода;*
- *самостоятельность, понятность объяснения и презентация проекта.*

Параметры и критерии оценки

Критерии оценки	**Параметр: уровни освоения программы**		
	Низкий (1 балл)	Средний (2 балла)	Высокий (3 балла)
Анализ условия задачи	Не выделяет ограничения и требуемый результат без постоянной помощи	Анализирует условие с отдельными подсказками	Самостоятельно выделяет данные, ограничения и критерии результата
Алгоритм и структуры	Алгоритм неполный или выбранная	Решение в целом верное, но требует	Выбирает подходящий алгоритм и обосновывает

данных	структура данных не подходит	доработки	структуру данных
Реализация на Python	Код работает только с постоянной помощью	Использует изученные конструкции, но допускает отдельные ошибки	Самостоятельно пишет понятный и корректный код
Тестирование и отладка	Не проверяет решение на разных данных	Проводит базовую проверку с подсказками	Подбирает тесты, проверяет граничные случаи и исправляет ошибки
Эффективность и структура кода	Не может объяснить структуру программы	Структурирует код частично, нуждается в рекомендациях	Разбивает программу на функции и может объяснить выбор решения
Проект и презентация	Проект не завершён или не соответствует задаче	Проект решает основную задачу, презентация требует поддержки	Проект работоспособен, протестирован и понятно представлен

Уровень освоения программы:

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