



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

Факультет управления территориями  
Кафедра информационно-измерительных систем



**ДОКУМЕНТ ПОДПИСАН  
ЭЛЕКТРОННОЙ ПОДПИСЬЮ**

Сертификат: 5387FD2AF5B17FED816FDE9A1D4973B1  
Владелец: Степанченко Алексей Леонидович  
Действителен: с 21.07.2025 по 14.10.2026

«УТВЕРЖДАЮ»

Проректор  
А. Л. Степанченко  
21 апреля 2023 г.

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)  
АНАЛИЗ И СИНТЕЗ ПРОЦЕССОВ И СИСТЕМ  
«Б1.О.14.03 ЯЗЫКИ ПРОГРАММИРОВАНИЯ»**

Уровень высшего образования: бакалавриат

Направление подготовки: 27.03.03 Системный анализ и управление

Направленность(профиль) подготовки: Управление земельно-информационными системами

Квалификация (степень) выпускника: бакалавр

Форма обучения: очная

Год набора (приема на обучение): 2023

Срок получения образования: 4 года

Объем:  
в зачетных единицах: 3 з.е.  
в академических часах: 108 ак.ч.

Москва, 2023

**Разработчики:**

Преподаватель кафедры Информационно-измерительных систем  
Калмыков С. А.

Рабочая программа дисциплины (модуля) составлена в соответствии с требованиями ФГОС ВО по направлению подготовки 27.03.03 Системный анализ и управление, утвержденного приказом Минобрнауки от 07.08.2020 № 902

**Согласование и утверждение**

| № | Подразделение                           | Ответственное<br>лицо                        | ФИО             | Виза        | Дата, протокол<br>(при наличии) |
|---|-----------------------------------------|----------------------------------------------|-----------------|-------------|---------------------------------|
| 1 | факультет<br>Управления<br>территориями | Председатель<br>ученого совета<br>факультета | Воронцова А. А. | Рассмотрено | 15.12.2022, №<br>2-2022         |

## 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 - приобретение обучающимися знаний в области структурного программирования, расширение практических навыков программирования на языке Python с использованием стандартных средств разработки.

Задачи изучения дисциплины:

- знакомство студентов с основными видами языков программирования высокого уровня;
- формирование навыков работы с различными средствами программирования и отладки для создания программного обеспечения на языках высокого уровня;
- обучение основным принципам алгоритмического подхода, от этапа формализации до реализации в виде программного кода.

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

*Компетенции, индикаторы и результаты обучения*

ОПК-6 Способен разрабатывать методы моделирования, анализа и технологии синтеза процессов и систем, а также алгоритмы и программы, основанные на этих методах, пригодные для практического применения в области техники и технологии

ОПК-6.1 Демонстрирует знание методов моделирования, анализа и технологии синтеза процессов и систем, а также основанные на них алгоритмы и программы.

*Знать:*

ОПК-6.1/Зн1 Основные методы и технологии моделирования, анализа и синтеза процессов и систем, включая математические, алгоритмические и программные подходы

*Уметь:*

ОПК-6.1/Ум1 Применять языки программирования для реализации алгоритмов моделирования, анализа и синтеза процессов и систем

*Владеть:*

ОПК-6.1/Нв1 Навыками разработки и оптимизации программных алгоритмов, обеспечивающих моделирование и синтез сложных технических систем

ОПК-6.2 Разрабатывает методы моделирования, анализа и технологии синтеза процессов и систем, а также основанные на них алгоритмы и программы для практического применения в области техники и технологии

*Знать:*

ОПК-6.2/Зн1 Современные языки программирования и их парадигмы, включая императивное, объектно-ориентированное, функциональное и декларативное программирование, применяемые для разработки алгоритмов моделирования и синтеза систем

*Уметь:*

ОПК-6.2/Ум1 Разрабатывать и реализовывать алгоритмы моделирования, анализа и синтеза процессов и систем на выбранных языках программирования с использованием современных средств разработки и библиотек

*Владеть:*

ОПК-6.2/Нв1 Навыками проектирования и программной реализации комплексных алгоритмов моделирования и синтеза, включая оптимизацию и отладку программного кода

