

Министерство образования Самарской области  
ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ САМАРСКОЙ ОБЛАСТИ  
«ПОВОЛЖСКИЙ ГОСУДАРСТВЕННЫЙ КОЛЛЕДЖ»

Принята на заседании  
методического совета

от «31» 09.2026

Протокол № 2 от 30.09.2026

Утверждаю

Директор ГБОУ «ПК»

Шувалов А.В.

«30» 09 г.



ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ  
ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА

«Основы промышленного программирования»

*Направленность: техническая*

Возраст обучающихся: 16-19 лет

Срок реализации: 1 год

Автор-составитель: Горнеев А.М., преподаватель

Самара, 2026 г.

## **СОДЕРЖАНИЕ**

|                                         |    |
|-----------------------------------------|----|
| 1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА .....          | 3  |
| 2. УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН .....       | 7  |
| 3. СОДЕРЖАНИЕ ПРОГРАММЫ .....           | 8  |
| 4. ВОСПИТАНИЕ .....                     | 12 |
| 5. РЕСУРСНОЕ ОБЕСПЕЧЕНИЕ .....          | 17 |
| 6. СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ ..... | 18 |

## **1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

Современный рынок труда предъявляет высокие требования к владению языками программирования. Python является одним из наиболее востребованных и универсальных инструментов, используемых в веб-разработке, науке о данных, автоматизации и многих других областях. Данная программа ориентирована на систематическое изучение языка с нуля, уделяя особое внимание алгоритмическому мышлению, стилю кода (PEP 8) и практическому применению полученных знаний.

### **Направленность программы:**

Программа «Основы промышленного программирования» – техническая.

### **Новизна программы:**

Программа построена как логичная последовательность тем, охватывающих все ключевые конструкции Python, включая углублённую работу со стандартными модулями (random, datetime), обработку файлов в различных форматах (CSV, JSON) и основы объектно-ориентированного программирования. Акцент сделан на решении реальных задач и разработке собственного проекта, что позволяет учащимся почувствовать себя настоящими разработчиками.

### **Актуальность:**

Python уверенно лидирует в рейтингах языков программирования. Спрос на Python-разработчиков постоянно растёт, а знание основ языка открывает двери в такие сферы, как Data Science, машинное обучение, бэкенд-разработка и автоматизация. Программа отвечает современным образовательным стандартам и помогает учащимся сделать осознанный выбор будущей профессии.

### **Педагогическая целесообразность:**

Обучение строится на принципах постепенного усложнения, активной практики и проектной работы. Это развивает не только технические навыки,

но и критическое мышление, умение работать с документацией, навыки командной работы и самоорганизации. Регулярное решение алгоритмических задач способствует формированию структурного мышления.

**Цель программы:**

Сформировать у обучающихся базовые и продвинутое навыки программирования на Python, достаточные для создания собственных программных продуктов и продолжения обучения в IT-направлениях.

**Задачи программы:**

- освоить синтаксис и основные конструкции языка Python (переменные, типы данных, условные операторы, циклы);
- изучить встроенные коллекции (строки, списки, кортежи, множества, словари) и методы работы с ними;
- освоить написание функций, использование лямбда-выражений, декораторов;
- научиться работать с файловой системой, читать и записывать данные в форматах CSV и JSON;
- углублённо изучить модули random и datetime для генерации случайных данных и работы с датами/временем;
- освоить основы объектно-ориентированного программирования (классы, наследование, полиморфизм, перегрузка операторов);
- привить навыки оформления кода в соответствии с PEP 8, использования отладчика;
- развить проектное мышление, умение планировать и реализовывать собственные проекты;
- подготовить итоговый проект и защитить его перед аудиторией.

**Возраст обучающихся:** 16-19 лет

**Срок реализации:** 144 академических часа

**Форма обучения:** очная

**Форма организации деятельности:** индивидуальная, групповая, проектная

**Режим занятий:** занятия 2 раза в неделю по 2 академических часа

**Ожидаемые результаты:**

- уверенное владение синтаксисом Python и основными конструкциями;

- умение писать программы средней сложности с использованием коллекций, функций и классов;
- навыки работы с файлами и модулями стандартной библиотеки;
- понимание принципов ООП и умение применять их на практике;
- готовность к самостоятельному изучению дополнительных библиотек и фреймворков;
- защищённый итоговый проект

**Критерии и способы определения результативности:**

- педагогическое наблюдение
- выполнение практических заданий
- успешная защита итогового проекта

