

МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Министерство образования Самарской области

Автономная некоммерческая образовательная организация средняя

общеобразовательная школа «Сота»

АНОО СОШ "СОТА"

РАССМОТРЕНА

на заседании кафедры
ЕМН _____

Очеповский А.В.

Протокол №1
от «21» августа 2026 г.

ПРИНЯТО

на заседании
педагогического
совета

Протокол №1
от «21» августа 2026 г.

УТВЕРЖДЕНА

Директор АНОО СОШ
“Сота” _____ Гендель И.А.

Приказ № 42-ОД
от «21» августа 2026 г.



СОТА
Школа нового
поколения

O="АНОО СОШ ""
СОТА""", CN=" "
Гендель Ирина
Абрамовна, директор",
E=sota_school@mail.ru
0x 00 FE BB 13 5A 73
87 C4 F5
2026.08.26
15:32:54
+04'00'

**РАБОЧАЯ ПРОГРАММА КУРСА ВНЕУРОЧНОЙ
ДЕЯТЕЛЬНОСТИ**

«Информационные технологии»

для обучающихся 8 классов

г.о.Тольятти, 2026

1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Рабочая программа курса внеурочной деятельности «Информационные технологии» для 8 класса составлена в соответствии с требованиями следующих нормативно-правовых и программно-методических документов:

1. Федеральный закон от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации».
2. Федеральный государственный образовательный стандарт основного общего образования, утвержденный Приказом Минпросвещения России от 31.05.2021 № 287 (в ред. Приказа Минпросвещения России от 18.07.2022 № 568).
3. Стратегия научно-технологического развития Российской Федерации (Указ Президента РФ от 28.02.2024 № 145).
4. Национальная стратегия развития искусственного интеллекта на период до 2030 года (Указ Президента РФ от 10.10.2019 № 490).
5. Приказ Минпросвещения России от 26.06.2025 № 495 «Об утверждении федерального перечня учебников...».
6. Основная образовательная программа основного общего образования и локальные нормативные акты АНОО СОШ «Сота» (г. Тольятти).

Общая характеристика и преемственность курса

Курс «Информационные технологии» в 8 классе представляет собой звено сквозной непрерывной цифровой подготовки школьников. Программа реализует практико-ориентированный и системно-деятельностный подходы, сочетая работу с современными системами генеративного искусственного интеллекта (LLM) и алгоритмическую подготовку на базе языка программирования Python.

Ключевой методический принцип курса базируется на двухэтапной модели:

- **1-е полугодие (Модули 1–2): «ИИ как ассистент и аналитик кода».** Самостоятельный кодирование вынесено за рамки этапа: обучающиеся осваивают промпт-инжиниринг, Сократовский диалог, реверс-инжиниринг алгоритмов, учатся дебаггингу и построчному анализу структуры кода, сгенерированного ИИ, без когнитивного барьера запоминания синтаксиса.
- **2-е полугодие (Модули 3–4): «Алгоритмизация, синтаксис Python и проектная разработка».** Опираясь на сформированное структурное и логическое мышление, учащиеся переходят к кодированию на Python (базовые типы данных, условные конструкции, циклы **while** и **for**, срезы строк, методы списков и функции) с последующей разработкой и защитой итогового проекта.

Цели и задачи курса

Цели курса:

- сформировать умения применять системы генеративного ИИ для академического анализа, самообразования и эффективного освоения предметов;
- освоить синтаксис и базовые алгоритмические структуры языка программирования Python для решения прикладных задач;
- развить системное, критическое, алгоритмическое и проектное мышление обучающихся.

Задачи курса:

- обучить технологиям диалогового взаимодействия с ИИ (Сократовский метод, Chain of Thought, метод «перевернутого урока»);

- сформировать устойчивые навыки чтения, смыслового комментирования и поиска ошибок (дебаггинга) в коде;
- освоить работу с переменными, ветвлениями, циклами, строками, списками и функциями в среде Python;
- сформировать навыки проектирования, написания, тестирования и публичной защиты консольного интерактивного проекта.

Место курса в учебном плане

Курс реализуется за счет часов плана внеурочной деятельности АНОО СОШ «Сота».

