

**УПРАВЛЕНИЕ ОБРАЗОВАНИЯ АДМИНИСТРАЦИИ ГОРОДА
ТУЛЫ**

**МУНИЦИПАЛЬНОЕ БЮДЖЕТНОЕ ОБЩЕОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ
«ЦЕНТР ОБРАЗОВАНИЯ № 27»**

ПРИНЯТА
на заседании педагогиче-
ского
совета МБОУ ЦО № 27
Протокол от 29.08.2025г. № 1

УТВЕРЖДАЮ
Директор МБОУ ЦО № 27
_____ О.И.Маленков
Приказ от 01.09.2025г. №
587а

**Рабочая программа курса внеурочной деятельности
«Программирование на языке Python»**

9а,9в,19б – по 2 часа

Срок реализации: 1 год

Составитель:
Гузанова О.Ю.
педагог дополнительного образования ФГОС

2025г.

г. Тула

Пояснительная записка

Цель программы: обучить учащихся основам программирования на языке Python, развить алгоритмическое мышление и навыки разработки программ различной сложности.

Задачи программы:

1. Познакомить учащихся с основными понятиями и методами программирования, включая проектирование программ, работу с подпрограммами (процедурами и функциями), рекурсией, символьными строками, массивами и матрицами.
2. Развить умения применять полученные знания для решения практических задач, в том числе обработки данных, работы с файлами, использования различных структур данных (словари, стеки, очереди, деки, деревья, графы).
3. Сформировать навыки работы с алгоритмами сортировки и поиска, а также понимания их сложности и эффективности.
4. Познакомить с методами динамического программирования и их применением для решения комбинаторных и оптимизационных задач.
5. Развить умение анализировать и оптимизировать программы, используя методы рефакторинга.
6. Сформировать навыки разработки проектов и написания программ для решения нестандартных задач.

Содержание программы

Второй уровень (9 класс)

Программирование на языке Python

Этапы создания программ. Методы проектирования программ «сверху вниз» и «снизу вверх». Интерфейс и реализация. Документирование программы.

Подпрограммы: процедуры и функции. Процедуры. Процедуры с параметрами. Локальные и глобальные переменные.

Функции. Логические функции.

Рекурсия. Рекурсивные процедуры и функции. Фракталы.

Символьные строки. Сравнение строк. Операции со строками. Обращение к символам. Перебор всех символов. Срезы. Удаление и вставка. Встроенные методы. Поиск в символьных строках. Замена символов. Преобразования «строка — число». Символьные строки в функциях. Рекурсивный перебор.

Массивы (списки). Массивы в языке Python. Создание массива. Обращение к элементу массива. Перебор элементов массива. Генераторы. Вывод массива. Ввод массива с клавиатуры. Заполнение массива случайными числами.

Алгоритмы обработки массивов. Сумма элементов массива. Подсчёт элементов массива, удовлетворяющих условию. Особенности копирования списков в Python.

Поиск в массивах. Линейный поиск. Поиск максимального элемента в массиве. Максимальный элемент, удовлетворяющий условию. Использование массивов в прикладных задачах.

Матриц. Создание и заполнение матриц. Вывод матрицы на экран. Перебор элементов матрицы. Квадратные матрицы.

Сложность алгоритмов. Асимптотическая сложность.

Резерв – 2 часа.

Третий уровень (10 класс)

Программирование на языке Python

Сортировка массивов. Метод пузырька (сортировка обмeнами). Метод выбора. Сортировка слиянием. Быстрая сортировка. Сортировка в языке Python.

Двоичный поиск в массиве данных. Двоичный поиск по ответу.

Обработка файлов. Типы файлов. Чтение данных. Запись данных. Обработка данных из файла.

Целочисленные алгоритмы. Решето Эратосфена. Целочисленный квадратный корень.

Словари. Алфавитно-частотный словарь. Перебор элементов словаря.

Структуры. Классы. Создание структур. Работа с полями структур. Хранение структур в файлах. Сортировка структур.

Стек. Использование списка. Вычисление арифметических выражений. Скобочные выражения. Системный стек. Очередь. Дек.

Деревья. Деревья поиска. Обход дерева. Использование связанных структур. Вычисление арифметических выражений.

Графы. Описание графа. Жадные алгоритмы. Минимальное остовное дерево. Алгоритм Дейкстры. Алгоритм Флойда–Уоршелла. Использование списков смежности.

Динамическое программирование. Числа Фибоначчи. Количество программ для исполнителя. Двумерные задачи. Поиск оптимального решения.

Игровые модели. Выигрышные и проигрышные позиции.

Резерв – 2 часа.