**Форма подведения итогов:** защита итогового проекта

**2. УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН**

|       | Название темы                                                                                                                      | Всего |        |          | Форма контроля                                                     |
|-------|------------------------------------------------------------------------------------------------------------------------------------|-------|--------|----------|--------------------------------------------------------------------|
|       |                                                                                                                                    | всего | теория | практика |                                                                    |
| 1.    | Введение. Установка среды. PEP 8. Ввод-вывод. Переменные                                                                           | 2     | 1      | 1        | Устный опрос, беседа, анкетирование                                |
| 2.    | Типы данных. Операции над числами. Приоритет операций                                                                              | 10    | 4      | 6        | Устный опрос, проверка практических работ, самостоятельных заданий |
| 3     | Математическая логика. Условный оператор. Сложные условия. Вложенные структуры. Отступы                                            | 12    | 6      | 6        | Устный опрос, проверка практических работ, самостоятельных заданий |
| 4     | Циклы while и for. Диапазоны. Булевы переменные. Прерывания и продолжения. Вложенные циклы                                         | 12    | 6      | 6        | Контрольные работы, тестирование, практические зачёты              |
| 5     | Строки. Индексация. Срезы. Методы строк. split() и join()                                                                          | 10    | 4      | 6        | Устный опрос, проверка практических работ, самостоятельных заданий |
| 6     | Списки. Методы. Вложенные списки. Списочные выражения                                                                              | 12    | 4      | 8        | Устный опрос, проверка практических работ, самостоятельных заданий |
| 7     | Кортежи. Множества в Python                                                                                                        | 8     | 4      | 4        | Устный опрос, проверка практических работ, самостоятельных заданий |
| 8     | Словари. Методы работы со словарями                                                                                                | 10    | 4      | 6        | Контрольные работы, тестирование, практические зачёты              |
| 9     | Функции. Области видимости. Передача параметров. *args, **kwargs. Лямбда-функции. Обработка коллекций. Поточковый ввод. Декораторы | 14    | 6      | 8        | Устный опрос, проверка практических работ, самостоятельных заданий |
| 10    | Модули random и datetime (углублённо)                                                                                              | 8     | 4      | 4        | Устный опрос, проверка практических работ, самостоятельных заданий |
| 11    | Файловая система. Работа с файлами. CSV-документы. JSON-файлы                                                                      | 10    | 4      | 6        | Устный опрос, проверка практических работ, самостоятельных заданий |
| 12    | Объектно-ориентированное программирование. Классы. Наследование. Полиморфизм. Перегрузка операторов                                | 10    | 4      | 6        | Контрольные работы, тестирование, практические зачёты              |
| 13    | Итоговый проект и защита                                                                                                           | 26    | 2      | 24       | Защита итогового проекта                                           |
| Итого |                                                                                                                                    | 144   |        |          |                                                                    |

### **3. СОДЕРЖАНИЕ ПРОГРАММЫ**

#### **Раздел 1. Введение. Установка среды. PEP 8. Ввод-вывод. Переменные (2 ч.)**

Краткая история Python, его особенности. Установка интерпретатора и выбор редактора (VS Code / PyCharm). Понятие интерпретатора, запуск скриптов. Первая программа «Hello, World!». Правила оформления кода (PEP 8): отступы, пробелы, длина строк, именование переменных. Функции print() и input(). Переменные, их объявление и динамическая типизация.

Практика: настройка рабочей среды, написание программы приветствия с запросом имени пользователя.

#### **Раздел 2. Типы данных. Операции над числами. Приоритет операций (10 ч.)**

Числовые типы: int, float. Арифметические операции (+, -, \*, /, //, %, \*\*). Приоритет операций. Логический тип bool. Преобразование типов (int(), float(), str()). Ввод чисел с клавиатуры. Понятие констант. Операции сравнения.

Практика: калькулятор для вычисления площади и периметра, расчёт среднего арифметического, программы с использованием целочисленного деления и остатка.

#### **Раздел 3. Математическая логика. Условный оператор. Сложные условия. Вложенные структуры. Отступы (12 ч.)**

Условный оператор if, elif, else. Логические операции and, or, not. Сложные составные условия. Вложенные условные конструкции. Правила оформления отступов. Тернарный оператор.

Практика: решение задач на выбор (определение времени года, проверка возраста, решение квадратного уравнения, игра «Камень-ножницы-бумага»).

