МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ГЕОДЕЗИИ И КАРТОГРАФИИ
(МИИГАиК)

Картографический факультет
Кафедра информатики и геоинформационных технологий

**ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ**

Сертификат: 5BFEE1D286E9C95A0C0D2C611E434BD5
Владелец: Степанченко Алексей Леонидович
Действителен: с 23.09.2026 по 17.12.2027

«УТВЕРЖДАЮ»
Проректор
А. Л. Степанченко
15 мая 2026 г.

# РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ) «Б1.О.34 ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ PYTHON»

Уровень высшего образования: бакалавриат

Направление подготовки: 05.03.06 Экология и природопользование

Направленность(профиль) подготовки: Экология и природопользование

Квалификация (степень) выпускника: бакалавр

Форма обучения: очная

Год набора (приема на обучение): 2026

Срок получения образования: 4 года

Объем: в зачетных единицах: 2 з.е.
в академических часах: 72 ак.ч.

Москва, 2026

**Разработчики:**
Старший преподаватель кафедры Информатики и геоинформационных технологий Стрельцов А. С.

Рабочая программа дисциплины (модуля) составлена в соответствии с требованиями ФГОС ВО по направлению подготовки 05.03.06 Экология и природопользование, утвержденного приказом Минобрнауки от 07.08.2020 № 894

Согласование и утверждение

| № | Подразделение | Ответственное лицо | ФИО | Виза | Дата, протокол (при наличии) |
|---|---|---|---|---|---|
| 1 | Картографический факультет | Председатель ученого совета факультета | Загребин Г. И. | Рассмотрено | 26.12.2025, № 6/25 |

Содержание

1. Цель и задачи освоения дисциплины (модуля)

2. Планируемые результаты обучения по дисциплине (модулю), соотнесенные с планируемыми результатами освоения образовательной программы

3. Место дисциплины в структуре ОП

4. Объем дисциплины и виды учебной работы

5. Содержание дисциплины
   5.1. Разделы, темы дисциплины и виды занятий
   5.2. Содержание разделов, тем дисциплины и формы текущего контроля

6. Оценочные материалы текущего контроля

7. Оценочные материалы промежуточной аттестации

8. Порядок проведения промежуточной аттестации

9. Материально-техническое и учебно-методическое обеспечение дисциплины
   9.1. Перечень основной и дополнительной учебной литературы
   9.2. Профессиональные базы данных и ресурсы «Интернет», к которым обеспечивается доступ обучающихся
   9.3. Программное обеспечение и информационно-справочные системы, используемые при осуществлении образовательного процесса по дисциплине
   9.4. Оборудование и технические средства обучения

10. Методические указания по освоению дисциплины (модуля)

## 1. Цель и задачи освоения дисциплины (модуля)

Цель освоения дисциплины - формирование у обучающихся базовых представлений о языке программирования Python и практических навыков разработки простых программ для обработки данных, выполнения несложных расчётов и автоматизации типовых задач.

Задачи изучения дисциплины:

- Ознакомление студентов с основами языка Python;
- Изучение базовых алгоритмов и простейших структур данных;
- Получение практики применения базовых возможностей и библиотек языка для решения прикладных задач.

## 2. Планируемые результаты обучения по дисциплине (модулю), соотнесенные с планируемыми результатами освоения образовательной программы

*Компетенции, индикаторы и результаты обучения*

ОПК-5 Способен понимать принципы работы информационных технологий и решать стандартные задачи профессиональной деятельности в области экологии, природопользования и охраны природы с использованием информационно - коммуникационных, в том числе геоинформационных технологий

ОПК-5.1 Создает пространственные базы данных и работает со специализированным ПО для обработки и хранения экологической информации.

*Знать:*

ОПК-5.1/Зн2 Методы и приемы разработки алгоритмов и компьютерных программ

ОПК-5.1/Зн3 Методы работы с простарнственными базами данных, актуальное специализированное ПО для обработки и хранинния экологической информации.

