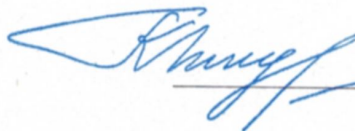


Министерство образования Владимирской области  
Государственное автономное образовательное учреждение дополнительного профессионального образования  
Владимирской области «Владимирский институт развития образования  
имени Л.И. Новиковой»

Кафедра цифрового образования и информационной безопасности



Дополнительная профессиональная программа  
(повышение квалификации)

---

**«Основы программирования на языке Python»**

Владимир  
2025

Организация — разработчик:

ГАОУ ДПО ВО «Владимирский институт развития образования имени Л.И. Новиковой»

Разработчик(и) программы:

Бурнин Д.А., методист кафедры цифрового образования и информационной безопасности  
ГАОУ ДПО ВО ВИРО.

Дубровина Н.Н., зав.кафедрой цифрового образования и информационной безопасности  
ГАОУ ДПО ВО ВИРО.

Программа **рекомендована** кафедрой цифрового образования и информационной безопасности ГАОУ ДПО ВО ВИРО к использованию в учебном процессе для повышения квалификации педагогов образовательных организаций основного общего, среднего общего и дополнительного образования, в частности, учителей информатики и ИКТ.

Протокол № 4 от «04» 12 2025г. Зав.кафедрой  
Н.Н. Дубровина /Дубровина Н.Н.

## Раздел 1. Характеристика программы

**1.1. Цель реализации программы:** повышение уровня предметной и профессиональной компетентности педагогов образовательных организаций, обучение педагогов основам языка программирования Python.

**1.2. Планируемые результаты обучения:**

Трудовая функция	Трудовое действие	Знать	Уметь
Педагогическая деятельность по проектированию и реализации образовательного процесса в образовательных организациях дошкольного, начального общего, основного общего, среднего общего образования Общепедагогическая функция. Обучение	Формирование навыков, связанных с информационно-коммуникационными технологиями (далее - ИКТ)	<ul style="list-style-type: none"> <li>– базовые концепции и синтаксис языка программирования Python;</li> <li>– управление потоком выполнения;</li> <li>– функции и модульность;</li> <li>– работа с файлами и исключения.</li> </ul>	<ul style="list-style-type: none"> <li>– Устанавливать и настраивать среду разработки (IDLE, Thonny, VS Code).</li> <li>– Создавать, сохранять и запускать скрипты.</li> <li>– Объявлять переменные, выполнять преобразования типов (int(), str(), float()).</li> <li>– Формировать сложные логические и арифметические выражения.</li> <li>– Использовать функцию print() для вывода данных и input() для ввода с клавиатуры;</li> <li>– Реализовывать разветвленную логику программы с помощью условных операторов.</li> <li>– Проектировать циклы для решения типовых задач (суммирование, поиск, перебор).</li> <li>– Контролировать выполнение циклов, прерывая их или пропуская итерации.</li> <li>– Обращивать бесконечные циклы и ошибки «off-by-one»;</li> <li>– Декомпозировать задачу на подзадачи, создавая чистые функции.</li> <li>– Писать функции с параметрами и возвращаемым значением.</li> <li>– Отлаживать функции, проверяя их на разных входных данных.</li> <li>– Использовать стандартные</li> </ul>

		<p>библиотечные функции (math.sqrt(), random.randint()).</p> <ul style="list-style-type: none"> <li>- Создавать многофайловые проекты, импортируя собственные модули.</li> <li>- Открывать файлы для чтения и записи, корректно закрывая их.</li> <li>- Обращаться к данным из текстовых файлов построчно или целиком.</li> <li>- Организовывать устойчивый ввод данных с обработкой ошибок пользователя.</li> <li>- Предвидеть и корректно обрабатывать исключительные ситуации (деление на ноль, неверный тип данных, отсутствие файла).</li> </ul>
--	--	---

**1.3. Категория слушателей:** педагоги образовательных организаций основного общего, среднего общего и дополнительного образования, в частности, учителя информатики и ИКТ, ориентированные на обучение детей программированию на языке Python, их профориентацию и решение задач, связанных с разработкой программного обеспечения.

