

МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Министерство образования Самарской области

МУНИЦИПАЛЬНОЕ БЮДЖЕТНОЕ ОБЩЕОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

« Школа № 94 имени полного кавалера ордена Славы Щеканова Н.Ф.»

городского округа Самара

РАССМОТРЕНО

Председатель МО
учителей естественно-
математического цикла

Третьякова Н.Н.
Протокол №1 от «27» августа
2026 г.

ПРОВЕРЕНО

Зам. директора по УВР

Горшкова О.Б.

УТВЕРЖДЕНО

Директор МБОУ "Школа
№94" г.о. Самара

Кольчугина А.А.
Приказ №170-од от «31»
августа 2026 г.

Элективный курс

Информатика без границ

учебного курса «Информатика»

для обучающихся 10-11 классов

Самара 2026

Пояснительная записка

Нормативная база

Программа разработана в соответствии со следующими документами:

1. Федеральный государственный образовательный стандарт среднего общего образования (ФГОС СОО).
2. Примерная основная образовательная программа среднего общего образования.
3. Концепция преподавания предметной области «Информатика».
4. Методические рекомендации ФИПИ по подготовке к ЕГЭ по информатике.
5. Локальный акт образовательной организации о порядке разработки рабочих программ.
6. Учебный план образовательной организации на текущий учебный год.
7. Санитарные правила и нормы (СанПиН) к организации образовательного процесса.

Общая характеристика курса

Курс носит практико-ориентированный характер и направлен на формирование у обучающихся навыков программирования на языке Python, основ алгоритмического мышления, а также умений разрабатывать простые программные продукты (скрипты, консольные приложения, веб-сервисы, инструменты анализа данных). В содержании курса сочетаются базовые понятия информатики и современные технологии разработки.

Цели и задачи курса

Цель: формирование компетенций в области программирования и разработки приложений для решения прикладных задач, подготовки к профильному обучению и участию в олимпиадах по информатике.

Задачи:

- освоить синтаксис и базовые конструкции языка Python;
- научиться разрабатывать, реализовывать и анализировать алгоритмы;
- познакомиться с принципами объектно-ориентированного программирования (ООП);
- получить навыки веб-разработки на базовом уровне;
- освоить методы обработки и визуализации данных;
- развить навыки проектной деятельности, командной работы и презентации результатов.

Место курса в учебном плане

Курс реализуется как элективный в 10 и 11 классах. Общий объём — 68 часов (по 34 часа в каждом классе, 1 час в неделю). Курс дополняет базовую программу по информатике, усиливая практическую составляющую и ориентируя обучающихся на применение знаний в реальных задачах.

Формы организации учебной деятельности

- фронтальная работа (объяснение, демонстрация кода, разбор задач);
- индивидуальная работа (написание кода, отладка, тестирование);
- групповая работа (совместная разработка проектов, ревью кода);
- проектная деятельность (разработка и защита итогового проекта).

Формы и методы контроля

- текущий контроль: практические задания, мини-проекты, тесты по темам;
- промежуточный контроль: защита промежуточных результатов проекта;
- итоговый контроль: защита итогового проекта с демонстрацией функционала и пояснительной запиской.

Технологии обучения

- проблемное обучение (решение задач с неоднозначными условиями);
 - обучение через практику (code-first подход);
 - кейс-метод (разбор реальных сценариев использования кода);
 - метод проектов (разработка собственных приложений).
-

Планируемые результаты освоения курса

Личностные результаты

- готовность к самообразованию и непрерывному развитию в области IT;
- осознание ценности алгоритмического мышления для повседневной жизни и профессиональной деятельности;
- развитие навыков критического мышления и анализа информации;
- формирование ответственного отношения к информационной безопасности и этике использования технологий.

Метапредметные результаты

Регулятивные УУД:

- умение ставить учебную задачу и планировать её решение;
- способность корректировать план действий в зависимости от промежуточных результатов;
- навыки самооценки и рефлексии.

Познавательные УУД:

- умение искать, анализировать и структурировать информацию для решения задач;
- владение методами обработки данных и визуализации результатов;
- способность выбирать подходящие алгоритмы и структуры данных для конкретной задачи.

Коммуникативные УУД:

- навыки работы в команде, распределение ролей и ответственности;
- умение презентовать результаты работы, аргументировать решения;
- способность давать и принимать конструктивную критику.