#### **Раздел 4. Циклы while и for. Диапазоны. Булевы переменные. Прерывания и продолжения. Вложенные циклы (12 ч.)**

Цикл `while` с предусловием. Бесконечные циклы. Цикл `for` по последовательностям. Функция `range()`. Булевы переменные как флаги. Операторы `break`, `continue`, `else` для циклов. Вложенные циклы (таблица умножения, перебор комбинаций).

Практика: задачи на суммирование, поиск простых чисел, числовые ряды, игра «Угадай число» с ограничением попыток, построение фигур в консоли.

## **Раздел 5. Строки. Индексация. Срезы. Методы строк. `split()` и `join()` (10 ч.)**

Строковые литералы. Индексация и срезы строк. Основные методы: `upper()`, `lower()`, `strip()`, `replace()`, `find()`, `count()`, `isdigit()` и др. Форматирование строк (f-строки, метод `format()`). Методы `split()` (разбиение) и `join()` (объединение).

Практика: обработка текста (поиск слов, замена подстрок, проверка на палиндром, шифрование/дешифрование, парсинг данных из строки).

## **Раздел 6. Списки. Методы. Вложенные списки. Списочные выражения (12 ч.)**

Создание списков. Индексация и срезы. Основные методы: `append()`, `extend()`, `insert()`, `remove()`, `pop()`, `sort()`, `reverse()`, `index()`, `count()`. Вложенные списки (матрицы). Генераторы списков (list comprehensions). Функции `len()`, `min()`, `max()`, `sum()`.

Практика: работа с массивами чисел, сортировка, удаление дубликатов, создание матриц и операции над ними, фильтрация списков с помощью генераторов.

## **Раздел 7. Кортежи. Множества в Python (8 ч.)**

Кортежи: неизменяемость, создание, доступ к элементам, методы `count()`, `index()`. Множества: создание, операции объединения, пересечения, разности, симметричной разности. Методы `add()`, `remove()`, `discard()`, `union()`, `intersection()`. Замороженные множества `frozenset`.

Практика: работа с уникальными элементами, задачи на множества (поиск общих элементов, проверка подмножеств).

### **Раздел 8. Словари. Методы работы со словарями (10 ч.)**

Создание словарей, доступ к значениям по ключу. Добавление и удаление пар. Итерация по ключам, значениям, парам. Методы `keys()`, `values()`, `items()`, `get()`, `pop()`, `update()`, `setdefault()`. Вложенные словари. Генераторы словарей.

Практика: создание телефонного справочника, инвентаризация товаров, подсчёт частоты слов в тексте, преобразование данных из списка в словарь.

### **Раздел 9. Раздел 9. Функции. Области видимости. Передача параметров. \*args, kwargs. Лямбда-функции. Обработка коллекций. Поточковый ввод. Декораторы (14 ч.)**

Определение функций, аргументы и возвращаемые значения. Области видимости переменных (локальные, глобальные, `nonlocal`). Передача по значению и по ссылке. Аргументы по умолчанию, именованные аргументы. Переменное число аргументов `*args` и `**kwargs`. Анонимные функции `lambda`. Функции `map()`, `filter()`, `reduce()`. Поточковый ввод данных (чтение строк из стандартного ввода). Понятие декораторов, создание простых декораторов.

Практика: написание функций для математических операций, обработка списков с помощью `map` и `filter`.

### **Раздел 10. Модули random и datetime (углублённо) (8 ч.)**

Модуль `random`: генерация псевдослучайных чисел (`randint`, `random`, `uniform`, `choice`, `shuffle`, `sample`, `seed`). Применение для моделирования и тестирования. Модуль `datetime`: классы `date`, `time`, `datetime`, `timedelta`. Форматирование и парсинг дат (`strftime`, `strptime`). Модуль `time`: задержки (`sleep`), измерение времени (`perf_counter`).

Практика: Генератор случайных паролей (с учётом сложности). Программа расчёта возраста по дате рождения и количества дней до следующего дня рождения. Симуляция бросков кубика и анализ статистики.

Утилита для планирования задач с напоминаниями (использование `datetime` и `timedelta`). Сравнение времени выполнения различных алгоритмов.

### **Раздел 11. Файловая система. Работа с файлами. CSV-документы. JSON-файлы (10 ч.)**

Работа с текстовыми файлами: открытие, чтение, запись, закрытие (менеджер контекста `with`). Чтение и запись построчно. Обработка исключений при работе с файлами. Формат CSV: чтение и запись с использованием модуля `csv`. Формат JSON: сериализация и десериализация с помощью модуля `json`.

