

Министерство науки и высшего образования Российской Федерации  
**Муромский институт (филиал)**  
федерального государственного бюджетного образовательного учреждения высшего образования  
**«Владимирский государственный университет  
имени Александра Григорьевича и Николая Григорьевича Столетовых»  
(МИ ВлГУ)**

**Отделение среднего профессионального образования**

«УТВЕРЖДАЮ»  
Заместитель директора по УР  
\_\_\_\_\_ Д.Е. Андрианов  
« 19 » 05 2026 г.

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**

**Язык программирования Python**

для специальности 09.02.11 Разработка и управление программным обеспечением

Муром, 2026 г.

Рабочая программа учебной дисциплины разработана на основе Федерального государственного образовательного стандарта (далее - ФГОС) по специальности среднего профессионального образования (далее - СПО) 09.02.11 Разработка и управление программным обеспечением №138 от 24 февраля 2025 года.

Кафедра-разработчик: информационных систем.

Рабочую программу составил: к.т.н., доцент кафедры ИС Канунова Е.Е.

от «05» мая 2026 г.

\_\_\_\_\_  
(подпись)

Рабочая программа рассмотрена и одобрена на заседании кафедры ИС.

Протокол № 21

от «05» мая 2026 г.

Заведующий кафедрой ИС *Андреанов Д.Е.*

\_\_\_\_\_  
(подпись)

## СОДЕРЖАНИЕ

1. ПАСПОРТ РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ	4
2. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ	6
3. УСЛОВИЯ РЕАЛИЗАЦИИ УЧЕБНОЙ ДИСЦИПЛИНЫ	8
4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ	10

# 1. ПАСПОРТ РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ

## Язык программирования Python

### 1.1. Область применения рабочей программы

Рабочая программа учебной дисциплины является частью программы подготовки специалистов среднего звена в соответствии с ФГОС по специальности 09.02.11 Разработка и управление программным обеспечением.

Рабочая программа учебной дисциплины может быть использована в дополнительном профессиональном образовании, для получения дополнительных компетенций, умений и знаний, необходимых для обеспечения конкурентоспособности выпускника на рынке труда и продолжения образования по специальности.

### 1.2. Место учебной дисциплины в структуре программы подготовки специалистов среднего звена

Дисциплина ОП.13 Язык программирования Python является общепрофессиональной дисциплиной

Дисциплина базируется на школьном курсе информатики и математики, и является базовой для всех специальных дисциплин направления подготовки.

### 1.3. Цели и задачи учебной дисциплины - требования к результатам освоения учебной дисциплины:

Целью дисциплины является:

- изучение и освоение методов программирования с использованием стандарта языка Python, применяемых на всех основных этапах разработки программ;
- подготовка к осознанному использованию как языков программирования, так и методов программирования.

Задачами дисциплины являются:

- формирование целостного представления о предмете;
- освоение теоретических основ структурного и объектно-ориентированного программирования;
- изучение стандарта языка программирования Python;
- использования языка программирования Python для решения практических задач

В результате освоения дисциплины обучающийся должен знать:

- Демонстрирует знание языка Python и технологий программирования, пригодных для практического применения в области информационных систем и технологий (ОК 01., ОК 03., ОК 05., ОК 07., ОК 09., ПК 2.2., ПК 2.4., ОК 02., ОК 04., ОК 06., ОК 08.).

В результате освоения дисциплины обучающийся должен уметь:

- Умеет применять технологии программирования и язык Python при решении профессиональных задач в области информационных систем и технологий (ОК 01., ОК 03., ОК 05., ОК 07., ОК 09., ПК 2.2., ПК 2.4., ОК 02., ОК 04., ОК 06., ОК 08.).

В результате освоения дисциплины обучающийся должен владеть следующими общими (ОК) и профессиональными (ПК) компетенциями:

- ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам;
- ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по правовой и финансовой грамотности в различных жизненных ситуациях;
- ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;

- ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях;
- ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках;
- ПК 2.2. Разрабатывать модули программного обеспечения;
- ПК 2.4. Выполнять тестирование и отладку программного обеспечения;
- ОК 02. Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности;
- ОК 04. Эффективно взаимодействовать и работать в коллективе и команде;
- ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных российских духовно-нравственных ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения;
- ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности;

#### **1.4. Количество часов на освоение программы учебной дисциплины:**

Максимальной учебной нагрузки обучающегося 130 часов, в том числе:  
обязательной аудиторной нагрузки обучающегося 64 часа;  
самостоятельной нагрузки обучающегося 66 часов.