Планируемые результаты

Второй уровень (9 класс)

В результате изучения курса на втором уровне учащийся

- 1) познакомится с методами проектирования программ «сверху вниз» и «снизу вверх»;
- 2) научится использовать вспомогательные алгоритмы (процедуры и функции) для структуризации программ;
- 3) научится применять рефакторинг для улучшения читаемости программ;
- 4) научится использовать символьные строки;
- 5) овладеет основными алгоритмами обработки одномерных и двумерных массивов;
- 6) познакомится с понятием сложности алгоритма;

Третий уровень (10 класс)

В результате изучения курса на третьем уровне учащийся

- 1) научится применять различные алгоритмы сортировки массивов;
- 2) научится использовать двоичный поиск;
- 3) научится обрабатывать данные, записанные в текстовые и двоичные файлы, и сохранять в файлах результаты работы программы;
- 4) научится использовать структуры для объединения данных;
- 5) научится применять словари, стеки, очереди, деки для решения задач обработки данных;
- 6) научится использовать деревья для организации данных;
- 7) познакомится с методами описания графов и некоторыми популярными алгоритмами на графах;
- 8) научится использовать динамическое программирование для решения комбинаторных и оптимизационных задач;
- 9) познакомится с понятием выигрышных и проигрышных позиций в играх с полной информацией;

Календарно-тематическое планирование

2 уровень 9 класс (68 часов)

Номер урока	Тема занятия	Кол-во часов		Всего	Форма контроля
		теория	практика		
	Программирование на языке Python				
1.	Проектирование программ	1	3	4	Практическая работа
2.	Процедуры	1	3	4	Практическая работа
3.	Рекурсия	1	3	4	Практическая работа
4.	Функции	1	3	4	Практическая работа
5.	Символьные строки	1	1	2	Практическая работа
6.	Обработка символьных строк	1	3	4	Практическая работа
7.	Строки в функциях	1	3	4	Практическая работа
8.	Массивы	1	1	2	Практическая работа
9.	Ввод и вывод массивов	1	1	2	Практическая работа
10.	Суммирование элементов массива	1	3	4	Практическая работа
11.	Подсчёт элементов массива, удовлетворяющих условию	1	3	4	Практическая работа
12.	Поиск значения в массиве	1	3	4	Практическая работа
13.	Поиск максимального элемента в массиве	1	3	4	Практическая работа
14.	Игра «Стрельба по тарелкам»	1	3	4	Практическая работа
15.	Матрицы	1	5	6	Практическая работа

Номер урока	Тема занятия	Кол-во часов		Всего	Форма контроля
		теория	практика		
16.	Сложность алгоритмов	1	3	4	Практическая работа
17.	Выполнение проекта	1	5	6	Защита проекта
18.	Резерв		2	2	

3 уровень 10 класс (34 часа)

Но-мер урока	Тема занятия	Кол-во часов		Всего	Форма контроля
		теория	практика		
	Программирование на языке Python				
1.	Простые алгоритмы сортировки	1	3	4	Практическая работа
2.	Сортировка слиянием	1	3	4	Практическая работа
3.	Быстрая сортировка	1	3	4	Практическая работа
4.	Двоичный поиск	1	3	4	Практическая работа
5.	Обработка файлов	1	1	2	Практическая работа
6.	Обработка файлов: практикум		4	4	Практическая работа
7.	Целочисленные алгоритмы	1	3	4	Практическая работа
8.	Словари	1	1	2	Практическая работа
9.	Структуры	1	1	2	Практическая работа
10.	Структуры: практикум		4	4	Практическая работа
11.	Стек, очередь, дек	1	1	2	Практическая работа
12.	Деревья	1	3	4	Практическая работа
13.	Графы	1	3	4	Практическая работа

Но- мер урока	Тема занятия	Кол-во часов		Всего	Форма контроля
		теория	прак- тика		
14.	Графы: практикум		4	4	Практическая работа
15.	Динамическое программирование	1	3	4	Практическая работа
16.	Динамическое программирование: практикум		6	6	Практическая работа
17.	Игровые модели	1	1	2	Практическая работа
18.	Игровые модели: практикум		6	6	Практическая работа
19.	Резерв		2	2	

Оценивание результатов обучения

Результатом обучения считается способность учащегося написать программу (разработать проект) определённого уровня сложности. Далее выделяются следующие уровни сложности:

- А:** начальный уровень, воспроизведение изучаемого материала с незначительными изменениями;
- В:** средний уровень, способно применять изученный материал для написания программ, которые отличаются от изученных;
- С:** высокий уровень, способно применять изученный материал для самостоятельного написания программ, решающих нестандартные задачи.