- Общий объем учебной нагрузки: **34 часа в год** (1 академический час в неделю при 34 учебных неделях).
- Структура занятия: вводно-мотивационная часть и теория (5–10 минут), практическая работа за ПК в среде разработки/ИИ-сервисе (25–30 минут), рефлексия и подведение итогов (5 минут).

2. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОСВОЕНИЯ КУРСА

Личностные результаты

- **Гражданское и патриотическое воспитание:** понимание роли цифрового суверенитета и вклада отечественных IT-разработок (GigaChat, YandexGPT, отечественные образовательные платформы) в развитие современной экономики страны.
- **Ценности научного познания:** развитие критического мышления, умения проверять фактологическую точность («галлюцинации») ИИ-моделей, исследовательская ориентация в работе с данными.
- **Трудовое воспитание и профориентация:** осознанное представление о современных профессиях цифровой индустрии (инженер по машинному обучению, промпт-инженер, Python-разработчик, специалист по тестированию).

- **Культура здоровья и безопасность:** строгое соблюдение эргономики работы за компьютером, правил информационной безопасности и защиты персональных данных при взаимодействии с веб-сервисами.

Метапредметные результаты (универсальные учебные действия)

Познавательные УУД:

- *Базовые логические действия:* выявлять закономерности, проводить аналогии, декомпозировать комплексные задачи на элементарные шаги, строить логические выводы.
- *Базовые исследовательские действия:* выдвигать гипотезы о причинах некорректной работы алгоритма, планировать эксперимент по тестированию программы на граничных условиях.
- *Работа с информацией:* структурировать текстовую информацию в виде таблиц, блок-схем и ментальных карт (Mind Maps), критически сопоставлять данные из независимых источников.

Коммуникативные УУД:

- *Общение:* формулировать четкие контекстные запросы (промнты), аргументировать технические решения при защите программного кода.
- *Совместная деятельность:* продуктивно взаимодействовать в парах при дебаггинге кода, соблюдать нормы цифровой этики.

Регулятивные УУД:

- *Самоорганизация:* составлять пошаговый план решения алгоритмической задачи, распределять временные ресурсы при выполнении проекта.
- *Самоконтроль:* использовать методы пошаговой трассировки и логического анализа ошибок, оперативно вносить корректировки в код по результатам тестов.

Предметные результаты

К концу обучения в 8 классе обучающийся научится:

- настраивать диалоговые роли и контекст в нейросетевых моделях для реализации сценария персонального ИИ-тьютора;
- читать чужой программный код на языке Python, формулировать словесный алгоритм его работы, выявлять синтаксические и логические ошибки (дебаггинг);
- работать в специализированной среде разработки (Thonny IDE, IDLE Python);
- корректно применять базовые типы данных (`int`, `float`, `str`, `bool`) и стандартные математические и логические операции над ними;
- организовывать ввод данных (`input()`) с явным преобразованием типов и форматированный вывод (`print()`, f-строки);
- реализовывать линейные алгоритмы, неполные и полные ветвления (`if - else`), множественный выбор (`if - elif - else`) и составные логические условия (`and`, `or`, `not`);
- программировать циклы с предусловием (`while`) и со счетчиком (`for` с функцией `range()`);
- обрабатывать строки (индексация, срезы, базовая длина `len()`) и списки (`list`: создание, индексация, методы `append()`, `remove()`, `sort()`, поиск минимума/максимума/суммы);
- создавать и вызывать пользовательские функции (`def`) с параметрами и возвратом результата (`return`);
- проектировать, реализовывать, тестировать и публично презентовать законченный консольный интерактивный проект.

3. СОДЕРЖАНИЕ КУРСА ВНЕУРОЧНОЙ ДЕЯТЕЛЬНОСТИ (34 часа)

Раздел 1. ИИ в образовании: Сократовский метод и самообучение (8 часов)

Понятие больших языковых моделей (LLM). Концепция персонального ИИ-тьютора. Промпт-инжиниринг: структура эффективного запроса (роль, контекст, ограничения, формат). Сократовский метод: настройка ИИ на обучение через наводящие вопросы без выдачи прямых ответов. Методология «перевернутого урока»: проверка собственного объяснения учебного материала нейросетью. Техника Ричарда Фейнмана: многоуровневая адаптация сложных понятий для разной целевой аудитории. Автоматизированная генерация адаптивных тестов, квизов и тренажеров с динамической сложностью. Ментальные карты (Mind Maps): визуализация сущностей и взаимосвязей. Мнемотехники и карточки интервального повторения.