## 2. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

### 2.1. Объем учебной дисциплины и виды учебной работы

Вид учебной работы	Объем часов
	3 семестр
Максимальная учебная нагрузка (всего)	130
Обязательная аудиторная учебная нагрузка (всего)	64
В том числе:	
лекционные занятия	32
практические занятия	16
лабораторные работы	16
контрольные работы	
курсовая работа / индивидуальный проект	0
Самостоятельная работа обучающегося (всего)	66
Итоговая аттестация в форме	Дифференцированный зачет

### 2.2. Тематический план и содержание учебной дисциплины

Наименование разделов и тем	Содержание учебного материала, лабораторные работы и практические занятия, самостоятельная работа обучающихся	Объем часов	Уровень освоения
1	2	3	4
	3 семестр		
<b>Раздел 1</b>	<b>Язык программирования Python</b>		
Тема 1.1 Основы языка Python	<i>Содержание учебного материала</i>		
	<i>Лекционные занятия.</i> Введение. История создания языка программирования Python. Типы данных. Операторы языка Python. Ввод и вывод данных. Математические операции в Python. Массивы. Списки.	12	1
	<i>Практические занятия.</i> Основы программирования на языке Python. Работа с массивами данных. Работа со списками и строками данных.	6	2
	<i>Самостоятельная работа обучающихся.</i> Обработка файлов разных форматов.	10	3
Тема 1.2 Практическое применение языка Python	<i>Содержание учебного материала</i>		
	<i>Лекционные занятия.</i> Библиотека Matplotlib. Функции. Фабричные методы. ООП в Python. Инкапсуляция. Наследование. ООП в Python. Полиморфизм. Библиотека SciPy. Pandas. Словари. Множества.	20	1
	<i>Практические занятия.</i> Визуализация данных. Функции. Интерполяция данных. Обработка данных. Визуализация таблиц.	10	2
	<i>Лабораторные работы.</i> Классы в Python. Наследование в Python. Полиморфизм в Python. Обработка файлов в Python.	16	3
	<i>Самостоятельная работа обучающихся.</i> Системы контроля версий (Git). Работа с базами данных в Python. Среды разработки. ИИ и Python. Работа в Cloud.	56	3
Всего:		130	



### 3. УСЛОВИЯ РЕАЛИЗАЦИИ УЧЕБНОЙ ДИСЦИПЛИНЫ

#### 3.1. Требования к минимальному материально – техническому обеспечению

Лекционная аудитория

Экран настенный Goldview; проектор Acer X128H DLP Projector; персональный компьютер - 1 шт.; колонки Genius -1 шт. Доступ к сети Интернет.

Лаборатория разработки информационных систем

12 персональных компьютеров; проектор View Sonic PG603X DLP; экран настенный Lumien

Программное обеспечение:

LibreOffice (Mozilla Public License v2.0)

7-Zip (GNU LGPL)

Mozilla Firefox (MPL)

Pycharm Community Edition (проприетарная лицензия и Apache License 2.0)

Foxit Reader (Foxit EULA)

QT Creator ((L)GPL)

Python 3 (PSF License Agreement)

OpenCV (Open Source)

NumPy (Модифицированная лицензия BSD)

SciPy (BSD)

Matplotlib (matplotlib licence)

Pandas (BSD)

Double Commander (GNU GPL 2+)

#### 3.2. Информационное обеспечение обучения

Перечень рекомендуемых учебных изданий, интернет – ресурсов, дополнительной литературы.

Основные источники:

1. Файфель, Б. Л. Основы языка Python : учебное пособие / Б. Л. Файфель, А. Е. Колобова. — Саратов : Саратовский государственный технический университет имени Ю.А. Гагарина, ЭБС АСВ, 2024. — 284 с. <https://www.iprbookshop.ru/147706.html>
2. Сузи, Р. А. Язык программирования Python : учебное пособие / Р. А. Сузи. — 4-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2024. — 350 с. . <https://www.iprbookshop.ru/142310.html>
3. Лужков, А. А. Основы работы в Python : учебно-методическое пособие / А. А. Лужков, А. С. Тюканов. — Санкт-Петербург : Издательство РГПУ им. А. И. Герцена, 2022. — 76 с. <https://www.iprbookshop.ru/131742.html>
4. Титов, А. Н. Python. Обработка данных : учебно-методическое пособие / А. Н. Титов, Р. Ф. Тагиева. — Казань : Издательство КНИТУ, 2022. — 104 с.. <https://www.iprbookshop.ru/129220.html>