ОПК-6.3 Владеет навыками разработки методов моделирования, анализа и технологий синтеза процессов и систем, а также основанных на них алгоритмов и программ для практического применения в области техники и технологии

*Знать:*

ОПК-6.3/Зн1 Принципы разработки алгоритмов и программных решений, обеспечивающих практическое применение методов моделирования и синтеза

*Уметь:*

ОПК-6.3/Ум1 Использовать современные среды разработки и библиотеки для создания, тестирования и отладки программных продуктов

*Владеть:*

ОПК-6.3/Нв1 Навыками разработки комплексных программных решений для моделирования, анализа и синтеза технических систем

### 3. Место дисциплины в структуре ОП

Дисциплина (модуль) Б1.О.14.03 «Языки программирования» относится к обязательной части образовательной программы и изучается в семестре(ах): 1.

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 обучения | Общая трудоемкость (часы) | Общая трудоемкость (ЗЕТ) | Лекционные занятия (часы) | Практические занятия (часы) | Самостоятельная работа (часы) | Промежуточная аттестация (часы) |
|-----------------|---------------------------|--------------------------|---------------------------|-----------------------------|-------------------------------|---------------------------------|
| 1 семестр       | 108                       | 3                        | 17                        | 34                          | 57                            | Зачет                           |
| Всего           | 108                       | 3                        | 17                        | 34                          | 57                            |                                 |

### 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, соотношенные с результатами освоения программы |
|---------------------------------------|-------|--------------------|----------------------|------------------------|---------------------------------------------------------------------------------|
| Раздел 1. Введение в программирование | 15    | 2                  | 4                    | 9                      | ОПК-6.1                                                                         |

|                                                |            |           |           |           |                               |
|------------------------------------------------|------------|-----------|-----------|-----------|-------------------------------|
| Тема 1.1. Введение в программирование          | 15         | 2         | 4         | 9         |                               |
| <b>Раздел 2. Синтаксис Python</b>              | <b>18</b>  | <b>3</b>  | <b>6</b>  | <b>9</b>  | ОПК-6.1                       |
| Тема 2.1. Синтаксис Python                     | 18         | 3         | 6         | 9         |                               |
| <b>Раздел 3. Классы, интерфейсы, пакеты</b>    | <b>18</b>  | <b>3</b>  | <b>6</b>  | <b>9</b>  | ОПК-6.1                       |
| Тема 3.1. Классы, интерфейсы, пакеты           | 18         | 3         | 6         | 9         |                               |
| <b>Раздел 4. Обработка исключений</b>          | <b>19</b>  | <b>3</b>  | <b>6</b>  | <b>10</b> | ОПК-6.1                       |
| Тема 4.1. Обработка исключений                 | 19         | 3         | 6         | 10        |                               |
| <b>Раздел 5. Ввод-вывод</b>                    | <b>19</b>  | <b>3</b>  | <b>6</b>  | <b>10</b> | ОПК-6.1                       |
| Тема 5.1. Ввод-вывод                           | 19         | 3         | 6         | 10        | ОПК-6.2<br>ОПК-6.3            |
| <b>Раздел 6. Параметризация типов в Python</b> | <b>19</b>  | <b>3</b>  | <b>6</b>  | <b>10</b> | ОПК-6.1<br>ОПК-6.2<br>ОПК-6.3 |
| Тема 6.1. Параметризация типов в Python        | 19         | 3         | 6         | 10        |                               |
| <b>Итого</b>                                   | <b>108</b> | <b>17</b> | <b>34</b> | <b>57</b> |                               |

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

### *Раздел 1. Введение в программирование*

#### *Тема 1.1. Введение в программирование*

Методологические основы процесса решения задач на ПК. Виды алгоритмов. Основные алгоритмические структуры. Алгоритм разветвляющейся структуры. Алгоритм циклической структуры. Алгоритмы цикла с предусловием, цикла с постусловием. Алгоритм цикла со счетчиком.