**1.4. Форма обучения:** очная с применением ЭО и ДОТ.

**1.5. Срок освоения программы:** 48 (час)

## Раздел 2. Содержание программы

### 2.1. Учебно-тематический план

№	Наименование разделов (модулей) и тем	Всего часов	Виды учебных занятий, учебных работ			Формы контроля
			Лекция, час	Интерактивное (практическое) занятие, час	Дистанционные занятия, час	
<b>1</b>	<b>Модуль 1. Введение в Python</b>	<b>6</b>			<b>6</b>	
1.1	Знакомство с языком высокого уровня Python	1			1	
1.2	Инструменты для работы с кодом на Python	2			2	Тест
1.3	Начало работы с Python	3			3	Практическая работа
<b>2.</b>	<b>Модуль 2. Структура программы</b>	<b>3</b>			<b>3</b>	

2.1	Лексика и синтаксис Python	1		1	
2.2	Структура программы на языке Python	2		2	Тест
3.	<b>Модуль 3. Переменные, типы данных и операторы Python</b>	8		8	
3.1	Объекты в Python. Переменные	2		2	
3.2	Типы данных Python	4		4	Практическая работа
3.3	Операторы Python	2		2	Тест
4.	<b>Модуль 4. Управляющие структуры Python</b>	11		11	
4.1	Условия	4		4	
4.2	Циклы	4		4	Практическая работа
4.3	Комбинация циклов и условий	3		3	Тест
5.	<b>Модуль 5. Структурированные типы данных Python</b>	8		8	
5.1	Последовательности	2		2	
5.2	Множества	3		3	Практическая работа
5.3	Отображения	3		3	Тест
6.	<b>Модуль 6. Подпрограммы в Python</b>	6		6	
6.1	Основные понятия. Функции в Python	3		3	Практическая работа
6.2	Рекурсия	3		3	Тест
7.	<b>Модуль 7. Файлы и модули в Python</b>	6		6	
7.1	Файловый объект в Python. Работа с файлами в Python	4		4	Практическая работа
7.2	Модули и пакеты в Python и их особенности. Собственные модули в Python	2		2	Тест
	<b>Итоговая аттестация</b>	0		0	<b>Зачет</b> по совокупности выполненных практических работ и тестов
	<b>ИТОГО</b>	<b>48</b>		<b>48</b>	

## 2.2. Рабочая программа

### Модуль 1. Введение в Python.

### **1. Знакомство с языком высокого уровня Python (лекция — 1 ч.)**

*Лекция:* История языка Python, отличие Python 2 и Python 3, статистика, преимущества и недостатки Python, философия Python, решаемые с помощью Python задачи, использующие Python проекты, Python на олимпиадах и ЕГЭ.

### **2. Инструменты для работы с кодом на Python (лекция — 1 ч., практическое занятие — 1 ч.)**

*Лекция:* Онлайн платформы, требования для работы с кодом на Python на персональном компьютере, установка интерпретатора Python в Windows и GNU/Linux, установка среды разработки; требования для работы с кодом на Python на мобильном устройстве.

*Практическое занятие:* Прохождение теста модуля 1.

### **3. Начало работы с Python (лекция — 1 ч., практическое занятие — 2 ч.)**

*Лекция:* Режимы работы Python: интерактивный и программный, первые программы, получение помощи, правила оформления кода (PEP 8).

*Практическое занятие:* Выполнение практической работы модуля 1.

## **Модуль 2. Структура программы**

### **1. Лексика и синтаксис Python (лекция — 1 ч.)**

*Лекция:* Алфавит Python, физические и логические строки, структура строк и отступы, лексический анализ, лексическая структура программы на Python, синтаксическая структура.