Раздел 2. ИИ в анализе кода и сложной информации (8 часов)

Программный код как формализованный язык управления исполнителем. Постановка технических заданий и генерация кода с помощью ИИ. Разбор анатомии кода: структура, логика, построчное смысловое аннотирование. Дебаггинг: выявление синтаксических ошибок, исключений времени выполнения и логических несоответствий с помощью нейросетей. Реверс-инжиниринг: перевод кода в словесные пошаговые алгоритмы и блок-схемы. Метод цепочки рассуждений (Chain of Thought) для решения олимпиадных и логических задач. Аналитическая обработка текстовых массивов, сведение разнородных данных в структурированные таблицы. Практикум конструирования интерактивного сценария текстовой игры через промптинг.

Раздел 3. Основы алгоритмизации и программирования на языке Python (10 часов)

Среда разработки Thonny/IDLE: интерфейс, запуск скриптов, режим интерактивной оболочки. Функция `print()`: позиционные аргументы, параметры `sep` и `end`. Переменные, динамическая типизация, правила именования идентификаторов. Числовые типы (`int`, `float`), базовые арифметические операции (`+`, `-`, `*`, `/`, `//`, `%`, `**`). Ввод данных с клавиатуры: функция `input()`, приведение типов данных `int()` и `float()`. Логический тип `bool`, операции сравнения (`>`, `<`, `>=`, `<=`, `==`, `!=`). Условный оператор ветвления `if`, блок `else`, каскадное ветвление `if - elif - else`. Сложные логические условия с операторами `and`, `or`, `not`. Цикл с предусловием `while`: условие продолжения работы, переменная-счетчик, защита от зацикливания. Цикл со счетчиком `for`, генерация числовых интервалов через `range(start, stop, step)`. Контрольный практикум по решению базовых алгоритмических задач.

Раздел 4. Разработка алгоритмов и проектов на Python (8 часов)

Строки (`str`): индексация символов, получение срезов `[start:stop:step]`, функция `len()`. Списки (`list`): понятие динамического массива, доступ к элементам по индексу. Базовые методы списков: добавление элементов (`append()`), удаление (`remove()`), упорядочивание (`sort()`). Типовые алгоритмы обработки списков: вычисление суммы, поиск минимального и максимального значений, подсчет среднего арифметического. Подпрограммы: объявление пользовательских функций через `def`, аргументы функций, оператор возврата значения `return`. Жизненный цикл разработки программного обеспечения: анализ требований, проектирование архитектуры, написание кода, функциональное тестирование. Разработка итогового консольного интерактивного проекта. Оформление кода в соответствии со стандартом PEP 8. Публичная защита проектов, демонстрация работы программы, рефлексия результатов курса.

4. ТЕМАТИЧЕСКОЕ ПЛАНИРОВАНИЕ (8 класс, 34 часа)

№ п/п	Наименование разделов и тем	Кол-во часов	Форма проведения	Основные виды деятельности обучающихся	Электронные (цифровые) образовательные ресурсы
Раздел 1	ИИ в образовании: Сократовский метод и самообучение	8			
1.1	Архитектура ИИ-ассистента и базовый промпт-инжиниринг	1	Беседа, практикум	Составление ролевых запросов, задание контекста и формата ответа модели.	Диалоговая платформа GigaChat (https://gigachat.sber.ru), портал «Цифровой ликбез» (https://digital-likbez.datalesson.ru).
1.2	Сократовский диалог: обучение через наводящие вопросы	1	Практикум	Интерактивная диалоговая сессия с ИИ по разбору темы школьного курса без прямых подсказок.	Диалоговая среда YandexGPT (https://ya.ru/ai), тренажер промптинга.
1.3	Метод «Перевернутого урока» с критическим анализом ИИ	1	Практикум	Самостоятельное объяснение учебной темы ИИ, получение рецензии и разбор допущенных неточностей.	Онлайн-сервис GigaChat , библиотека материалов ФГИС «Моя школа» (https://myschool.edu.ru).
1.4	Техника Фейнмана: объяснение концепций разным аудиториям	1	Практикум	Формирование запросов на многоуровневое объяснение сложных терминов (для новичка,	Интеллектуальный ассистент YandexGPT , образовательный ресурс «Урок цифры» (https://урокцифры.рф).