*Уметь:*

ОПК-5.1/Ум2 Выбирать алгоритмы и приемы разработки компьютерных программ, соответствующие практическим задачам

ОПК-5.1/Ум3 Применять специализированное ПО для обработки и хранения экологической информации.

*Владеть:*

ОПК-5.1/Нв2 Приемами обеспечения корректности разработанных алгоритмов и программ

ОПК-5.1/Нв3 Навыки создания баз данных, безопасного хранения и обработки экологической информации.

## 3. Место дисциплины в структуре ОП

Дисциплина (модуль) Б1.О.34 «Программирование на языке Python» относится к обязательной части образовательной программы и изучается в семестре(ах): 3.

В процессе изучения дисциплины студент готовится к решению типов задач профессиональной деятельности, предусмотренных ФГОС ВО и образовательной программой.

## 4. Объем дисциплины (модуля) и виды учебной работы

| Период | /доемкость сы) | /доемкость ЗТ) | ые занятия сы) | кие занятия сы) | ьная работа сы) | ая аттестация сы) |
|---|---|---|---|---|---|---|

| обучения | Общая тру (ча | Общая тру (ЗЕ | Лекционни (ча | Практическ (ча | Самостоятел (ча | Промежуточн (ча |
|---|---|---|---|---|---|---|
| 3 семестр | 72 | 2 | 18 | 36 | 18 | Зачет с оценкой |
| Всего | 72 | 2 | 18 | 36 | 18 | |

## 5. Содержание дисциплины (модуля)

### 5.1. Разделы, темы дисциплины и виды занятий

(часы промежуточной аттестации не указываются)

| Наименование раздела, темы | Всего | Лекционные занятия | Практические занятия | Самостоятельная работа | Планируемые результаты обучения, соотнесенные с результатами освоения программы |
|---|---|---|---|---|---|
| **Раздел 1. Основы программирования и среда разработки** | **5** | **1** | **3** | **1** | ОПК-5.1 |
| Тема 1.1. Введение в язык программирования Python | 1 | | 1 | | |
| Тема 1.2. Интегрированная среда разработки | 1 | | 1 | | |
| Тема 1.3. Базовые элементы языка | 3 | 1 | 1 | 1 | |
| **Раздел 2. Линейные алгоритмы и базовые операции** | **5** | **1** | **3** | **1** | ОПК-5.1 |
| Тема 2.1. Функции ввода-вывода | 1 | | 1 | | |
| Тема 2.2. Арифметические операции и математические функции | 1 | | 1 | | |
| Тема 2.3. Генерация случайных чисел | 3 | 1 | 1 | 1 | |
| **Раздел 3. Управляющие конструкции** | **7** | **2** | **3** | **2** | ОПК-5.1 |
| Тема 3.1. Условные операторы | 1 | | 1 | | |
| Тема 3.2. Циклические конструкции | 1 | | 1 | | |
| Тема 3.3. Вложенные циклы и сложные алгоритмы | 5 | 2 | 1 | 2 | |
| **Раздел 4. Обработка строк и текстовых данных** | **7** | **2** | **3** | **2** | ОПК-5.1 |
| Тема 4.1. Строковый тип данных | 1 | | 1 | | |
| Тема 4.2. Операции со строками | 1 | | 1 | | |