Предметные результаты

10 класс («научится»)

- писать программы на Python средней сложности с использованием базовых конструкций;
- реализовывать линейные, разветвляющиеся и циклические алгоритмы;
- работать со структурами данных (списки, словари, множества);
- создавать и использовать функции, модули, обрабатывать исключения;
- читать и записывать файлы, работать с JSON/CSV, делать простые запросы к API.

10 класс («получит возможность научиться»)

- проектировать простые приложения с модульной структурой;
- применять принципы тестирования кода и отладки;
- использовать системы контроля версий (Git) для совместной работы.

11 класс («научится»)

- применять принципы ООП (классы, объекты, наследование, полиморфизм);
- разрабатывать простые веб-приложения на Flask;

- анализировать и визуализировать данные с помощью NumPy, Pandas, Matplotlib;
- реализовывать и сравнивать алгоритмы сортировки и поиска;
- оценивать алгоритмическую сложность решений.

11 класс («получит возможность научиться»)

- разрабатывать полноценные проекты с разделением на фронтенд и бэкенд;
 - применять паттерны проектирования на базовом уровне;
 - автоматизировать рутинные задачи и строить пайплайны обработки данных.
-

Содержание курса

10 класс (34 часа)

Раздел 1. Введение в программирование (4 часа)

- История языков программирования, классификация языков, парадигмы.
- Установка и настройка среды разработки (Python, IDE).
- Первая программа, вывод и ввод данных, переменные и типы данных.
- Операторы и выражения, форматированный вывод, работа с консолью.

Раздел 2. Основы алгоритмизации (6 часов)

- Понятие алгоритма, свойства и способы записи (псевдокод, блок-схемы).
- Линейные алгоритмы, ветвления (if/elif/else), логические выражения.
- Циклы (for, while), вложенные циклы, операторы break/continue.
- Трассировка алгоритмов, решение задач на построение алгоритмов.

Раздел 3. Работа с данными и структурами (6 часов)

- Списки: создание, индексация, срезы, методы, перебор элементов.
- Кортежи, множества, словари: сравнение структур, выбор подходящей структуры.
- Фильтрация и преобразование данных, работа с вложенными структурами.
- Практические задачи на обработку коллекций.

Раздел 4. Функции и модули (6 часов)

- Определение и вызов функций, параметры и аргументы, возвращаемые значения.
- Область видимости переменных, локальные и глобальные переменные.

- Рекурсия, обработка исключений (try/except).
- Создание собственных модулей, импорт библиотек, организация кода.

Раздел 5. Работа с файлами и внешними данными (4 часа)

- Чтение и запись текстовых файлов, режимы открытия, контекстный менеджер with.
- Работа с CSV и JSON, парсинг данных, обработка ошибок.
- Основы работы с API, отправка HTTP-запросов, получение данных.
- Практические задания на автоматизацию обработки файлов.

Раздел 6. Проектная работа (8 часов)

- Выбор темы, планирование, разработка, тестирование.
 - Оформление документации, подготовка к защите.
 - Защита проекта, обратная связь, рефлексия.
-

11 класс (34 часа)

Раздел 1. Объектно-ориентированное программирование (8 часов)

- Классы и объекты, атрибуты и методы, инкапсуляция.
- Наследование, переопределение методов, полиморфизм.
- Магические методы, абстрактные классы.
- Проектирование классов, паттерны проектирования (на базовом уровне).
- Практикум по проектированию и реализации классов.

Раздел 2. Веб-разработка (6 часов)

- Основы HTML/CSS, структура веб-страницы.
- Введение в Flask, маршрутизация, обработка запросов.
- Шаблоны (Jinja2), формы, передача данных.
- Разработка мини-веб-приложения, развертывание на локальном сервере.
- Практикум по созданию простого веб-сервиса.

Раздел 3. Анализ и визуализация данных (6 часов)

- Введение в NumPy и Pandas, загрузка и очистка данных.
- Агрегация, группировка, фильтрация, работа с пропущенными значениями.
- Визуализация с помощью Matplotlib, построение графиков и диаграмм.
- Анализ реального набора данных, формулирование выводов.

- Практикум по обработке и визуализации данных.

Раздел 4. Алгоритмы и структуры данных (8 часов)

- Алгоритмическая сложность, нотация O, оценка эффективности.
- Сортировка (пузырьковая, быстрая, слиянием), сравнение эффективности.
- Поиск (линейный, бинарный), реализация и анализ.
- Структуры данных: стеки, очереди, деревья, графы.
- Практикум по реализации и применению структур данных.