Дополнительные источники:

1. Непретимова, Е. В. Математика на Python : практикум / Е. В. Непретимова, Н. Н. Масловский. — Ставрополь : Северо-Кавказский федеральный университет, 2023. — 190 с. . <https://www.iprbookshop.ru/149327.html>
2. Гаско, Р. Простой Python просто с нуля / Р. Гаско. — Москва : СОЛОН-ПРЕСС, 2023. — 256 с.. <https://www.iprbookshop.ru/142050.html>
3. Титов, А. Н. Python. Обработка данных : учебно-методическое пособие / А. Н. Титов, Р. Ф. Тагиева. — Казань : Издательство КНИТУ, 2022. — 104 с.. <https://www.iprbookshop.ru/129220.html>

4. Гегечкори, Е. Т. Программирование на языке Python : учебное пособие / Е. Т. Гегечкори. — Омск : Омский государственный технический университет, 2023. — 172 с. .  
<https://www.iprbookshop.ru/140856.html>

Интернет-ресурсы:

1. Электронная библиотечная система - [iprbookshop.ru](http://iprbookshop.ru)
2. Электронная библиотека ВлГУ - [e.lib.vlsu.ru](http://e.lib.vlsu.ru)

#### 4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

<b>Результаты обучения (освоенные умения, усвоенные знания)</b>	<b>Формы и методы контроля и оценки результатов обучения</b>
Умеет применять технологии программирования и язык Python при решении профессиональных задач в области информационных систем и технологий	контрольная работа, устный опрос
Демонстрирует знание языка Python и технологий программирования, пригодных для практического применения в области информационных систем и технологий	контрольная работа, устный опрос

**Фонд оценочных материалов (средств) по дисциплине**  
**Язык программирования Python**

## **1. Оценочные материалы для проведения текущего контроля успеваемости по дисциплине**

Список вопросов:

1. Как создать массив целых чисел типа `'i'` с помощью модуля `'array'`?
2. Чем отличается массив из модуля `'array'` от списка (`'list'`)?
3. Какие ограничения на тип данных накладывает использование массива из модуля `'array'`?
4. Как добавить элемент в конец массива и как вставить элемент по индексу?
5. Как удалить элемент из массива по значению и по индексу?
6. Какими способами можно объявить строку в Python (кавычки, тройные кавычки)?
7. Как выполнить доступ к отдельному символу строки по индексу и срезу?
8. Чем отличается метод `'replace()'` от передачи строки через `'%'` или `'format()'`?
9. Какие методы строк позволяют удалить пробелы в начале и в конце?
10. Как проверить, является ли строка числом (целым или с плавающей точкой)?
11. В чем разница между методами `'join()'` и простой конкатенацией строк?
12. Как разбить строку на список подстрок по разделителю?
13. Что такое f-строки и какие преимущества они дают перед другими способами форматирования?
14. Какими способами можно создать список (литерал, конструктор `'list()'`, генератор)?
15. Как объединить два списка (оператор `'+'` и метод `'extend()'`)?
16. В чем разница между методами `'append()'` и `'insert()'`?
17. Как получить последний элемент списка, не зная его длину?
18. Как создать поверхностную копию списка и как — глубокую?
19. Что делает метод `'pop()'` и чем он отличается от `'remove()'`?
20. Как отсортировать список с сохранением оригинала и без сохранения?
21. Как работает распаковка списка в переменные (`'a, b, *rest = my_list'`)?
22. Как создать пустое множество и почему нельзя использовать `'{}'`?
23. Какие типы данных могут быть элементами множества и почему не все?
24. Как добавить элемент во множество и как удалить его без ошибки, если элемент отсутствует?
25. Для каких операций множества (объединение, пересечение, разность, симметрическая разность) используются операторы `'|'`, `'&'`, `'-'`, `'^'`?
26. Как проверить, является ли одно множество подмножеством другого?
27. Как с помощью множеств эффективно удалить дубликаты из списка?
28. В чем отличие метода `'discard()'` от `'remove()'`?
29. Какими способами можно создать словарь (литерал, `'dict()'`, `'fromkeys()'`)?
30. Какие ограничения на ключи существуют в словаре?
31. Как безопасно получить значение по ключу, если ключа может не быть (без возникновения ошибки)?
32. В чем разница между методами `'get()'` и обращением по ключу через квадратные скобки?
33. Как перебрать пары ключ-значение в словаре в цикле `'for'`?
34. Что такое словарное включение (dictionary comprehension) и как его записать?
35. Как объединить два словаря (оператор `'|'` и метод `'update()'`)?
36. Как удалить элемент из словаря, получив при этом его значение?
37. Какие основные структуры данных предоставляет библиотека Pandas (Series, DataFrame)?
38. Как создать DataFrame из словаря, списка списков и CSV-файла?
39. Как просмотреть первые 5 строк и последние 5 строк DataFrame?