| | | | | | |
|---|---|---|---|---|---|
| Тема 4.3. Форматирование и регулярные выражения | 5 | 2 | 1 | 2 | |
| **Раздел 5. Структуры данных** | **7** | **2** | **3** | **2** | ОПК-5.1 |
| Тема 5.1. Списки и операции над ними | 1 | | 1 | | |
| Тема 5.2. Кортежи и множества | 1 | | 1 | | |
| Тема 5.3. Словари и их применение | 5 | 2 | 1 | 2 | |
| **Раздел 6. Функциональное программирование** | **7** | **2** | **3** | **2** | ОПК-5.1 |
| Тема 6.1. Определение и вызов функций | 1 | | 1 | | |
| Тема 6.2. Расширенные возможности функций | 1 | | 1 | | |
| Тема 6.3. Функции высшего порядка и декораторы | 5 | 2 | 1 | 2 | |
| **Раздел 7. Модульная организация программ** | **9** | **2** | **5** | **2** | ОПК-5.1 |
| Тема 7.1. Работа с модулями | 2 | | 2 | | |
| Тема 7.2. Пакеты и их структура | 2 | | 2 | | |
| Тема 7.3. Стандартная библиотека | 5 | 2 | 1 | 2 | |
| **Раздел 8. Объектно-ориентированное программирование** | **9** | **2** | **5** | **2** | ОПК-5.1 |
| Тема 8.1. Основы ООП | 2 | | 2 | | |
| Тема 8.2. Наследование и полиморфизм | 1 | | 1 | | |
| Тема 8.3. Специальные методы и инкапсуляция | 6 | 2 | 2 | 2 | |
| **Раздел 9. Итераторы и функциональные возможности** | **7** | **2** | **3** | **2** | ОПК-5.1 |
| Тема 9.1. Протокол итерации | 1 | | 1 | | |
| Тема 9.2. Контейнерные типы | 1 | | 1 | | |
| Тема 9.3. Перечисления и именованные константы | 5 | 2 | 1 | 2 | |
| **Раздел 10. Работа с файловой системой** | **9** | **2** | **5** | **2** | ОПК-5.1 |
| Тема 10.1. Файловый ввод-вывод | 1 | | 1 | | |
| Тема 10.2. Операции с файлами и каталогами | 3 | | 3 | | |
| Тема 10.3. Сериализация данных | 5 | 2 | 1 | 2 | |
| **Итого** | **72** | **18** | **36** | **18** | |

### 5.2. Содержание разделов, тем дисциплин и формы текущего контроля

***Раздел 1. Основы программирования и среда разработки***

*Тема 1.1. Введение в язык программирования Python*

История развития языка Python. Области применения. Сравнительный анализ с другими языками программирования. Версии Python и их особенности.

*Тема 1.2. Интегрированная среда разработки*

Установка и настройка Python. Знакомство со средой IDLE. Альтернативные IDE: PyCharm, VS Code, Jupyter Notebook. Настройка рабочего окружения.

*Тема 1.3. Базовые элементы языка*

Структура программы. Правила именования идентификаторов. Комментарии и документирование кода. Основные типы данных: числовые, логические, строковые. Динамическая типизация. Преобразование типов.

***Раздел 2. Линейные алгоритмы и базовые операции***

*Тема 2.1. Функции ввода-вывода*

Функция input() и ее особенности. Функция print() и форматированный вывод. Управляющие последовательности. Работа с потоками ввода-вывода.

*Тема 2.2. Арифметические операции и математические функции*

Оператор присваивания. Арифметические операторы. Приоритет операций. Встроенные математические функции. Модуль math и его возможности.

*Тема 2.3. Генерация случайных чисел*

Модуль random. Генерация случайных целых и вещественных чисел. Случайный выбор из последовательности. Инициализация генератора случайных чисел.

***Раздел 3. Управляющие конструкции***

*Тема 3.1. Условные операторы*

Операторы сравнения и логические операторы. Конструкция if-elif-else. Тернарный условный оператор. Вложенные условия.

*Тема 3.2. Циклические конструкции*

Цикл while и его особенности. Цикл for и итерирование. Функции range() и enumerate(). Операторы break, continue, else в циклах.

*Тема 3.3. Вложенные циклы и сложные алгоритмы*

Организация вложенных циклов. Оптимизация циклических конструкций. Примеры реализации классических алгоритмов.

***Раздел 4. Обработка строк и текстовых данных***

*Тема 4.1. Строковый тип данных*

Создание и инициализация строк. Специальные символы и экранирование. Строки Unicode. Неизменяемость строк.

*Тема 4.2. Операции со строками*

Конкатенация и репликация строк. Индексация и срезы. Методы строк: поиск, замена, разделение, объединение. Проверка свойств строк.