Второй уровень (9 класс)

Глава 1. Программирование на языке Python

§ 1. Проектирование программ

§ 2. Процедуры

- А:** Напишите процедуру, которая принимает параметр – натуральное число N – и выводит на экран квадрат из звёздочек размером $N \times N$ символов.
- В:** Напишите процедуру, которая выводит на экран в столбик все цифры переданного ей числа, начиная с последней.
- С:** Напишите процедуру, которая выводит на экран запись переданного ей числа в римской системе счисления.

§ 3. Рекурсия

- А:** Напишите рекурсивную процедуру, которая переводит число в троичную систему счисления.
- В:** Напишите рекурсивную процедуру, которая переводит число в шестнадцатеричную систему счисления.
- С:** Напишите рекурсивную процедуру, которая переводит число в негадвоичную систему счисления (систему с основанием -2).

§ 4. Функции

- А:** Напишите функцию, которая возвращает старшую цифру переданного ей натурального числа.
- В:** Напишите функцию, которая возвращает количество делителей переданного ей натурального числа.
- С:** Напишите логическую функцию, которая возвращает результат *True*, если переданное ей число – это число Фибоначчи.

§ 5. Символьные строки

- А:** Напишите программу, которая заменяет в символьной строке все точки на нули и все буквы «X» на единицы.
- В:** Напишите программу, которая выполняет инверсию битовой строки: заменяет в ней все нули на единицы и наоборот.
- С:** Напишите программу, которая вычисляет сумму неизвестного количества натуральных чисел, записанную в виде символьной строки, например, «1+25+12+34+89»

§ 6. Массивы (списки)

- А:** Напишите программу, которая вводит целое число X и заполняет массив последовательными натуральными числами, начиная с X .
- В:** Напишите программу, которая вводит целое число X и заполняет массив последовательными натуральными числами, начиная с X , в обратном порядке.
- С:** Напишите программу, которая заполняет массив из N элементов первыми N числами Фибоначчи.

§ 7. Алгоритмы обработки массивов

- А:** Напишите программу, которая определяет сумму элементов массива с чётными значениями.
- В:** Напишите программу, которая в массиве с чётным количеством элементов меняет местами пары соседних элементов.
- С:** Напишите программу, которая находит в массиве все числа Фибоначчи и строит из них новый массив.

§ 8. Поиск в массивах

- А:** Напишите программу, которая находит в массиве минимальный и максимальный элементы и их индексы.
- В:** Напишите программу, которая находит в массиве количество элементов, равных минимальному, и их индексы.
- С:** Напишите программу, которая находит в массиве два максимальных элемента и их индексы.

§ 9. Используем массивы

- А:** Закончите программу из параграфа.
- В:** Доработайте программу из параграфа так, чтобы тарелки двигались в другом направлении.
- С:** Доработайте программу из параграфа так, чтобы попадание в меньшую по размеру тарелку приводило к большему увеличению счёта.

§ 10. Матрицы

А: Напишите программу, которая находит максимальный элемент на главной диагонали квадратной матрицы.

В: Напишите программу, которая находит максимальный элемент матрицы и его индексы (номера строки и столбца).

С: Напишите программу, которая выполняет транспонирование матрицы.

§ 11. Сложность алгоритмов

Контроль не предусмотрен.

Третий уровень (10 класс)

Глава 1. Программирование на языке Python

§ 1. Простые алгоритмы сортировки

А: Напишите программу, которая выполняет сортировку массива методом «пузырька».

В: Напишите программу, которая выполняет сортировку массива методом «пузырька» в обратном порядке (сверху вниз).

С: Напишите программу, которая выполняет сортировку двух половин массива (отдельно) методом выбора.

§ 2. Быстрые алгоритмы сортировки

А: Напишите программу, которая выполняет сортировку массива по убыванию методом слияния.

В: Напишите программу, которая выполняет быструю сортировку массива символьных строк по убыванию длины.

С: Напишите программу, которая сравнивает количество операций при сортировке массива методами «пузырька» и слияния.

§ 3. Двоичный поиск

А: Напишите программу, которая находит в отсортированном массиве индексы всех элементов, равных заданному значению X .

В: Напишите программу, которая сравнивает количество проверок при линейном и двоичном поиске в отсортированном массиве.

С: Напишите программу, которая определяет среднее количество проверок при двоичном поиске в отсортированном массиве. Используя результаты работы этой программы, постройте зависимости количества проверок от размера массива.

§ 4. Обработка файлов

А: Напишите программу, которая вычисляет сумму чисел, записанных в файле в столбик.

В: Напишите программу, которая сортирует числа, записанные в файле в столбик.

С: Напишите программу, которая сортирует набор чисел, записанных в двух файлах в столбик. Отсортированные числа должны быть записаны в новый файл.

§ 5. Целочисленные алгоритмы

А: Напишите программу, которая вводит натуральное число N и находит все простые числа на отрезке $[2; N]$. Используйте алгоритм «решето Эратосфена».