40. Как выбрать из DataFrame один столбец как Series и как подмножество строк по условию (булева маскировка)?
41. Какие методы используются для обработки пропущенных значений (`'isna'`, `'dropna'`, `'fillna'`)?
42. Как применить функцию к каждому элементу столбца или строки (метод `'apply'`)?
43. Как выполнить группировку данных с агрегацией (сумма, среднее, количество) с помощью `'groupby'`?
44. Как объединить два DataFrame по общему столбцу (аналог SQL JOIN — `'merge'`)?
45. Какие основные модули внутри Matplotlib используются для построения графиков (например, `'pyplot'`)?
46. Как построить простой линейный график (`'plot'`) и точечную диаграмму (`'scatter'`)?
47. Как создать фигуру с несколькими подграфиками (субплотами) с помощью `'subplot()'` или `'subplots()'`?
48. Какие методы позволяют добавить заголовок графика, подписи осей и легенду?
49. Как сохранить построенный график в файл (например, PNG или PDF)?
50. Как настроить размер фигуры (дюймы) и разрешение (dpi)?
51. Как построить гистограмму распределения (`'hist'`) и столбчатую диаграмму (`'bar'`)?
52. Как изменить стиль линий, цвет и маркеры на графике?
53. Какие подмодули SciPy отвечают за линейную алгебру (`'linalg'`), оптимизацию (`'optimize'`) и интегралы (`'integrate'`)?
54. Как с помощью `'scipy.linalg'` решить систему линейных уравнений?
55. Как найти корень (или минимум) скалярной функции с помощью `'scipy.optimize.minimize_scalar'`?
56. Как вычислить определенный интеграл функции с помощью `'scipy.integrate.quad'`?
57. Чем отличается `'scipy.interpolate.interpld'` от `'numpy.interp'` (при условии, что вы упоминаете NumPy как зависимость SciPy)?
58. Как выполнить одномерную интерполяцию данных и получить сглаживающий сплайн?
59. Как сгенерировать случайные числа из заданного распределения вероятностей с помощью `'scipy.stats'`?
60. Какие возможности `'scipy.spatial'` позволяют вычислить расстояние между точками или построить выпуклую оболочку?

### Общее распределение баллов текущего контроля по видам учебных работ для студентов

Рейтинг-контроль 1	Контрольная работа, практические работы, лабораторные работы	20
Рейтинг-контроль 2	Контрольная работа, практические работы, лабораторные работы	20
Рейтинг-контроль 3	Контрольная работа, практические работы, лабораторные работы	20
Посещение занятий студентом		10
Дополнительные баллы (бонусы)		10
Выполнение семестрового плана самостоятельной работы		20

**2. Промежуточная аттестация по дисциплине**  
**Перечень вопросов к экзамену / зачету / зачету с оценкой.**  
**Перечень практических задач / заданий к экзамену / зачету / зачету с оценкой (при наличии)**

Темы для устного опроса.

**Методические материалы, характеризующих процедуры оценивания**

При проведении промежуточных аттестаций используются вопросы, приведенные в пункте "Оценочные средства для промежуточной аттестации". Из каждого раздела, освоенного студентом, выбирается по два теоретических и одному практическому вопросу. Теоретические вопросы раскрываются в устной, либо в письменной форме. Практические задания как правило реализуются с помощью персонального компьютера.

При проверке знаний, приобретенных в рамках выполнения практических и лабораторных работ, используются контрольные вопросы, приведенные в методических указаниях к практическим работам. Защита практических и лабораторных работы также является средством промежуточной аттестации.

Максимальная сумма баллов, набираемая студентом по дисциплине равна 100.