*Тема 4.3. Форматирование и регулярные выражения*

Методы форматирования: %, format(), f-строки. Модуль re. Синтаксис регулярных выражений. Поиск и замена по шаблону.

***Раздел 5. Структуры данных***

*Тема 5.1. Списки и операции над ними*

Создание и инициализация списков. Методы списков. Многомерные списки. Генераторы списков и списковые включения. Сортировка и поиск элементов.

*Тема 5.2. Кортежи и множества*

Особенности кортежей. Распаковка кортежей. Создание множеств. Операции над множествами: объединение, пересечение, разность.

*Тема 5.3. Словари и их применение*

Создание словарей. Методы словарей. Итерирование по словарям. Генераторы словарей. Вложенные словари. OrderedDict, defaultdict, Counter.

***Раздел 6. Функциональное программирование***

*Тема 6.1. Определение и вызов функций*

Синтаксис определения функций. Параметры и аргументы. Позиционные и именованные аргументы. Значения параметров по умолчанию.

*Тема 6.2. Расширенные возможности функций*

Функции с переменным числом параметров (*args, **kwargs). Область видимости переменных. Замыкания. Рекурсия. Вложенные функции.

*Тема 6.3. Функции высшего порядка и декораторы*

Lambda-функции. Функции map(), filter(), reduce(). Функции-генераторы и yield. Декораторы функций. Параметризованные декораторы.

***Раздел 7. Модульная организация программ***

*Тема 7.1. Работа с модулями*

Концепция модульности. Создание модулей. Инструкции import и from. Пространства имен. Повторная загрузка модулей.

*Тема 7.2. Пакеты и их структура*

Создание пакетов. Файл __init__.py. Абсолютный и относительный импорт. Пути поиска модулей. Переменная sys.path.

*Тема 7.3. Стандартная библиотека*

Обзор основных модулей стандартной библиотеки. Работа с датой и временем. Математические и статистические модули. Системные модули.

***Раздел 8. Объектно-ориентированное программирование***

*Тема 8.1. Основы ООП*

Концепция классов и объектов. Определение классов. Атрибуты и методы. Конструктор init и деструктор del. Атрибуты класса и экземпляра.

*Тема 8.2. Наследование и полиморфизм*

Простое и множественное наследование. Переопределение методов. Функции super() и isinstance(). Абстрактные классы и методы. Миксины.

*Тема 8.3. Специальные методы и инкапсуляция*

Магические методы. Перегрузка операторов. Свойства (property). Дескрипторы. Приватные и защищенные атрибуты. Декораторы классов.

***Раздел 9. Итераторы и функциональные возможности***

*Тема 9.1. Протокол итерации*

Итерируемые объекты и итераторы. Методы iter и next. Создание пользовательских итераторов. Функция iter().

*Тема 9.2. Контейнерные типы*

Протокол контейнеров. Реализация последовательностей. Методы getitem, setitem, delitem. Эмуляция словарей.

*Тема 9.3. Перечисления и именованные константы*

Модуль enum. Создание перечислений. Атрибуты и методы перечислений. Функциональный API. IntEnum и Flag.

***Раздел 10. Работа с файловой системой***

*Тема 10.1. Файловый ввод-вывод*

Открытие файлов и режимы доступа. Контекстный менеджер with. Чтение и запись текстовых файлов. Работа с бинарными файлами. Кодировки.

*Тема 10.2. Операции с файлами и каталогами*

Модули os и os.path. Создание, переименование, удаление файлов и каталогов. Обход дерева каталогов. Модуль pathlib.

*Тема 10.3. Сериализация данных*

Модуль pickle для сохранения объектов. Форматы JSON и CSV. Модули json и csv. Работа с конфигурационными файлами. Модуль configparser.