Раздел 5. Проектная работа (6 часов)

- Выбор темы, планирование, разработка.
- Тестирование, демонстрация, защита проекта.
- Рефлексия, анализ результатов, оформление портфолио.

Тематическое планирование

10 класс

№	Тема	Количество часов	Теория	Практика	Формы контроля
1	Введение в программирование	4	2	2	Практическая работа, тест
2	Основы алгоритмизации	6	2	4	Практические задания, мини-проект
3	Работа с данными и структурами	6	2	4	Практическая работа, самопроверка
4	Функции и модули	6	2	4	Практические задания, код-ревью
5	Работа с файлами и внешними данными	4	1	3	Практическая работа, анализ ошибок
6	Проектная работа	8	1	7	Защита проекта, презентация
	Итого	34	10	24	

11 класс

№	Тема	Количество часов	Теория	Практика	Формы контроля
1	Объектно-ориентированное программирование	8	3	5	Практические задания, код-ревью
2	Веб-разработка	6	2	4	Мини-проект, демонстрация
3	Анализ и визуализация данных	6	2	4	Практическая работа, отчёт
4	Алгоритмы и структуры данных	8	3	5	Практические задания, тест
5	Проектная работа	6	1	5	Защита проекта, портфолио
	Итого	34	11	23	

Поурочное планирование

10 класс

Урок	Тема урока	Тип урока	Основные виды деятельности	Оборудование и ПО	Домашнее задание
1	Знакомство с Python. Установка среды разработки	Комбинированный	Установка Python и IDE, запуск первой программы, вывод текста, переменные и типы данных	ПК, Python, PyCharm/VS Code	Установить среду, написать 3 простые программы
2	Ввод данных и простые вычисления	Практический	Работа с input(), арифметические операции, форматированный вывод	ПК, Python	Написать калькулятор для 3 операций
3	Условные операторы и логические	Комбинированный	if/elif/else, логические операторы,	ПК, Python	Решить 5 задач на условия

Урок	Тема урока	Тип урока	Основные виды деятельности	Оборудование и ПО	Домашнее задание
	выражения		задачи на ветвления		
4	Циклы: for и while	Практический	Написание циклов, вложенные циклы, задачи на перебор	ПК, Python	Реализовать вывод таблицы умножения
5	Списки и операции над ними	Практический	Создание списков, индексация, срезы, методы списков	ПК, Python	Обработать список чисел (сумма, среднее, максимум)
6	Кортежи, множества и словари	Практический	Сравнение структур данных, выбор подходящей структуры	ПК, Python	Преобразовать данные между структурами
7	Функции: определение и вызов	Комбинированный	Создание функций, передача аргументов, возвращаемые значения	ПК, Python	Написать 3 функции для разных задач
8	Область видимости и рекурсия	Комбинированный	Локальные и глобальные переменные, примеры рекурсивных функций	ПК, Python	Реализовать рекурсивную функцию (факториал/числа Фибоначчи)
9	Модули и работа с библиотеками	Практический	Импорт модулей, использование встроенных и сторонних библиотек	ПК, Python	Подключить библиотеку и использовать её функции
10	Обработка исключений	Практический	try/except, обработка ошибок ввода и работы с файлами	ПК, Python	Написать программу с обработкой ошибок
11	Чтение и запись файлов	Практический	Работа с текстовыми файлами, режимы	ПК, Python	Прочитать файл, обработать данные,

Урок	Тема урока	Тип урока	Основные виды деятельности	Оборудование и ПО	Домашнее задание
			открытия, with		записать результат
12	Работа с CSV и JSON	Практический	Чтение и запись данных в форматах CSV и JSON, парсинг	ПК, Python	Обработать CSV-файл, вывести статистику
13	Основы работы с API	Практический	Отправка HTTP-запросов, получение данных из открытых API	ПК, Python, requests	Получить данные из API, вывести в консоль
14	Повторение и подготовка к проекту	Обобщающий	Разбор типовых задач, выбор тем для проектов	ПК, Python	Выбрать тему проекта, составить план
15–22	Проектная работа (блок 1)	Проектный	Планирование, разработка, тестирование проекта	ПК, Git, Python	Выполнить этапы проекта по плану
23–34	Проектная работа (блок 2)	Проектный	Доработка, оформление, подготовка к защите, защита проектов	ПК, Git, Python	Оформить документацию, подготовить презентацию

