Муниципальное общеобразовательное учреждение
«Гимназия №7 Красноармейского района Волгограда»
400026, Волгоград, б-р им. Энгельса, 33 тел.69-98-66, 69-56-77, 67-05-55 E-mail: gymnasium7@volgadmin.ru

УТВЕРЖДЕНО
на педагогическом совете
МОУ гимназии № 7
протокол № 11 от «31» августа 2026г.

Введено в действие
приказом директора
МОУ гимназии № 7
от «21» октября 2026 г. № 253ОД
Директор МОУ гимназии № 7
_____ И.Г.Салагина
«21» октября 2026 г.

РАБОЧАЯ ПРОГРАММА

Дополнительной общеинтеллектуальной программы

«IT - лаборатория»

ПЛАТНЫЕ ОБРАЗОВАТЕЛЬНЫЕ УСЛУГИ

Возраст обучающихся: 13-18 лет
Срок реализации: 6 месяцев

Разработчик: Ордынцев Виктор Игоревич,
педагог дополнительного образования

Волгоград
2026 г.

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Программа курса «IT – лаборатория» разработана для реализации в рамках системы платных образовательных услуг МОУ гимназии №7 в 2026-2027 уч. году для 7-11 классов, предоставляемых сверх установленного муниципальным заданием.

Цели курса:

- изучение основных приемов написания программ на современном языке программирования Python, развитие алгоритмического мышления обучающихся, творческих способностей, аналитических и логических компетенций.

Задачи курса:

Предметные:

- Изучить синтаксис и основные конструкции языка Python, такие как переменные, операторы, циклы, условные выражения и функции;
- Освоить принципы объектно-ориентированного программирования в Python, включая создание и использование классов и объектов;
- Научиться работать с встроенными типами данных Python (списки, кортежи, множества, словари) и использовать их для эффективного решения задач;
- Освоить работу с файлами и файловой системой: чтение и запись данных, управление файлами и директориями;
- Научиться обрабатывать исключения и ошибки в коде для создания надежных и устойчивых к сбоям приложений;
- Развить навыки работы с библиотеками и модулями Python, включая стандартные библиотеки (например, os, sys, datetime) и внешние библиотеки (например, numpy, pandas);
- Научиться использовать системы управления версиями (Git) для контроля изменений в коде и совместной работы над проектами.

Метапредметные:

- Развить навыки критического мышления и логического анализа для решения алгоритмических задач и написания эффективного кода;
- Научиться работать в команде, участвовать в групповых проектах, обмениваться кодом и опытом, используя платформы для совместной работы;

- Развить способность планировать и управлять временем при выполнении программных заданий и проектов, устанавливая приоритеты и дедлайны;
- Освоить основы проектной деятельности, включая постановку задач, выбор подходящих инструментов, разработку и тестирование программных решений;
- Научиться использовать документацию и ресурсы сообщества для поиска и анализа информации, необходимой для решения задач;
- Развить навыки презентации и демонстрации своих программных продуктов, включая написание технической документации и подготовку презентаций;
- Улучшить умение адаптироваться к изменениям и новым требованиям в быстро развивающейся сфере разработки программного обеспечения.

Личностные:

- Сформировать устойчивый интерес к изучению программирования и новым технологиям в области Python-разработки.
- Развить уверенность в своих способностях решать сложные задачи и создавать программные продукты, преодолевая трудности и ошибки.
- Воспитать ответственность и дисциплинированность в выполнении учебных и проектных задач, соблюдая сроки и качество работы.
- Сформировать навыки саморефлексии и самооценки для анализа своих достижений и определения направлений для дальнейшего развития.
- Развить творческое мышление и инициативность в поиске новых идей и подходов к разработке программных решений.
- Стимулировать личностный рост и стремление к постоянному совершенствованию своих профессиональных навыков и компетенций в области программирования.
- Поддерживать и развивать навыки эффективной коммуникации и взаимодействия с коллегами и партнерами в профессиональной среде.

Курс «IT – лаборатория» предназначен для обучающихся, интересующихся информационными технологиями и программированием, желающих освоить основы разработки программ на языке Python.