## 6. Порядок проведения промежуточной аттестации

Порядок проведения промежуточной аттестации регламентируется локальными нормативными актами Университета:
- Положением о формах, периодичности, порядке текущего контроля успеваемости и промежуточной аттестации обучающихся по основным образовательным программам высшего образования – программам бакалавриата, программам специалитета, программам магистратуры в федеральном государственном бюджетном образовательном учреждении высшего образования «Московский государственный университет геодезии и картографии», п.6 «Организация и проведение промежуточной аттестации».
- Положением о порядке осуществления образовательного процесса, проведения текущего контроля успеваемости и промежуточной аттестации обучающихся с применением электронного обучения и дистанционных образовательных технологий в федеральном государственном бюджетном образовательном учреждении высшего образования «Московский государственный университет геодезии и картографии», п.8 «Проведение промежуточной аттестации с применением ЭО и ДОТ».

*Промежуточная аттестация: Зачет с оценкой, 3 семестр.*

## 7. Оценочные материалы текущего контроля

## 8. Оценочные материалы промежуточной аттестации

*3 семестр, Зачет с оценкой*
*Контролируемые ИДК: ОПК-5.1*
Вопросы/Задания:

1. Какой тип данных в Python является неизменяемым (immutable)?

A. list
B. dict
C. set
D. tuple

2. Что выведет следующий код: print(type(5/2))?

A. <class 'int'>
B. <class 'float'>
C. <class 'double'>
D. <class 'number'>

3. Какой оператор используется для целочисленного деления в Python?

A. /
B. //
C. %
D. div

4. Какой метод используется для добавления элемента в конец списка?

A. add()
B. insert()
C. append()
D. push()

5. Что выведет код: print("Hello"[1:4])?

A. Hel
B. ell
C. ello
D. Hell

6. Какое ключевое слово используется для определения функции в Python?

A. function
B. def
C. func
D. define

7. Что делает оператор continue в цикле?

A. Завершает выполнение цикла
B. Переходит к следующей итерации цикла
C. Приостанавливает выполнение цикла
D. Возвращает значение из цикла

8. Какой модуль используется для работы с регулярными выражениями?

A. regex
B. regexp
C. re
D. pattern

9. Что означает *args в определении функции?

A. Обязательный аргумент
B. Произвольное количество позиционных аргументов
C. Именованный аргумент
D. Аргумент по умолчанию

10. Какой метод используется для удаления пробелов в начале и конце строки?

A. remove()
B. delete()
C. strip()
D. clean()

11. Что выведет код: print(len([1, 2, [3, 4]]))?

A. 4
B. 3

C. 2
D. 5

12. Какая функция используется для генерации последовательности чисел?

A. seq()
B. sequence()
C. range()
D. numbers()

13. Что такое lambda-функция в Python?

A. Функция для работы со списками
B. Анонимная функция, определяемая в одну строку
C. Функция для математических вычислений
D. Функция высшего порядка

14. Какой оператор используется для проверки вхождения элемента в последовательность?

A. has
B. contains
C. in
D. exists

15. Какое исключение возникает при делении на ноль?

A. ValueError
B. TypeError
C. ZeroDivisionError
D. MathError

16. Какой метод используется для преобразования строки в список?

A. list()
B. split()
C. tolist()
D. array()

17. Что такое множество (set) в Python?

A. Упорядоченная коллекция элементов
B. Неупорядоченная коллекция уникальных элементов
C. Список с фиксированным размером
D. Словарь без значений

18. Какой оператор используется для остатка от деления?

A. //
B. /
C. %
D. mod

19. Какой метод используется для объединения списков?

A. merge()
B. concat()
C. extend()
D. combine()

20. Что такое декоратор в Python?

A. Функция, которая модифицирует поведение другой функции
B. Оформление кода
C. Комментарий к функции
D. Тип переменной

21. Какой модуль используется для работы с JSON?

A. jsonlib
B. json
C. simplejson

D. data

22. Что делает оператор with при работе с файлами?

A. Открывает файл для записи
B. Автоматически закрывает файл после выполнения блока кода
C. Создает новый файл
D. Читает содержимое файла

23. Какой метод используется для получения всех ключей словаря?