Практика: программа для заметок (сохранение и загрузка из файла), экспорт/импорт контактов в CSV и JSON, парсинг лог-файлов, генерация отчётов.

### **Раздел 12. Объектно-ориентированное программирование. Классы. Наследование. Полиморфизм. Перегрузка операторов (10 ч.)**

Основы ООП: классы и объекты. Конструктор `__init__`. Атрибуты и методы. Инкапсуляция (приватные атрибуты). Наследование, переопределение методов. Полиморфизм. Магические методы: `__str__`, `__repr__`, `__add__`, `__eq__` и др. Перегрузка операторов.

Практика: создание классов «Студент», «Автомобиль», «Библиотека»; реализация иерархии «Животное -> Собака/Кошка»; создание класса «Комплексное число» с перегруженными арифметическими операторами.

### **Раздел 13. Итоговый проект и защита (26 ч.)**

Самостоятельная разработка проекта. В процессе выполнения — консультации, написание документации, подготовка презентации. На последних занятиях — защита проектов с демонстрацией кода и ответами на вопросы.

#### **4. ВОСПИТАНИЕ**

Цель воспитательной работы — развитие личности через самоопределение и социализацию детей на основе социокультурных, духовно-нравственных ценностей, принятых в российском обществе, а также формирование устойчивого интереса к техническому творчеству, программированию и цифровым технологиям.

Задачи:

- формирование интереса к изучению истории и традиций российской и мировой науки, программирования, информационных технологий и цифровых медиа;
- просвещение в сфере достижений отечественной науки, техники и культуры, духовно-нравственных традиций российского общества, знакомство с выдающимися российскими учёными и программистами;
- воспитание чувства патриотизма и уважения к истории России и Самарского края через создание программных продуктов и проектов, посвящённых достижениям региона и страны;
- формирование ответственного отношения к информации, навыков критического мышления, медиаграмотности и понимания этических норм работы с данными;
- развитие коммуникативной культуры, умения работать в команде и уважать мнение других при решении коллективных алгоритмических и проектных задач;
- формирование навыков самоорганизации, дисциплины и доведения начатого дела до конца через работу над долгосрочным итоговым проектом.

Ожидаемые результаты:

- освоение детьми понятия о своей российской культурной принадлежности (идентичности) через осмысление вклада отечественных учёных и программистов в развитие мировой науки и технологий;

- понимание значимости достижений отечественной науки, техники и культуры, духовно-нравственных традиций российского общества, знание ключевых достижений российской ИТ-отрасли;
- уважение к старшим, бережное отношение к истории и традициям своей семьи, понимание важности знания истории своей страны и малой родины;
- сформированность навыков ответственного создания и распространения информации в цифровой среде, понимание этических норм работы с данными и авторского права;
- умение работать с разными аудиториями, учитывая их интересы и уровень подготовки, представлять свои проекты в доступной и наглядной форме;
- готовность к командной деятельности и взаимопомощи при создании коллективных программных проектов;
- сформированность устойчивого интереса к программированию и цифровым технологиям, готовность к дальнейшему профессиональному самоопределению в сфере ИТ.

В воспитательной работе с детьми по программе используются следующие методы воспитания:

- метод убеждения (беседы о ценности научного знания, обсуждение этических норм работы программиста, авторского права и ответственности за создаваемый код);
- метод положительного примера (демонстрация примеров выдающихся отечественных и зарубежных программистов, учёных, а также примеров педагога, родителей и сверстников);
- метод упражнений (регулярная практика написания кода, решение алгоритмических задач, работа над качеством и читаемостью программ);

- метод переключения деятельности (чередование теоретических занятий, практической работы за компьютером, обсуждений, экскурсий и творческих мастерских);
- метод развития самоконтроля и самооценки детей (рефлексия по итогам каждого проекта, самоанализ кода, взаимное рецензирование работ);
- методы воспитания воздействием группы, в коллективе (совместная работа над коллективными проектами, взаимопомощь, обмен опытом, парное программирование).