11 класс

Урок	Тема урока	Тип урока	Основные виды деятельности	Оборудование и ПО	Домашнее задание
1	Классы и объекты в Python	Комбинированный	Определение классов, создание объектов, атрибуты и методы	ПК, Python, Flask	Написать простой класс с методами
2	Инкапсуляция и наследование	Практический	Реализация инкапсуляции, создание подклассов,	ПК, Python	Создать иерархию классов (родитель + 2

Урок	Тема урока	Тип урока	Основные виды деятельности	Оборудование и ПО	Домашнее задание
			переопределение методов		потомка)
3	Полиморфизм и абстрактные классы	Комбинированный	Понимание полиморфизма, работа с абстрактными классами	ПК, Python	Реализовать полиморфное поведение для разных классов
4–5	Практикум по ООП	Практический	Решение задачи на проектирование классов и взаимодействие объектов	ПК, Python	Разработать мини-систему (например, каталог товаров)
6–7	Введение в веб-разработку. HTML/CSS	Практический	Основы разметки и стилизации, создание простой страницы	ПК, браузер, редактор кода	Сверстать страницу с формой и таблицей
8–9	Flask: основы и маршрутизация	Практический	Установка Flask, создание маршрутов, обработка запросов	ПК, Flask, Python	Создать веб-приложение с 3 маршрутами
10	Шаблоны и формы в Flask	Практический	Использование Jinja2, обработка форм, передача данных	ПК, Flask, Python	Добавить форму в приложение, обработать данные
11–12	Практикум по веб-разработке	Практический	Разработка мини-веб-приложения с сохранением данных	ПК, Flask, SQLite	Реализовать CRUD-операции для простой сущности
13–14	Введение в NumPy и Pandas	Практический	Загрузка данных, базовые операции, фильтрация и агрегация	ПК, Jupyter, NumPy, Pandas	Загрузить CSV, выполнить анализ данных
15–16	Визуализация данных с Matplotlib	Практический	Построение графиков, диаграмм,	ПК, Matplotlib	Построить 3 разных графика для

Урок	Тема урока	Тип урока	Основные виды деятельности	Оборудование и ПО	Домашнее задание
			настройка стилей и подписей		одного набора данных
17–18	Практикум по анализу данных	Практический	Анализ реального набора данных, построение выводов	ПК, Pandas, Matplotlib	Провести анализ данных, оформить отчёт
19	Алгоритмическая сложность	Комбинированный	Понятие О-нотации, оценка сложности алгоритмов	ПК, Python	Оценить сложность 3 алгоритмов
20–21	Сортировка и поиск	Практический	Реализация алгоритмов сортировки и поиска, сравнение эффективности	ПК, Python	Реализовать 2 алгоритма сортировки, сравнить время работы
22–23	Структуры данных: стеки, очереди, деревья	Комбинированный	Реализация и применение структур данных	ПК, Python	Реализовать стек и очередь, применить в задаче
24–25	Графы и алгоритмы на графах	Практический	Представление графов, поиск пути, обходы	ПК, Python	Реализовать поиск пути в графе (DFS/BFS)
26–34	Проектная работа	Проектный	Выбор темы, разработка, тестирование, защита проекта	ПК, Git, Python, Flask, Pandas	Разработать и защитить итоговый проект

Учебно-методическое обеспечение

Литература для учителя

- К. Ю. Поляков «Информатика. Углублённый уровень» (10–11 класс).
- М. Лутц «Изучаем Python» (для углублённого изучения языка).

- Документация по Flask, NumPy, Pandas, Matplotlib (официальные сайты).
- Методические рекомендации ФИПИ по подготовке к ЕГЭ по информатике.

Литература для учеников

- Учебники по информатике базового и углублённого уровня.
- Онлайн-курсы и tutorиалы по Python, Flask, анализу данных.
- Документация Python и библиотек (официальные ресурсы).

Цифровые ресурсы

- Официальная документация Python.
- GitHub для хранения и совместной работы над проектами.
- Онлайн-компиляторы и песочницы для тестирования кода.
- Открытые наборы данных для анализа (Kaggle, Open Data).

Материально-техническое обеспечение

- Компьютеры с установленным Python 3.x и необходимыми библиотеками.
- Доступ к интернету для работы с API и документацией.
- Среда разработки (PyCharm, VS Code или аналоги).
- Проектор и экран для демонстрации