Программа курса рассчитана на 48 часов (2 ч в неделю, 24 учебные недели) и реализуется один год.

Для достижения поставленных целей наиболее целесообразными являются различные формы занятий: лекции, практикумы, тренинги.

Формой итогового контроля является защита индивидуального или группового проекта.

Предусмотрены следующие формы организации работы:

1. теоретический блок, посвященный изучение литературы, документаций, инструктажей;
2. практический блок, представленный упражнениями, соревнованиями, защитой проекта.

Будут реализованы активные методы обучения такие, как:

- метод проектов;
- практические методы (упражнения, задачи, практические работы и др.);
- словесные методы (рассказ, беседа, инструктаж, чтение справочной литературы и др.);
- наглядные методы (демонстрации мультимедийных презентаций, проектов, программ, фотографии, видеороликов и др.);
- проблемные методы (методы проблемного изложения – детям дается часть готового знания и др.);
- эвристические (частично-поисковые) – детям предоставляется большая возможность выбора вариантов;
- исследовательские – дети сами открывают и исследуют знания;
- конкретные и абстрактные методы: синтез и анализ, сравнение, обобщение, абстрагирование, классификация, систематизация, т.е.
- методы как мыслительные операции;
- индуктивные методы, дедуктивные методы.

Ключевые содержательные разделы курса рекомендуется изучать с использованием наглядных материалов, схем, алгоритмов, примеров программ и опорных конспектов, позволяющих систематизировать и закрепить знания по основам программирования на языке

Python. Практические занятия должны быть направлены на применение изученного материала при решении задач, моделировании различных ситуаций и разработке программных продуктов, связанных с практическими потребностями и интересами обучающихся.

Особое внимание необходимо уделять проектной деятельности, предусматривающей последовательное прохождение этапов постановки задачи, разработки алгоритма, написания программного кода, тестирования и представления полученного результата. Организация образовательного процесса должна учитывать возрастные особенности, индивидуальные интересы и различный уровень начальной подготовки обучающихся. Для отслеживания динамики освоения программы рекомендуется проводить входную диагностику, текущий контроль практических навыков, оценку результатов выполнения тематических заданий и мини-проектов, а также итоговую оценку разработанного проекта.

В результате прохождения курса «IT – лаборатория» у обучающихся формируются следующие умения:

Предметные:

- Знание синтаксиса и основных конструкций языка Python, такие как переменные, операторы, циклы, условные выражения и функции;
- Знание принципов объектно-ориентированного программирования в Python, включая создание и использование классов и объектов;
- Умение работать с встроенными типами данных Python (списки, кортежи, множества, словари) и использовать их для эффективного решения задач;
- Умение работать с файлами и файловой системой: чтение и запись данных, управление файлами и директориями;
- Умение обрабатывать исключения и ошибки в коде для создания надежных и устойчивых к сбоям приложений;
- Умение работать с библиотеками и модулями Python, включая стандартные библиотеки (например, os, sys, datetime) и внешние библиотеки (например, numpy, pandas);
- Умение использовать системы управления версиями (Git) для контроля изменений в коде и совместной работы над проектами.

Метапредметные:

- Развитие навыков критического мышления и логического анализа для решения алгоритмических задач и написания эффективного кода;

- Умение работать в команде, участвовать в групповых проектах, обмениваться кодом и опытом, используя платформы для совместной работы;
- Развитие способности планировать и управлять временем при выполнении программных заданий и проектов, устанавливая приоритеты и дедлайны;
- Знание основ проектной деятельности, включая постановку задач, выбор подходящих инструментов, разработку и тестирование программных решений;
- Умение использовать документацию и ресурсы сообщества для поиска и анализа информации, необходимой для решения задач;
- Развитие навыков презентации и демонстрации своих программных продуктов, включая написание технической документации и подготовку презентаций;
- Умение адаптироваться к изменениям и новым требованиям в быстро развивающейся сфере разработки программного обеспечения.