### **2. Структура программы на языке Python (лекция — 1 ч., практическое занятие — 1 ч.)**

*Лекция:* Модульное, структурное и объектно-ориентированное программирование в Python, конструкции в Python, концептуальная иерархия.

*Практическое занятие:* Прохождение теста модуля 2.

## **Модуль 3. Переменные, типы данных и операторы Python**

### **1. Объекты в Python. Переменные (дистанционное занятие: лекция — 1 ч., практическое занятие — 1 ч.)**

*Лекция:* Объекты в Python, общие функции, переменные, инструкция присваивания, динамическая типизация, управление памятью и сборка мусора.

*Практическое занятие:* Написание программ из лекции.

### **2. Типы данных Python (лекция — 2 ч., практическое занятие — 2 ч.)**

*Лекция:* Классификация типов данных Python, проверка типов, скалярные типы, коллекции, строки и кратко о других последовательностях, кратко о множествах и отображениях, взаимное преобразование типов.

*Практическое занятие:* Выполнение практической работы модуля 3

### **3. Операторы Python (дистанционное занятие: лекция — 1 ч., практическое занятие — 1 ч.)**

*Лекция:* Арифметические операторы, операторы сравнения, логические операторы, побитовые операторы, операторы присваивания, поверхностное и глубокое копирование, операторы принадлежности, операторы тождественности, приоритет операторов, ассоциативность, изменение порядка вычисления, оператор доступа к атрибутам и методам, ввод / вывод.

*Практическое занятие:* Выполнение теста модуля 3.

#### **Модуль 4. Управляющие структуры Python**

##### **1. Условия (дистанционное занятие: лекция — 2 ч., практическое занятие — 2 ч.)**

*Лекция:* Управляющие структуры, условная инструкция, её синтаксис, отступы, варианты использования, короткая запись, инструкция pass, вложенные условия, примеры программ.

*Практическое занятие:* Написание программ из лекции.

##### **2. Циклы (дистанционное занятие: лекция — 2 ч., практическое занятие — 2 ч.)**

*Лекция:* Цикл с предусловием, его синтаксис, бесконечный цикл, короткая запись цикла while, цикл по коллекциям, его синтаксис, примеры использования, прерывание и продолжение циклов (break, continue).

*Практическое занятие:* Выполнение практической работы модуля 4.

##### **3. Комбинация циклов и условий (дистанционное занятие: лекция — 2 ч., практическое занятие — 1 ч.)**

*Лекция:* Примеры применения комбинаций циклов и условий в программе.

*Практическое занятие:* Прохождение теста модуля 4.

#### **Модуль 5. Структурированные типы данных Python**

##### **1. Последовательности (дистанционное занятие: лекция — 1 ч., практическое занятие — 1 ч.)**

*Лекция:* Список, коллекционные включения, цикл по списку, вложенные списки, методы split и join, вывод списка на экран, работа со списками, копирование списка в Python, примеры задач со списками, кортеж, числовой диапазон, байтовые строки, байтовый массив.

*Практическое занятие:* Написание программ из лекции.

##### **2. Множества (дистанционное занятие: лекция — 1 ч., практическое занятие — 2 ч.)**

*Лекция:* Множества в Python, создание множеств, операции над множествами.

*Практическое занятие:* Выполнение практической работы модуля 5.

##### **3. Отображения (дистанционное занятие: лекция — 1 ч., практическое занятие — 2 ч.)**

*Лекция:* Отображения: словари, создание словарей, операции со словарями.

*Практическое занятие:* Прохождение теста модуля 5.

#### **Модуль 6. Подпрограммы в Python**

##### **1. Основные понятия. Функции в Python (дистанционное занятие: лекция — 2 ч., практическое занятие — 1 ч.)**

*Лекция:* Подпрограммы, функции в Python, позиционные и ключевые параметры/аргументы в Python, упаковка и распаковка аргументов, передача аргументов по значению и по ссылке, область видимости, возврат нескольких значений, строки документации, анонимные функции, побочный эффект и чистые функции, время выполнения функции.