A. get_keys()
B. keys()
C. all_keys()
D. list_keys()

24. Что означает self в методах класса?

A. Статический метод
B. Ссылка на текущий экземпляр класса
C. Приватная переменная
D. Конструктор класса

25. Какой модуль используется для генерации случайных чисел?

A. rand
B. random
C. randint
D. rng

26. Что выведет код: print([1, 2, 3] * 2)?

A. [2, 4, 6]
B. [1, 2, 3, 1, 2, 3]
C. 6
D. [1, 2, 3, 2]

## 9. Материально-техническое и учебно-методическое обеспечение дисциплины

### 9.1. Перечень основной и дополнительной учебной литературы

*Основная литература*

1. Лысаков К. Ф. Практическое программирование на Python: учебное пособие / К. Ф. Лысаков. - Практическое программирование на Python - Новосибирск: Новосибирский государственный университет, 2023. - 75 с. - 2227-8397. - Текст: электронный // IPR SMART: [сайт]. - URL: https://www.iprbookshop.ru/134584.html - Режим доступа: по подписке

2. Сузи Р. А. Язык программирования Python: учебное пособие / Р. А. Сузи. - Язык программирования Python - Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2024. - 350 с. - 978-5-4497-3351-1. - Текст: электронный // IPR SMART: [сайт]. - URL: https://www.iprbookshop.ru/142310.html - Режим доступа: по подписке

*Дополнительная литература*

1. Амоа К. А. Разработка программных пакетов на языке Python: учебное пособие / К. А. Амоа, Н. А. Рындин, Ю. С. Скворцов. - Разработка программных пакетов на языке Python - Воронеж: Воронежский государственный технический университет, ЭБС АСВ, 2020. - 61 с. - 978-5-7731-0887-0. - Текст: электронный // IPR SMART: [сайт]. - URL: https://www.iprbookshop.ru/108184.html - Режим доступа: по подписке

### 9.2. Профессиональные базы данных и ресурсы «Интернет», к которым обеспечивается доступ обучающихся

*Профессиональные базы данных*

Не используются.

*Ресурсы «Интернет»*

1. https://docs.python.org/3/index.html - Техническая документация по языку программирования Python

### 9.3. Программное обеспечение и информационно-справочные системы, используемые при осуществлении образовательного процесса по дисциплине

*Перечень программного обеспечения*
*(обновление производится по мере появления новых версий программы)*

1. Anaconda;
2. PyCharm;
3. Microsoft Visual Studio;

*Перечень информационно-справочных систем*
*(обновление выполняется еженедельно)*

Не используется.

### 9.4. Оборудование и технические средства обучения

Компьютерный класс

Автоматизированное рабочее место (АРМ) - 25 шт.
Интерактивная панель - 1 шт.
Проектор - 1 шт.

## 10. Методические указания по освоению дисциплины (модуля)

С целью успешного освоения данного предмета студенту необходимо выполнять домашние работы по указанным преподавателям темам и представленным на занятиях материалам. Для более успешного выполнения работы студенту предлагается использование учебно-методических материалов, соответствующих указанной теме.

Самостоятельная работа студентов описывается и регулируется методическими рекомендациями по дисциплине.

***Описание возможностей изучения дисциплины лицами с ОВЗ и инвалидами***

При необходимости обучения студентов-инвалидов и лиц с ограниченными возможностями здоровья аудиторные занятия могут быть заменены или дополнены изучением полнотекстовых лекций, презентаций, видео- и аудиоматериалов в электронной информационно-образовательной среде (ЭИОС) университета. Индивидуальные задания подбираются в адаптированных к ограничениям здоровья формах (письменно или устно, в форме презентаций). Выбор методов обучения зависит от их доступности для инвалидов и лиц с ограниченными возможностями здоровья.

Рабочая программа дисциплины для инвалидов и лиц с ограниченными возможностями здоровья разрабатывается индивидуально с учетом особенностей психофизического развития, индивидуальных возможностей и состояния здоровья таких обучающихся.