				школьника, специалиста).	
1.5	Генерация адаптивных тренажеров и квизов	1	Практикум	Генерация контрольно-диагностических тестов с нарастающей сложностью, прохождение тестирования в чате.	Платформа GigaChat , конструктор тестов Online Test Pad (https://onlinetestpad.com).
1.6	Визуализация знаний: ментальные карты (Mind Maps)	1	Практикум	Извлечение ключевых сущностей из текста параграфа и построение древовидной карты связей.	Онлайн-сервисы визуализации MindMeister (https://www.mindmeister.com), Miro или XMind .
1.7	Мнемотехники и карточки интервального повторения	1	Практикум	Создание мнемонических формул с помощью ИИ, генерация наборов карточек для запоминания.	Сервис карточек интервального повторения AnkiWeb (https://ankiweb.net), платформа GigaChat .
1.8	Защита персонального трека самообучения	1	Защита мини-кейсов	Презентация эффективных шаблонов промптов, пополнение классного репозитория промптов.	Интерактивная доска для совместной работы (SBoard/Miro), классный репозиторий на Яндекс Диск .
Раздел 2	ИИ в анализе кода и сложной информации	8			
2.1	Понятие кода. Генерация кода по техническому описанию	1	Беседа, практикум	Формулирование ТЗ, генерация кода скриптов через нейросеть, первичный запуск без правки.	Среда разработки Thonny IDE (https://thonny.org), диалоговая система GigaChat .

2.2	Построчный разбор анатомии и структуры кода	1	Практикум	Анализ листинга программы, запрос построчных комментариев от ИИ, выявление назначения блоков кода.	Онлайн-интерпретатор Python (https://replit.com/ / https://trinket.io/), диалоговая система YandexGPT .
2.3	Дебаггинг с ИИ: поиск логических и синтаксических ошибок	1	Лабораторная работа	Подача поврежденного скрипта в нейросеть, выявление синтаксических сбоев и опечаток, отладка.	Среда Thonny IDE , интерактивный курс Stepik: Поколение Python (курс для начинающих) .
2.4	Реверс-инжиниринг: перевод кода в словесный алгоритм	1	Практикум	Чтение кода на Python и составление пошагового словесного описания и блок-схемы алгоритма.	Онлайн-редактор блок-схем Draw.io (https://app.diagrams.net/), сервис GigaChat .
2.5	Пошаговое мышление: метод Chain of Thought	1	Практикум	Решение комплексных логических задач с требованием генерации пошаговой цепочки рассуждений.	Диалоговая платформа YandexGPT , банк олимпиадных задач платформы Сириус.Курсы (https://edu.sirius.online/).
2.6	Структурирование и аналитическое сравнение текстов	1	Практикум	Сравнение двух подходов/документов, сведение аналитических выводов в единую сводную таблицу.	Яндекс Таблицы (https://docs.yandex.ru/), диалоговая система GigaChat .

2.7	Практикум: Конструирование сценария игры через ИИ-промптинг	1	Практикум	Пошаговое наращивание логики текстового квеста посредством серии последовательных инструкций.	Среда Thonny IDE , диалоговая система GigaChat / YandexGPT .
2.8	Презентация и аналитический разбор сгенерированных алгоритмов	1	Семинар-дискуссия	Публичный разбор алгоритмических конструкций, выявление узких мест и путей оптимизации.	Презентационная платформа, облачное хранилище учебных работ.
Раздел 3	Основы алгоритмизации и программирования на языке Python	10			
3.1	Знакомство со средой разработки. Функция <code>print()</code>	1	Практикум	Установка среды Thonny, написание и запуск первой программы, вывод форматированного текста.	Среда Thonny IDE , учебный курс « Яндекс Учебник: Информатика 8 класс » (https://education.yandex.ru/informatics).
3.2	Переменные и базовые типы данных (<code>int</code> , <code>float</code> , <code>str</code>)	1	Практикум	Объявление переменных, выполнение арифметических операций, вычисление математических выражений.	Среда Thonny IDE , платформа Stepik: Введение в Python (https://stepik.org).
3.3	Ввод данных <code>input()</code> и явное преобразование типов	1	Практикум	Написание интерактивных консольных программ (калькулятор стоимости, расчет	Платформа Stepik , тренажер задач Питонтьютор (http://pythontutor.ru).