*Практическое занятие:* Выполнение практической работы модуля 6.

##### **2. Рекурсия (дистанционное занятие: лекция — 1 ч., практическое занятие — 2 ч.)**

*Лекция:* Примеры применения рекурсии в программе

*Практическое занятие:* Прохождение теста модуля 6.

## **Модуль 7. Файлы и модули в Python**

### **1. Файловый объект в Python. Работа с файлами в Python (лекция — 1 ч., практическое занятие — 3 ч.)**

*Лекция:* Файлы и целесообразность их применения, файловый объект в Python, работа с файлами в Python: основные свойства и методы, простое чтение и запись, чтение файла целиком.

*Практическое занятие:* Выполнение практической работы модуля 7.

### **2. Модули и пакеты в Python и их особенности. Собственные модули в Python (лекция — 1 ч., практическое занятие — 1 ч.)**

*Лекция:* Модули и пакеты в Python, подключение и использование, область поиска, особенности модулей в Python, собственные модули, использование модуля: запуск или импорт.

*Практическое занятие:* Прохождение теста модуля 7.

## **Раздел 3. Формы аттестации и оценочные материалы**

### **Промежуточный контроль**

**Раздел программы:** Типы данных Python

**Форма:** практическая работа по написанию небольших программ на Python.

**Описание, требования к выполнению:** практическая работа по программированию на Python, 20 заданий по написанию небольших программ с использованием знаний о переменных, типах данных и операторах.

**Критерии оценивания:** оценка «зачтено» ставится тогда, когда обучающийся успешно выполнил от 60% заданий практической работы.

**Примеры заданий:** Задание 1. В интерактивном режиме: Переменной `var_int` присвойте значение 100, `var_float` - значение 27.4, `var_str` - "La". Значение, хранимое в переменной `var_int`, увеличьте в 4.3 раза. Полученный результат свяжите с переменной `var_big`. Измените значение, хранимое в переменной `var_float`, уменьшив его на единицу, результат свяжите с той же переменной. Разделите `var_int` на `var_float`, а затем `var_big` на `var_float`. Результат данных выражений не привязывайте ни к каким переменным. Измените значение переменной `var_str` на "TuTuTuLaLaLa". При формировании нового значения используйте операции конкатенации (+) и повторения строки (\*). Выведите значения всех переменных с пояснением, один раз использовав `print()`. Сделайте подтверждающий скриншот работы в интерактивном режиме.

Задание 2. Составьте программу, которая запрашивает у пользователя 2 целых числа и выполняет операции: - арифметические: +, -, \*, /, //, %, \*\*; - сравнения: >, >=, !=, ==, выводя на экран результат каждого действия. В случае получения вещественного результата, округлите его до 2-х знаков после запятой (используя функцию `round()`). Пример входных данных: 2 3 Пример выходных данных: 5 -1 6 0.67 0 2 8 True True False False True False

Задание 3. Напишите программу, которая считывает длины двух катетов в прямоугольном треугольнике и выводит его площадь. Каждое число записано в отдельной строке. Пример входных данных: 3 5 Пример выходных данных: 7.5

**Количество попыток:** не ограничено

**Раздел программы:** Основные понятия. Функции в Python

**Форма:** практическая работа по написанию небольших программ на Python.

**Описание, требования к выполнению:** практическая работа по программированию на Python, 5 заданий по написанию небольших программ с использованием знаний о переменных, типах данных и операторах.

**Критерии оценивания:** оценка «зачтено» ставится тогда, когда обучающийся успешно выполнил от 60% заданий практической работы.