Работа осуществляется в следующих формах:

- игровые тренинги, творческие мастерские, практикумы по программированию — упражнения на развитие алгоритмического мышления, командные соревнования по решению задач, хакатоны (способствуют усвоению и применению правил поведения и коммуникации в цифровой среде, формированию позитивного и конструктивного отношения к событиям, в которых они участвуют, к членам своего коллектива, готовность к командной деятельности и взаимопомощи);
- участие в проектной деятельности — создание программных продуктов, утилит, игр и приложений, посвящённых культурному наследию России и Самарского края (например, разработка программ для обработки данных о достижениях региона, создание обучающих программ, игр на основе народных сказок и былин). Проектная деятельность способствует формированию умений в области целеполагания, планирования и рефлексии, укрепляет внутреннюю дисциплину, даёт опыт долгосрочной системной деятельности;
- встречи с профессионалами (программистами, разработчиками, инженерами, аналитиками данных, преподавателями ИТ-дисциплин) — способствуют профессиональному самоопределению и формированию уважения к труду специалистов;

- экскурсии (в ИТ-компаниях, музеи науки и техники, технопарки, центры разработки) — расширяют кругозор и формируют интерес к достижениям отечественной науки и техники;
- участие в конкурсах и олимпиадах по программированию — развивает ответственность, целеустремлённость и навыки публичной презентации результатов своей работы;
- викторины и беседы — обсуждение истории развития вычислительной техники и программирования, достижений отечественных учёных, роли цифровых технологий в современном обществе.

Работа с родителями или законными представителями осуществляется в форме:

- родительских собраний;
- открытых занятий для родителей (демонстрация творческих работ, презентация проектов);
- творческого взаимодействия в процессе подготовки творческого продукта (программные проекты, игры, утилиты, документация);
- консультаций в групповом чате объединения;
- анкетирования, опросов, собеседований;
- совместных мероприятий (выставки проектов, экскурсии, мастер-классы).

Диагностика результатов воспитательной работы осуществляется с помощью:

- педагогического наблюдения за деятельностью обучающихся на занятиях и во время мероприятий;
- оценки творческих проектов педагогом, родителями, сверстниками (защита итоговых проектов, конкурсы работ);
- отзывов, интервью, материалов рефлексии (опросы родителей, анкетирование родителей и детей, беседы с детьми, отзывы других участников мероприятий);
- анализа динамики развития алгоритмического мышления, навыков командной работы и профессионального самоопределения.

Воспитательная работа осуществляется на основной учебной базе в Поволжском государственном колледже в рамках кружкового объединения занятий (беседы, творческие проекты, викторины, игры, дискуссии о медиаэтике, информационной безопасности и авторском праве), а также на выездных площадках, в других организациях во время воспитательных мероприятий (экскурсии в музеи, посещение ИТ-компаний и центров разработки, встречи с профессиональными программистами и инженерами, студийные праздники и др.) (см. приложение «Календарный план воспитательной работы»), организуемых с помощью и при активном участии родительского сообщества.

## **5. РЕСУРСНОЕ ОБЕСПЕЧЕНИЕ ПРОГРАММЫ**

### **5.1 Учебно-методическое обеспечение**

Формы и методы проведения занятий:

- мини-лекции с демонстрацией примеров кода;
- практические работы за компьютерами;
- решение алгоритмических задач;

Учебные (дидактические) материалы:

- комплект презентаций по каждому разделу;
- методические указания к практическим работам;
- сборники задач для самостоятельного решения (с ответами и разборами);
- шаблоны проектов и чек-листы для самопроверки;

Техническое оснащение:

- компьютеры с установленными: Python 3.13+, среда разработки (PyCharm Community / VS Code);

Формы контроля:

- устный опрос (проверка теории);
- проверка выполненных практических заданий;
- промежуточные тесты по ключевым темам;
- оценка итогового проекта по критериям: функциональность, качество кода, защита.

## 5. СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Лутц М. Изучаем Python. — 5-е изд. — СПб.: О'Reilly, 2021.
2. Доусон М. Програмируем на Python. — СПб.: Питер, 2022.
3. Седжвик Р. Алгоритмы на Python. — М.: Вильямс, 2020.
4. Официальная документация Python ([docs.python.org](https://docs.python.org)) – разделы по стандартной библиотеке.
5. Горелик Н. Н. Основы программирования на Python. Учебное пособие. — М.: КноРус, 2023.
6. Онлайн-курсы Stepik, Codecademy, LeetCode, GeekBrains, Яндекс.Практикум.
7. PEP 8 – Style Guide for Python Code ([python.org/dev/peps/pep-0008](https://python.org/dev/peps/pep-0008)).

*Программа может быть скорректирована в зависимости от уровня подготовленности обучающихся и наличия технических ресурсов.*