Личностные:

- Формирование устойчивого интереса к изучению программирования и новым технологиям в области Python-разработки.
- Развитие уверенности в своих способностях решать сложные задачи и создавать программные продукты, преодолевая трудности и ошибки.
- Формирование ответственности и дисциплинированности в выполнении учебных и проектных задач, соблюдая сроки и качество работы.
- Формирование навыков саморефлексии и самооценки для анализа своих достижений и определения направлений для дальнейшего развития.
- Развитие творческого мышления и инициативности в поиске новых идей и подходов к разработке программных решений.
- Стимулирование личностного роста и стремление к постоянному совершенствованию своих профессиональных навыков и компетенций в области программирования.
- Поддержание и развитие навыков эффективной коммуникации и взаимодействия с коллегами и партнерами в профессиональной среде.

Тематическое планирование курса

№ п/п	Раздел	Теория	Практика	Кол-во часов	Дата проведения	
					По плану	По факту
1	Техника безопасности и правила пользования компьютером. Знакомство с проектной деятельностью	– Правила техники безопасности и поведения в компьютерном классе – Понятие проекта, этапы проектной деятельности – Обзор курса, язык Python и среда разработки	– Входная диагностика – Установка и настройка среды программирования – Запуск первой программы – Обсуждение возможных тем проектов	1	1 неделя	
2	Математические вычисления в Python. Ввод и вывод данных	– Числовые типы int и float – Арифметические операции, //, %, ** – Функции input() и print(), приведение типов – Форматирование вывода (f-строки)	– Задачи на вычисления: площадь и периметр фигур, среднее арифметическое – Разбор цифр числа – Простой калькулятор	1	1 неделя	
3	Строки, списки, словари и кортежи в Python	– Строки: индексы, срезы, методы (split, join, find, replace) – Списки и их методы (append, pop, sort), срезы – Кортежи и неизменяемость – Словари: ключ-значение, keys, values, items – Множества	– Обработка строк: подсчёт символов, проверка палиндрома – Поиск максимума и сортировка списка – Программа «Телефонный справочник» на словаре – Наблюдение за списками и кортежами в Python Tutor	2	2 неделя	
4	Условия if-elif-else и операторы сравнения. Модуль random	– Тип bool, операторы сравнения – Логические операции and, or, not – Операторы if, if-else, if-elif-else, вложенные условия	– Задачи на ветвление: чётность числа, високосный год, существование треугольника – Игра «Угадай число»	2	3 неделя	

		– Модуль random: randint, choice, random	– Игра «Камень, ножницы, бумага»			
5	Циклы for и while. Модуль array	– Цикл for и функция range – Цикл while – Операторы break и continue, вложенные циклы – Модуль array: создание и методы массивов	– Таблица умножения, суммы и произведения – Числа Фибоначчи, поиск простых чисел – Игра с циклом while – Задачи на обработку массивов array	2	4 неделя	
6	Решение задач по изученным темам	– Повторение: типы данных, условия, циклы – Алгоритм решения задачи: анализ условия, план, код, тестирование – Приёмы отладки программ	– Самостоятельное решение задач разного уровня сложности (informatics.mccme.ru) – Мини-соревнование – Взаимопроверка кода, разбор типичных ошибок	2	5 неделя	
7	Исключения и их обработка	– Синтаксические ошибки и исключения – Конструкция try-except-else-finally – Типы исключений (ValueError, ZeroDivisionError и др.), оператор raise	– Защита программы от неверного ввода – Безопасный калькулятор – Ввод числа в цикле до получения корректного значения	1	6 неделя	
8	Функции. Рекурсия	– Определение функции (def), параметры и return – Аргументы по умолчанию, *args – Локальные и глобальные переменные – Рекурсия: базовый случай, глубина, lambda-функции	– Оформление ранее решённых задач в виде функций – Рекурсивный факториал и числа Фибоначчи – Визуализация рекурсии в Python Tutor – Задачи на функции	3	6 неделя 7 неделя	