В: Напишите программу, которая вводит натуральные числа M и N и находит все простые числа на отрезке $[M; N]$. Используйте алгоритм «решето Эратосфена».

С: Напишите программу, которая определяет количество операций, которые выполняются при решении задачи А. Используя результаты работы этой программы, постройте зависимость количества операций (или времени работы программы) от N .

§ 6. Словари

А: Напишите программу, которая строит алфавитно-частотный словарь для заданного файла, в котором каждое слово записано в отдельной строке.

В: Напишите программу, которая строит алфавитно-частотный словарь для заданного файла с произвольным текстом.

С: Напишите программу, которая сравнивает тексты разных авторов на основе частоты использования предлогов и союзов.

§ 7. Структуры

А: Напишите программу управления базой данных. Она должна поддерживать добавление, удаление и просмотр записей.

В: Напишите программу управления базой данных. Она должна поддерживать добавление, удаление, просмотр и поиск записей по ключу.

С: Напишите программу управления базой данных. Она должна поддерживать добавление, удаление, просмотр, поиск записей по ключу и сортировку по ключу.

§ 8. Стек, очередь, дек

А: Напишите программу, которая проверяет правильность скобочного выражения с тремя типами скобок.

В: Напишите программу, которая вычисляет значение арифметического выражения, записанного в префиксной форме.

С: Напишите программу, которая применяет алгоритм заливки области для поиска пути в лабиринте.

§ 9. Деревья

А: Напишите программу, которая вычисляет введённое арифметическое выражение без скобок.

В: Напишите программу, которая вычисляет введённое арифметическое выражение со скобками.

С: Напишите программу, которая вычисляет введённое арифметическое выражение со скобками и функциями (*sin*, *cos*, *sqrt*, *abs*).

§ 10. Графы

А: Напишите программу, которая строит минимальное остовное дерево для неориентированного графа.

В: Напишите программу, которая определяет кратчайший маршрут из одной вершины графа в другую с помощью алгоритма Дейкстры.

С: Напишите программу, которая определяет кратчайшие маршрут между каждой парой вершин графа с помощью алгоритма Флойда-Уоршелла.

§ 11. Динамическое программирование

А: Напишите программу, которая вычисляет количество битовых цепочек длины N , в которых нет двух единиц подряд.

В: Напишите программу, которая вычисляет количество битовых цепочек длины N , в которых нет трёх единиц подряд.

С: Напишите программу, которая решает задачу о ранце.

§ 12. Игровые модели

А: Напишите программу, которая играет с человеком в игру Баше.

В: Напишите программу, которая определяет тип позиции (выигрышная или проигрышная) в игре «крестики-нолики» на доске 3×3 .

С: Напишите программу, которая играет с человеком в «крестики-нолики».

Список литературы

1. Программирование. Python. С. Часть 2. Учебное пособие, Поляков К. Ю.
2. Программирование. Python. С. Часть 3. Учебное пособие, Поляков К. Ю.

Ресурсы по языку Python

- www.python.org — официальный сайт поддержки языка Python, дистрибутивы для различных операционных систем;
- wingware.com — *Wing IDE 101* — бесплатная среда программирования на Python;
- sourceforge.net/projects/pyscripter/ — *PyScripter* — бесплатная среда программирования на Python;
- www.jetbrains.com/pycharm/ — *PyCharm* — среда программирования на Python, версия *Community* — бесплатная;
- www.pyinstaller.org — программа *PyInstaller* для преобразования скриптов на языке Python в исполняемые файлы;
- pypi.org/project/Pillow — библиотека *Pillow* для работы с изображениями в Python;
- www.numpy.org — пакет для научных исследований в Python, содержащий быстрые алгоритмы обработки матриц;
- **модуль [graph](#)** для создания простых графических программ на языке Python (автор — *К.Ю. Поляков*);
- **модуль [simpletk](#)** для создания программ с графическим интерфейсом на языке Python (автор — *К.Ю. Поляков*).
- www.pygame.org — библиотека *PyGame* для программирования игр на языке Python;
- interactivepython.org — «Алгоритмы и структуры данных с использованием Python» (бесплатная книга с интерактивным тренажёром);
- younglinux.info/oopython.php — Python. Введение в объектно-ориентированное программирование;
- wxpython.org — библиотека *wxPython* для разработки программ с графическим интерфейсом;
- pygtk.org — библиотека *PyGTK* для разработки программ с графическим интерфейсом;
- www.riverbankcomputing.com/software/pyqt/intro — библиотека *PyQt* для разработки программ с графическим интерфейсом;
- younglinux.info/tkinter.php — *Tkinter*. Программирование графического интерфейса;