Текущий контроль

| Вид (форма) контроля |
|----------------------|
| Тестирование         |

### *Раздел 2. Синтаксис Python*

#### *Тема 2.1. Синтаксис Python*

Основные понятия программирования на языке Python. Принципы работы с интерпретатором языка. Структура программы. Типы данных, операции, операторы.

Текущий контроль

| Вид (форма) контроля |
|----------------------|
| Тестирование         |

### *Раздел 3. Классы, интерфейсы, пакеты*

#### *Тема 3.1. Классы, интерфейсы, пакеты*

Классы. Описание класса. Описание объектов. Конструкторы объектов. Статические элементы класса. Дружественные функции. Деструкторы.

Текущий контроль

| Вид (форма) контроля |
|----------------------|
| Тестирование         |

#### ***Раздел 4. Обработка исключений***

##### *Тема 4.1. Обработка исключений*

Наследование. Шаблоны классов и обработка исключительных ситуаций.

Текущий контроль

| Вид (форма) контроля |
|----------------------|
| Тестирование         |

#### ***Раздел 5. Ввод-вывод***

##### *Тема 5.1. Ввод-вывод*

Основные операторы циклического выполнения в языке Python. Вложенные циклические структуры. операторы прерывания цикла. Функция в языке Python. Определение пользовательских функций. Файловый ввод -вывод. Чтение строк с помощью итераторов файлов.

Текущий контроль

| Вид (форма) контроля |
|----------------------|
| Тестирование         |

#### ***Раздел 6. Параметризация типов в Python***

##### *Тема 6.1. Параметризация типов в Python*

Формы наследования. Формы вложения.

Текущий контроль

| Вид (форма) контроля |
|----------------------|
| Тестирование         |

### **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, 1 семестр.*

Зачет оценивается по следующей шкале:

- оценка «зачтено» выставляется, если обучающийся имеет глубокие, исчерпывающие знания в объеме освоенной программы; знание основной (обязательной) литературы; дает правильные и уверенные ответы, свидетельствующие о наличии твердых знаний и навыков в использовании технических средств; полное, четкое, грамотное и логически стройное изложение материала; демонстрирует свободное применение теоретических знаний; а также автоматически, когда обучающийся за промежуточную аттестацию имеет оценки «отлично»;
- оценка «не зачтено» выставляется обучающемуся в случае наличия грубых ошибок при изложении ответа на основные вопросы программы, свидетельствующих о неправильном понимании предмета; при решении практических задач показано незнание способов их решения, материал изложен беспорядочно и неуверенно.

## **7. Оценочные материалы текущего контроля**

### ***Раздел 1. Введение в программирование***

*Контролируемые ИДК: ОПК-6.1*

#### ***Тема 1.1. Введение в программирование***

Форма контроля/оценочное средство: Тестирование

Вопросы/Задания:

1. Какой результат выведет следующий код?

```
x = 5
y = "5"
print(x == y)
```

- 1) True
- 2) False
- 3) Error
- 4) None

2. Что такое изменяемый (mutable) тип данных в Python?

- 1) int
- 2) str
- 3) list
- 4) tuple

3. Какой индекс у последнего элемента списка [1, 2, 3, 4, 5] ?

- 1) 5
- 2) 4
- 3) -1

4) Оба варианта b и c верны

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

1) "Hel"

2) "ell"

3) "ello"

4) "Hell"

5. Какой оператор используется для проверки принадлежности элемента списку?

1) ==

2) is

3) in

4) has

## **Раздел 2. Синтаксис Python**

*Контролируемые ИДК: ОПК-6.1*

### *Тема 2.1. Синтаксис Python*

Форма контроля/оценочное средство: Тестирование

Вопросы/Задания:

1. Что выведет следующий код?

```
for i in range(3):
```

```
    print(i)
```

1) 1 2 3

2) 0 1 2

3) 1 2

4) 0 1 2 3

2. Как правильно определить функцию в Python?

1) `function myFunc():`

2) `def myFunc():`

3) `func myFunc():`

4) `define myFunc():`

3. Что выведет `print(len("Python"))` ?

1) 5

2) 6

3) 7

4) Error

4. Какой тип данных возвращает функция `input()` ?

1) int

2) float

3) str

4) bool

5. Что выведет `print(3 // 2)` ?