9	Модули datetime, time и sys. Создание собственного модуля	– Подключение модулей: import, from ... import – Модули datetime, time, sys – Структура собственного модуля, конструкция if __name__ == "__main__" – Установка сторонних пакетов (pip)	– Вычисление количества дней до дня рождения – Таймер и измерение времени работы кода – Создание собственного модуля с функциями и подключение его в другой программе	2	8 неделя	
10	Работа с файлами	– Функция open, режимы r, w, a, оператор with – Чтение и запись текстовых файлов, формат CSV – Модуль os: пути, каталоги, список файлов	– Список дел с сохранением в файл – Подсчёт слов в тексте – Сохранение результатов игры – Обход каталога с помощью os	2	9 неделя	
11	Подготовка и реализация мини-проекта	– Этапы проекта: идея, требования, алгоритм, код, тестирование – Планирование работы и распределение времени – Оформление результата	– Выбор темы (текстовая игра, викторина, учёт расходов и др.) – Разработка алгоритма, написание и отладка кода – Тестирование и подготовка демонстрации	3	10 неделя 11 неделя	
12	Защита мини-проектов	– Критерии оценивания проекта – Правила публичного представления результата	– Демонстрация программ и ответы на вопросы – Взаимооценивание и рефлексия	1	11 неделя	
13	Основы объектно-ориентированного программирования. Классы, свойства и методы. Инкапсуляция и полиморфизм	– Парадигмы программирования, идея ООП – Класс и объект, атрибуты и методы, параметр self – Инкапсуляция: публичные, защищённые и приватные атрибуты, property	– Создание классов «Студент», «Банковский счёт» – Работа с объектами и методами – Приватные поля, геттеры и сеттеры – Полиморфный вызов одноимённых методов разных классов	2	12 неделя	

		– Полиморфизм				
14	Конструктор и деструктор класса. Перегрузка операторов	– Конструктор __init__ и деструктор __del__ – Методы __str__ и __repr__ – Перегрузка операторов: __add__, __eq__, __lt__, __len__	– Класс «Вектор» с перегрузкой сложения и сравнения – Класс «Дробь» – Счётчик созданных объектов	2	13 неделя	
15	Наследование	– Базовый и производный классы – Функция super(), переопределение методов – Множественное наследование (обзор) – isinstance и issubclass	– Иерархия «Животное — Собака, Кошка» – Игровые персонажи с общим базовым классом – Иерархия «Транспорт»	2	14 неделя	
16	Решение задач по изученным темам	– Повторение ООП – Проектирование классов: схема классов, композиция и наследование	– Задачи «Библиотека», «Инвентарь игрока», «Зоопарк» – Взаимопроверка кода – Мини-соревнование	2	15 неделя	
17	Библиотеки. Система контроля версий Git	– Понятие системы контроля версий – Репозиторий, commit, ветки, merge – Удалённый репозиторий, GitHub, файл .gitignore	– Установка Git, git init, add, commit, status, log – Создание ветки и слияние – Публикация проекта на GitHub – Совместная работа в парах, решение конфликта слияния	2	16 неделя	
18	Библиотеки. NumPy	– Массив ndarray, создание, индексация и срезы – Векторные операции и агрегирующие функции – Формы массивов: shape, reshape	– Статистика оценок класса – Сравнение скорости обработки list и ndarray – Операции с матрицами	2	17 неделя	