				параметров фигур).	
3.4	Неполное и полное ветвление: конструкция <code>if - else</code>	1	Практикум	Программирование алгоритмов с проверкой условий (проверка возраста, сравнение двух чисел).	Интерактивный учебник «Яндекс Учебник: Ветвления в Python», среда Thonny IDE .
3.5	Множественный выбор: конструкция <code>if - elif - else</code>	1	Практикум	Реализация программ-классификаторов (определение четверти координатной плоскости, шкала оценок).	Интерактивный задачник Питонтьютор , среда Thonny IDE .
3.6	Сложные логические условия (<code>and</code> , <code>or</code> , <code>not</code>)	1	Практикум	Написание программ проверки принадлежности числа диапазону, валидация входных параметров.	Платформа Stepik , задачник Информатикс (https://informatics.msk.ru).
3.7	Практикум: Разработка консольного опросника/тестера	1	Практикум	Создание программы-анкеты со счетчиком правильных ответов, подсчетом баллов и итоговым выводом.	Среда Thonny IDE , задачник Яндекс Учебник: Информатика .
3.8	Цикл с предусловием <code>while</code> . Счетчики и условия выхода	1	Практикум	Реализация циклов с контролем ввода (валидация пароля, алгоритм накопления суммы).	Интерактивный тренажер Питонтьютор: цикл while , среда Thonny IDE .

3.9	Цикл со счетчиком for и генератор диапазона range()	1	Практикум	Программирование алгоритмов перебора числовых рядов, вывод таблицы умножения.	Интерактивный тренажер Питонтьютор: цикл for , среда Thonny IDE .
3.10	Контрольный практикум: Реализация базовых алгоритмов	1	Практическая работа	Самостоятельное решение 3–4 прикладных алгоритмических задач на ветвления и циклы.	Автоматизированная тестирующая система Яндекс Контест / Stepik .
Раздел 4	Разработка алгоритмов и проектов на Python	8			
4.1	Работа со строками: индексы, срезы и функция len()	1	Практикум	Разбор и написание программ анализа текстовых строк (проверка палиндрома, цензор, подсчет символов).	Курс Stepik: Строки в Python , тренажер Питонтьютор .
4.2	Списки (list): создание, индексы и базовые операции	1	Практикум	Создание списков, обращение к элементам по индексам, динамическое заполнение списка в цикле.	Интерактивный задачник Питонтьютор: Списки , среда Thonny IDE .
4.3	Методы списков (append , remove , sort) и агрегация	1	Практикум	Модификация списков, нахождение минимального, максимального и среднего значений элементов.	Интерактивный учебник Яндекс Учебник: Списки , среда Thonny IDE .

4.4	Пользовательские функции: оператор def и директива return	1	Практикум	Декомпозиция кода программы на вспомогательные функции, передача параметров и возврат значений.	Платформа Stepik: Функции в Python , среда Thonny IDE .
4.5	Проектирование и прототипирование итогового проекта	1	Проектная работа	Выбор темы (игра «Угадай число», викторина, калькулятор), составление ТЗ и структуры проекта.	Онлайн-доска Miro / SBoard , текстовый редактор спецификаций.
4.6	Программирование, отладка и тестирование проекта	1	Проектная работа	Написание программного кода консольного приложения, тестирование крайних случаев, устранение ошибок.	Среда Thonny IDE / IDLE , диалоговая система GigaChat (для ревью кода).
4.7	Подготовка сопроводительной документации и презентации	1	Практикум	Комментирование кода по стандарту PEP 8, оформление демонстрационного слайда и паспорта проекта.	Сервис презентаций Яндекс Презентации , локальный репозиторий проектов.
4.8	Публичная защита проектов и подведение итогов года	1	Конференция / Защита проектов	Демонстрация работы программы перед аудиторией, ответы на вопросы, рефлексия освоенного материала.	Мультимедийное презентационное оборудование, онлайн-витрина проектов класса.
ИТОГО		34			