**Примеры заданий:** Задание 1. Напишите функцию `modify_list(lst)`, которая принимает на вход список целых чисел, удаляет из него все нечётные значения, а чётные нацело делит на два. Функция не должна ничего возвращать, требуется только изменение переданного списка. Функция не должна осуществлять ввод/вывод информации. После объявления функции запишите следующий код: `lst = [1, 2, 3, 4, 5, 6] print(modify_list(lst)) print(lst) modify_list(lst) print(lst) lst = [10, 5, 8, 3] modify_list(lst) print(lst)` Результат: `None [1, 2, 3] [1] [5, 4]`

Задание 2. Группа биологов в институте биоинформатики завела себе черепашку. После дрессировки черепашка научилась понимать и запоминать указания биологов следующего вида: север 10 запад 20 юг 30 восток 40 где первое слово — это направление, в котором должна двигаться черепашка, а число после слова — это положительное расстояние в сантиметрах, которое должна пройти черепашка. Но команды даются быстро, а черепашка ползёт медленно, и программисты догадались, что можно написать программу, которая определит, куда в итоге биологи приведут черепашку. Для этого программисты просят вас написать программу, которая выведет точку, в которой окажется черепашка после всех команд. Для простоты они решили считать, что движение начинается в точке (0, 0), и движение на восток увеличивает первую координату, а на север — вторую. Программе подаётся на вход число команд `n`, которые нужно выполнить черепашке, после чего `n` строк с самими командами. Вывести нужно два числа в одну строку: первую и вторую координату конечной точки черепашки. Все координаты целочисленные. Решение необходимо оформить с помощью функций! Пример входных данных: 4 север 10 запад 20 юг 30 восток 40. Пример выходных данных: 20 -20

**Количество попыток:** не ограничено.

**Раздел программы:** Инструменты для работы с кодом на Python

**Форма:** тест по введению в Python.

**Описание, требования к выполнению:** Обучающийся должен выбрать ответы на вопросы теста, тест содержит 18 вопросов разного типа.

**Критерии оценивания:** оценка «зачтено» ставится при условии, если слушатель дал не менее 70% правильных ответов в рамках теста.

**Примеры заданий:** 1) Отступы расставлены верно?

```
number = 23
guess = int(input('Введите целое число : '))
if guess == number:
    print('Поздравляю, вы угадали,') # Здесь начинается новый блок
    print('(хотя и не выиграли никакого приза!))' # Здесь заканчивается новый блок
elif guess < number:
    print('Нет, загаданное число немного больше этого.') # Ещё один блок
    # Внутри блока вы можете выполнять всё, что угодно ...
```

*else:*

```
print('Нет, загаданное число немного меньше этого.')
```

```
# чтобы попасть сюда, guess должно быть больше, чем number
```

Выберите один ответ:

Верно

Неверно

2) Python 3 несовместим с предыдущими версиями: программы, написанные на более ранних версиях языка, оказалось невозможно запускать на новой версии языка - Python 3.

Выберите один ответ:

Верно

Неверно

3) В алфавит языка программирования Python входят (выберите все верные утверждения):

Выберите один или несколько ответов:

a. 10 арабских цифр от 0 до 9.

b. Латинские прописные буквы.

c. Буквы национальных алфавитов в Unicode.

d. Римские цифры.

e. Специальные символы (знак нижнего подчёркивания; знаки препинания; круглые, квадратные скобки; знаки арифметических операций, # -знак комментариев и др.)

f. Латинские строчные буквы.

**Количество попыток: 3**

### Итоговая аттестация

Итоговая аттестация слушателей проводится после освоения всех модулей программы в форме зачета по совокупности положительных результатов выполнения практических работ и прохождения промежуточных тестов.

**Форма:** Зачет по совокупности выполненных практических работ и тестов.

**Описание, требования к выполнению:** необходимо выполнить все практические работы и пройти все тесты по материалам курса, в которых можно выбрать один или несколько вариантов ответа.

**Критерии оценивания:** оценка «зачтено» выставляется при условии, если слушатель успешно выполнил от 60% заданий практических работ и дал не менее 70% правильных ответов в рамках каждого теста.