19	Библиотеки. Pandas	– Структуры Series и DataFrame – Чтение CSV-файлов – Фильтрация, сортировка, группировка, пропущенные значения	– Загрузка набора данных (оценки, погода и др.) – Фильтрация и группировка данных – Сохранение результата в файл	2	18 неделя	
20	Библиотеки. Matplotlib и визуализация данных	– Модуль pyplot, виды графиков – Подписи, оси, легенда, оформление – Построение графиков по данным из Pandas	– Построение графиков функций – Диаграммы по данным из Pandas – Мини-отчёт с несколькими графиками	2	19 неделя	
21	Постановка проблемы, задачи, утверждение темы. Определение проектного решения. Разработка дорожной карты проекта	– Проблема, цель, задачи и ожидаемый продукт проекта – Метод проектов, критерии оценки проекта – Дорожная карта: этапы, сроки, роли	– Мозговой штурм и выбор темы проекта – Формулировка цели и задач, паспорт проекта – Составление дорожной карты – Создание репозитория проекта на GitHub	2	20 неделя	
22	Реализация проекта	– Консультации по структуре программы – Тестирование и документирование (docstring, README) – Работа с Git в проекте	– Написание кода по дорожной карте – Регулярные коммиты в репозиторий – Тестирование и отладка – Промежуточный отчёт о ходе работы	4	21 неделя 22 неделя	
23	Подготовка презентации, репетиция выступления	– Структура публичного выступления – Правила оформления презентации – Критерии оценивания защиты	– Подготовка презентации и README – Репетиция выступления – Взаимные отзывы и доработка	2	23 неделя	
24	Защита проектов	– Регламент защиты проектов	– Публичная защита проектов, ответы на вопросы	2	24 неделя	

			– Рефлексия и подведение итогов курса			
	ИТОГО			48		

Литература

1. Босова, Л.Л. Информатика 8-9 классы. Начало программирования на языке Python. Дополнительные главы к учебникам / Л.Л.Босова, Н.А.Аквилянов, И.О.Кочергин и др. – М.:Бином. Лаборатория знаний, 2020. – 96 с.: ил.
2. Васильев, А.Н. Программирование на Python в примерах и задачах / А.Н. Васильев. – М.: Бомбора, 2021. – 616 с.
3. Гуриков, С.Р. Основы алгоритмизации и программирования на Python / С.Р. Гуриков. – М.: Форум, 2020. – 342 с.
4. Дауни, А. Основы Python. Научитесь думать как программист / Аллен Б. Дауни; пер. с англ. С.Черникова. – М.:Манн, Иванов и Фербер, 2021. – 304 с.
5. Доусон, М. Программируем на Python / М. Доусон. – СПб.: Питер, 2020. – 416 с.: ил.
6. Жуков, Р.А. Язык программирования Pyrhon. Практикум. Учебное пособие / Р.А. Жуков. – М.: ИНФА-М, 2019. – 216 с.
7. Реализация дополнительной общеобразовательной программы по тематическому направлению «Программирование на языке Python» с использованием оборудования центра цифрового образования детей «IT-куб». Методическое пособие. Под ред. Григорьева С. Г. – Москва, 2021.
8. Бхаргава, А. Грокаем алгоритмы: иллюстрированное пособие для программистов и любопытствующих / А. Бхаргава. — СПб.: Питер, 2019. — 288 с.
9. Лутц, М. Изучаем Python. Том 1, пер. с англ. 5-е изд. / М. Лутц. — СПб.: Символ Плюс, 2019. — 848 с.
10. Пэйн, Б. Python для детей и родителей / Б. Пэйн; пер. с анг. М.А.Райтман. – М.:Издательсов «Э», 2017. – 352 с.: ил.
11. Свейгарт, Э. Учим Python, делая крутые игры / Э. Свейгарт; пер. с анг. М.А.Райтман. – М.: Бомбора, 2021. – 416 с.

12. Стивенсон, Б. Python. Сборник упражнений / Б. Стивенсон; пер с англ. А.Ю. Гинько. – М:ДМК_Пресс, 2021. – 238 с.

13. Таке, А. Программируем с детьми. Создайте 50 крутых игр на Python

14. / А. Таке; пер. с анг. М.А.Райтман. – М.: Бомбора, 2021. – 288 с.

15. Шуманн, Х.-Г. Python для детей / Х.-Г. Шуманн; пер. с англ. М.А.Райтман. – М.:ДМК- Пресс, 2019. – 344 с.

ИНТЕРНЕТ-РЕСУРСЫ

http://www.school.edu.ru – российский общеобразовательный портал

http://www.elibrary.ru/defaultx.asp – научная электронная библиотека

http://pythontutor.com - сервис «PythonTutor», позволяющий визуализировать исполнение кода на языке Python

http://informatics.mccme.ru - всероссийский портал «Дистанционная подготовка по информатике»