1) 1.5

2) 1

3) 2

4) Error

## **Раздел 3. Классы, интерфейсы, пакеты**

*Контролируемые ИДК: ОПК-6.1*

### *Тема 3.1. Классы, интерфейсы, пакеты*

Форма контроля/оценочное средство: Тестирование

Вопросы/Задания:

1. Какая конструкция используется для обработки исключений?

- 1) if-else
- 2) try-except
- 3) switch-case
- 4) while-else

2. Что такое self в методах класса?

- 1) Ключевое слово
- 2) Ссылка на экземпляр класса
- 3) Глобальная переменная
- 4) Имя функции

3. Какой метод вызывается при создании объекта класса?

- 1) create
- 2) new
- 3) init
- 4) start

4. Как открыть файл для чтения в Python?

- 1) file = open("file.txt", "r")
- 2) file = read("file.txt")
- 3) file = open("file.txt", "read")
- 4) file = file.open("file.txt")

5. Что выведет следующий код?

```
def func(x=[]):  
    x.append(1)  
    return x  
print(len(func()))  
print(len(func()))
```

- 1) 1, 1
- 2) 1, 2
- 3) 0, 1
- 4) Error

## **Раздел 4. Обработка исключений**

*Контролируемые ИДК: ОПК-6.1*

### *Тема 4.1. Обработка исключений*

Форма контроля/оценочное средство: Тестирование

Вопросы/Задания:

1. Какая функция преобразует список в итератор?

- 1) list()
- 2) iter()
- 3) next()
- 4) range()

2. Что такое lambda в Python?

- 1) Ключевое слово цикла
- 2) Анонимная функция
- 3) Тип данных
- 4) Оператор

3. Какой модуль используется для работы с регулярными выражениями?

- 1) regex
- 2) re
- 3) regexp
- 4) pattern

4. Что выведет `print([1, 2, 3] + [4, 5])` ?

- 1) [1, 2, 3, 4, 5]
- 2) [5, 7, 8]
- 3) Error
- 4) [1, 2, 3], [4, 5]

5. Как создать пустой словарь?

- 1) `dict = []`
- 2) `dict = {}`
- 3) `dict = ()`
- 4) `dict = set()`

## **Раздел 5. Ввод-вывод**

Контролируемые ИДК: ОПК-6.1 ОПК-6.2 ОПК-6.3

### *Тема 5.1. Ввод-вывод*

Форма контроля/оценочное средство: Тестирование

Вопросы/Задания:

1. Что такое GIL в Python?

- 1) Графический интерфейс
- 2) Global Interpreter Lock
- 3) Garbage Collector
- 4) Generic Iterator Library

2. Какой декоратор делает метод статическим?

- 1) `@static`
- 2) `@staticmethod`
- 3) `@classmethod`
- 4) `@property`

3. Что выведет следующий код?

```
class A:
    x = 1
a1 = A()
a2 = A()
A.x = 2
print(a1.x, a2.x)
```

- 1) 1, 1
- 2) 2, 2
- 3) 1, 2
- 4) 2, 1

4. Какой модуль используется для создания виртуального окружения?

- 1) `virtualenv`
- 2) `venv`
- 3) `pyenv`
- 4) `env`

5. Что такое comprehension в Python?

- 1) Способ создания коллекций
- 2) Метод сжатия
- 3) Тип данных
- 4) Функция

## **Раздел 6. Параметризация типов в Python**

Контролируемые ИДК: ОПК-6.1 ОПК-6.2 ОПК-6.3

### Тема 6.1. Параметризация типов в Python

Форма контроля/оценочное средство: Тестирование

Вопросы/Задания:

1. Как правильно импортировать конкретную функцию из модуля?
  - 1) `import module.function`
  - 2) `from module import function`
  - 3) `import function from module`
  - 4) `using module import function`
2. Что выведет ``print(2 3 2)``?
  - 1) 64
  - 2) 512
  - 3) 18
  - 4) Error
3. Какой протокол должен реализовывать объект для использования в цикле `for`?
  - 1) Iterable
  - 2) Callable
  - 3) Hashable
  - 4) Comparable
4. Что такое monkey patching?
  - 1) Исправление ошибок
  - 2) Динамическое изменение кода во время выполнения
  - 3) Тестирование
  - 4) Отладка
5. Какой магический метод отвечает за преобразование объекта в строку?
  - 1) `string`
  - 2) `str`
  - 3) `text`
  - 4) `format`