## Раздел 4. Организационно-педагогические условия реализации программы

### 4.1. Организационно-методическое и информационное обеспечение программы

#### Нормативные документы

1. Федеральный закон от 29.12.2012 г. № 273-ФЗ «Об образовании в Российской Федерации»;
2. Федеральный закон от 27.07.2006 г. № 149-ФЗ «Об информации, информационных технологиях по защите информации»;
3. Федеральный закон от с 01.01.2008 г. № 152-ФЗ РФ «О персональных данных»;
4. СанПиН 2.4.2.2821-10 «Санитарно-эпидемиологические требования к условиям и организации обучения в общеобразовательных учреждениях».

#### Локальные акты

1. Положение об итоговой аттестации слушателей по программам повышения квалификации в ГАОУ ДПО ВО ВИРО.
2. Положение об организации дополнительного профессионального образования слушателей ГАОУ ДПО ВО ВИРО.

## Литература

1. Широбокова, С. Н. Программирование на языке Python для лабораторных занятий : учебное пособие / С. Н. Широбокова, А. А. Кацупеев, А. В. Сулыз. — Новочеркасск : ЮРГПУ, 2020. — 104 с. — ISBN 978-5-9997-0725-3.
2. Федоров, Д. Ю. Программирование на языке высокого уровня Python : учебное пособие для вузов / Д. Ю. Федоров. — 3-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2021. — 210 с. — (Высшее образование). — ISBN 978-5-534-14638-7.
3. Федоров, Д. Ю. Программирование на языке высокого уровня Python : учеб. пособие для прикладного бакалавриата / Д. Ю. Федоров. — 2-е изд., перераб. и доп. — М. : Издательство Юрайт, 2019. — 161 с. — (Серия : Бакалавр. Прикладной курс).
4. Лутц М. Изучаем Python. 5-е издание. - Пер. с англ. - Диалектика, 2019 -2020.
5. Марк Саммерфилд. Программирование на Python 3. Подробное руководство - 2009.
6. Дэвид Бизли. Python. Подробный справочник. 4-е издание. - М.: Символ-Плюс, 2010.
7. Доусон М. Программируем на Python. - СПб.: Питер, 2019.
8. Swaroop С Н, A Byte of Python. - это свободная книга по программированию на языке Python. В переводе Владимира Смоляра: <http://wombat.org.ua/AByteOfPython/>.
9. Брайсон Пэйн. Python для детей и родителей. Играй и программируй. - М.: 2017.

## Электронные обучающие материалы

### Интернет ресурсы:

1. Сайт для обучения программированию на Python, позволяющий писать и запускать программы прямо в браузере: <http://pythontutor.ru>
2. Сайт дистанционной подготовки, содержащий множество задач самого различного уровня, от простейших до международных олимпиад (для решения задач необходима регистрация): <http://informatics.mscme.ru>. Имеется автоматическая система проверки с поддержкой различных языков, в том числе Python.
3. Крупнейший в России архив задач по программированию с автоматической проверяющей системой Timus Online Judge. Основной источник задач для архива — соревнования Уральского федерального университета, Чемпионаты Урала, Уральские четвертьфиналы ICPC, Петрозаводские сборы по программированию. <https://acm.timus.ru/>
4. Сайт с материалами для изучения программирования на языке Python на основе учебника «Информатика. Углублённый уровень» для 10-11 классов К.Ю. Полякова и Е.А. Еремина. Здесь размещены варианты глав учебника по программированию, написанные на основе языка Python, презентации и отлаженные примеры всех программ, рассмотренных в учебнике. <http://kpolyakov.spb.ru/school/probook/python.htm>
5. Учебник Фоксфорда - <http://foxford.ru/wiki/informatika/o-yazyke-python>
6. Курсы Stepik <https://stepik.org/course/67>, <https://stepik.org/course/58638> и др.

## 4.2. Материально-технические условия реализации программы

### Технические средства обучения

Реализация программы требует наличия у слушателей и преподавателя персонального компьютера с выходом в Интернет и установленным пакетом офисных программ. Должна иметься возможность установки на персональный компьютер интерпретатора Python в Windows или GNU/Linux.