Оценка в баллах	Оценка по шкале	Обоснование	<i>Уровень сформированности компетенций</i>
Более 80	«Отлично»	Содержание курса освоено полностью, без пробелов, необходимые практические навыки работы с освоенным материалом сформированы, все предусмотренные программой обучения учебные задания выполнены, качество их выполнения оценено числом баллов, близким к максимальному	<i>Высокий уровень</i>
66-80	«Хорошо»	Содержание курса освоено полностью, без пробелов, некоторые практические навыки работы с освоенным материалом сформированы недостаточно, все предусмотренные программой обучения учебные задания выполнены, качество выполнения ни одного из них не оценено минимальным числом баллов, некоторые виды заданий выполнены с ошибками	<i>Продвинутый уровень</i>

50-65	«Удовлетворительно»	Содержание курса освоено частично, но пробелы не носят существенного характера, необходимые практические навыки работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий, возможно, содержат ошибки	<i><b>Пороговый уровень</b></i>
Менее 50	«Неудовлетворительно»	Содержание курса не освоено, необходимые практические навыки работы не сформированы, выполненные учебные задания содержат грубые ошибки	<i><b>Компетенции не сформированы</b></i>

### **3. Задания в тестовой форме по дисциплине**

Примеры заданий:

1. Расположите встроенные типы данных в порядке невозрастания их размера в памяти (типичная реализация CPython 64-bit)

- int
- float
- bool
- str (пустая строка)

2. Расположите коллекции в порядке неубывания типичного потребления памяти на один элемент (без учёта служебных накладных расходов)

- list
- tuple
- set
- dict

3. Укажите назначение следующих методов строк

- strip()
- find()
- split()
- join()

4. Укажите размеры (в байтах) следующих объектов в CPython 64-bit

- bool
- int (малое целое)
- float
- пустой list

5. Укажите тип приведённых литералов и выражений

- 0xFF
- 3.14
- True
- 'a'
- "True"

6. Укажите тип приведённых литералов и конструкций

- 0b1010

- 1e-3
- False
- b'hello'
- None

7. В каких вариантах допущены ошибки?

- `s = "Hello"; s[0] = "h"`
- `s = list("Hello"); s[0] = "h"`
- `a = [1, 2, 3]; a[3] = 4`
- `d = {}; d[1] = "one"`
- `t = (1, 2, 3); t[0] = 99`

8. В каких случаях условие `if s1 == s2:` будет истинным?

- Строки `s1` и `s2` содержат одинаковые символы
- Строки `s1` и `s2` являются одним и тем же объектом в памяти
- Строки `s1` и `s2` имеют одинаковую длину
- Строки `s1` и `s2` равны по значению (содержимому)
- Такая запись для строк недопустима

9. В каких случаях доступ к элементам последовательности осуществляется неправильно?

- `a = [1, 2, 3]; a[3]`
- `a = [1, 2, 3]; a[-1]`
- `a = (1, 2, 3); a[2]`
- `a = "abc"; a[5]`
- `a = [1, 2, 3]; a[1.5]`

10. В каких случаях инициализация (создание) коллекции осуществляется без синтаксических ошибок?

- `a = [1, 2, 3,]`
- `a = (1, 2, 3)`
- `a = {1, 2, 3}`
- `a = {}` (пустое множество)
- `a = dict([(1, "one"), (2, "two")])`

11. В каких случаях доступ к элементу словаря выполнен корректно?

- `d = {1: "a", 2: "b"}; d[1]`
- `d = {1: "a", 2: "b"}; d[3]`
- `d = {"1": "a", "2": "b"}; d["1"]`
- `d = {[1,2]: "a"}; d[[1,2]]`
- `d = {(1,2): "a"}; d[(1,2)]`

12. Какие из следующих операций со множествами допустимы?

- `{1, 2, 3} | {3, 4, 5}`
- `{1, 2, 3} & {3, 4, 5}`
- `{1, 2, 3} + {4, 5, 6}`
- `{1, 2, 3} - {2, 3, 4}`
- `{1, 2, 3} * 2`

13. Укажите, какие из приведённых выражений в Pandas вернут новый DataFrame без изменения исходного

- `df.dropna()`
- `df.fillna(0)`
- `df.sort_values('col')`

- `df.iloc[:, 1]`
- `df['col'] = df['col'] + 1`

14. Какие методы Matplotlib используются для следующих действий?

- `plot()`
- `scatter()`
- `hist()`
- `imshow()`
- `bar()`

15. В каких случаях результат использования `scipy.optimize.minimize()` может быть неудовлетворительным?

- Функция не является выпуклой
- Не задан начальный вектор параметров
- Целевая функция имеет несколько локальных минимумов
- Целевая функция является константой
- Максимальное число итераций слишком мало

Полный перечень тестовых заданий с указанием правильных ответов, размещен в банке вопросов на информационно-образовательном портале института по ссылке <https://www.mivlgu.ru/iop/course/view.php?id=4577>

Оценка рассчитывается как процент правильно выполненных тестовых заданий из их общего числа.