## 8. Оценочные материалы промежуточной аттестации

*1 семестр, Зачет*

*Контролируемые ИДК: ОПК-6.1 ОПК-6.2 ОПК-6.3*

Вопросы/Задания:

1. Что такое переменная в языке программирования Python и как она объявляется?
2. Перечислите основные типы данных в Python и приведите примеры.
3. Объясните разницу между списком (list) и кортежем (tuple) в Python.
4. Что такое индексация в Python? Как работает отрицательная индексация?
5. Опишите основные операторы сравнения в Python.
6. Что такое срезы (slicing) в Python? Приведите примеры.
7. Объясните работу цикла `for` в Python.
8. В чем разница между циклами `for` и `while`?

9. Что такое функция в Python? Как она определяется и вызывается?
10. Объясните понятие области видимости переменных в Python.
11. Что такое исключения в Python? Как их обрабатывать?
12. Объясните принципы объектно-ориентированного программирования.
13. Что такое класс и объект в Python? Приведите пример.
14. Объясните работу с файлами в Python.
15. Что такое модули и пакеты в Python?
16. Объясните понятие итератора и генератора.
17. Что такое декораторы в Python?
18. Объясните работу с многопоточностью в Python.
19. Что такое lambda-функции в Python?
20. Объясните работу с регулярными выражениями в Python.
21. Объясните принцип работы сборщика мусора в Python.
22. Что такое метаклассы в Python?
23. Объясните паттерны проектирования Singleton и Factory в Python.
24. Что такое контекстные менеджеры в Python?
25. Объясните принципы тестирования в Python.
26. Что такое виртуальное окружение и зачем оно нужно?
27. Объясните работу с базами данных в Python.
28. Что такое асинхронное программирование в Python?
29. Объясните принципы оптимизации производительности Python-программ.
30. Что такое сериализация данных? Какие форматы поддерживает Python?

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

### **9.1. Перечень основной и дополнительной учебной литературы**

### *Основная литература*

1. Советов Б. Я. Информационные технологии: теоретические основы: учебное пособие для вузов / Б. Я. Советов, В. В. Цехановский. - 3-е изд., стер. - Санкт-Петербург: Лань, 2022. - 444 с. - 978-5-507-45305-4. - Текст: электронный // Издательство Лань: [сайт]. - URL: <https://e.lanbook.com/book/264935> - Режим доступа: по подписке

### *Дополнительная литература*

1. Цветкова А. В. Информатика и информационные технологии: учебное пособие / А. В. Цветкова. - Информатика и информационные технологии - Саратов: Научная книга, 2012. - 189 с. - 2227-8397. - Текст: электронный // IPR SMART: [сайт]. - URL: <https://www.iprbookshop.ru/6276.html> - Режим доступа: по подписке

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

### *Профессиональные базы данных*

1. <https://student2.consultant.ru/cgi/online.cgi> - Правовой Сервер КонсультантПлюс

### *Ресурсы «Интернет»*

1. <https://intuit.ru/> - Электронная образовательная среда "ИНТУИТ"
2. <https://znanium.ru/> - Электронно-библиотечная система "Znanium"
3. <https://www.elibrary.ru/> - НЭБ eLIBRARY
4. <https://rusneb.ru/> - Национальная электронная библиотека (НЭБ)

## **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. Sublime Text 3;
2. Microsoft Visual Studio;

### *Перечень информационно-справочных систем*

(обновление выполняется еженедельно)

Не используется.

## **9.4. Оборудование и технические средства обучения**

### Компьютерный класс

Маркерная доска - 1 шт.

Моноблок - 96 шт.

Проектор - 7 шт.

Автоматизированное рабочее место (АРМ) - 15 шт.

Плоттер - 1 шт.

Сканер - 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.