

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Андрей Драгомирович Хлутков  
Должность: директор  
Дата подписания: 28.10.2022 15:32:30  
Уникальный программный ключ:  
880f7c07c583b07b775f6604a630281b13ca9fd2

Приложение 7 ОП ВО

**Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«РОССИЙСКАЯ АКАДЕМИЯ НАРОДНОГО ХОЗЯЙСТВА  
И ГОСУДАРСТВЕННОЙ СЛУЖБЫ  
ПРИ ПРЕЗИДЕНТЕ РОССИЙСКОЙ ФЕДЕРАЦИИ»**

**Северо-Западный институт управления – филиал РАНХиГС**

---

Кафедра бизнес-информатики  
*(наименование кафедры)*

УТВЕРЖДЕНО  
Директор СЗИУ РАНХиГС  
А.Д.Хлутков

**ПРОГРАММА БАКАЛАВРИАТА  
«Бизнес-аналитика»**

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ  
реализуемой без применения электронного (онлайн) курса**

**ФТД.04 Программирование на языке Python**  
*(индекс, наименование дисциплины, в соответствии с учебным планом)*  
**Python**  
*(краткое наименование дисциплины)*

**38.03.05 Бизнес-информатика**  
*(код, наименование направления подготовки)*

**очная**  
*(форма обучения)*

Год набора – 2022

Санкт-Петербург, 2022 г.

**Автор–составитель:**

Доктор военных наук, кандидат технических наук, профессор, заведующий кафедрой бизнес-информатики Наумов Владимир Николаевич

**Заведующий кафедрой бизнес-информатика**  
д.в.н., профессор

Наумов Владимир Николаевич

В новой редакции РПД одобрена протоколом заседания кафедры бизнес-информатики № 9 от 04.07.2022 г

## СОДЕРЖАНИЕ

1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы
2. Объем и место дисциплины в структуре образовательной программы
3. Содержание и структура дисциплины
4. Материалы текущего контроля успеваемости обучающихся и фонд оценочных средств промежуточной аттестации по дисциплине
5. Оценочные материалы промежуточной аттестации
6. Методические материалы по освоению дисциплины
7. Учебная литература и ресурсы информационно-телекоммуникационной сети "Интернет", учебно-методическое обеспечение самостоятельной работы обучающихся по дисциплине
  - 7.1. Основная литература
  - 7.2. Дополнительная литература
  - 7.3. Учебно-методическое обеспечение самостоятельной работы
  - 7.4. Нормативные правовые документы
  - 7.5. Интернет-ресурсы
  - 7.6. Иные источники
8. Материально-техническая база, информационные технологии, программное обеспечение и информационные справочные системы

1.

**2. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения программы**

1.1. Дисциплина «Программирование на языке Python» обеспечивает овладение следующими компетенциями:

Таблица 1.1

Код компетенции	Наименование компетенции	Код этапа освоения компетенции	Наименование этапа освоения компетенции
ОПК-1	Способен проводить моделирование, анализ и совершенствование бизнес-процессов и информационно-технологической инфраструктуры предприятия в интересах достижения его стратегических целей с использованием современных методов и программного инструментария	ОПК-1.1	Способен демонстрировать понимание стратегических целей развития предприятия

В результате освоения дисциплины у студентов должны быть сформированы:

Таблица 1.2

ОТФ/ТФ (при наличии профстандарта)/ профессиональные действия	Код этапа освоения компетенции	Результаты обучения
<p>Обоснование решений D/6</p> <p>Формирование возможных решений на основе разработанных для них целевых показателей D/01/6</p> <p>Анализ, обоснование и выбор решения D/02.6</p>	ОПК-1.1	<p><b>На уровне знаний:</b></p> <ul style="list-style-type: none"> <li>- основные понятия и методы программирования, классификацию языков программирования, парадигмы программирования, основы синтаксиса языка Python;</li> </ul> <p><b>На уровне умений:</b></p> <ul style="list-style-type: none"> <li>- реализовывать алгоритмы на языке Python;</li> <li>- анализировать результаты решения задач программирования для поддержки принятия решений;</li> <li>- работать в инструментальных средах программирования Anaconda, Jupyter Notebook;</li> <li>- кодировать на языке программирования Python, использовать основные библиотеки</li> </ul>

		языка при решении простейших задач.
--	--	-------------------------------------

## 8. Объем и место дисциплины в структуре ОП ВО

### Объем дисциплины

Общая трудоемкость дисциплины составляет 1 зачетную единицу 36 академических часов.

Дисциплина реализуется частично с применением дистанционных образовательных технологий (далее – ДОТ).

Доступ к системе дистанционных образовательных технологий осуществляется каждым обучающимся самостоятельно с любого устройства на портале: <https://lms.ranepa.ru/>. Пароль и логин к личному кабинету / профилю предоставляется студенту в деканате.

Таблица 2

Вид работы	Трудоемкость в акад. часах ауд./ЭО, ДОТ	Трудоемкость в астрон. часах ауд./ЭО, ДОТ
<b>Общая трудоемкость</b>	36	27
<b>Контактная работа с преподавателем</b>	16	12
Лекции	8	6
Практические занятия	8	6
<b>Практическая подготовка</b>		
<b>Самостоятельная работа</b>	20	15
Контроль		
Формы текущего контроля	Задания, опрос, тест	
<b>Форма промежуточной аттестации</b>	Зачет	

### Место дисциплины в структуре ОП ВО

Дисциплина ФТД.04 «Программирование на языке Python» относится к факультативным дисциплинам учебного плана по направлению «Бизнес-информатика» 38.03.05. Преподавание данной дисциплины основано на школьном курсе информатики и математики, а также на положениях дисциплины Б1.О.11 «Основы информатики», изучаемой с ней одновременно. В свою очередь она создает условия для освоения следующих дисциплин: Б1.О.12 – «Программирование», Б1.О.23 «Анализ данных», Б1.О.08.06 «Эконометрика», Б1.В.01 «Нечеткая логика и нейронные сети», а также Б1.В.ДВ.03.01– «Методы прогнозирования».

Дисциплина изучается в 1-м семестре 1-го курса. Формой промежуточной аттестации в соответствии с учебным планом является зачет.

## 3. Содержание и структура дисциплины

### 3.1. Структура дисциплины

Таблица 3

№ п/п	Наименование тем	Объем дисциплины, час.						Форма текущего контроля успеваемости**, промежуточной аттестации***
		Всего	Контактная работа обучающихся с преподавателем по видам учебных занятий			СР		
			Л	ПЗ	КСР	СРО	СП	
Тема 1	Общая	18	4	4		10		О/Зад/Т

	характеристика языка Python. Типы данных и операции. Интегрированные среды разработки							
Тема 2	Функции, модули и пакеты	18	4	4		10		О/Зад/Т
Промежуточная аттестация								Зачет
Всего (акад./астр. часы):		36/27	8/6		8/6		/5,33	

Используемые сокращения:

Л – занятия лекционного типа (лекции и иные учебные занятия, предусматривающие преимущественную передачу учебной информации педагогическими работниками организации и (или) лицами, привлекаемыми организацией к реализации образовательных программ на иных условиях, обучающимся) ;

ПЗ – практические занятия (виды занятия семинарского типа за исключением лабораторных работ) ;

КСР – индивидуальная работа обучающихся с педагогическими работниками организации и (или) лицами, привлекаемыми организацией к реализации образовательных программ на иных условиях (в том числе индивидуальные консультации) ;

СР – самостоятельная работа, осуществляемая без участия педагогических работников организации и (или) лиц, привлекаемых организацией к реализации образовательных программ на иных условиях;

СП – самопроверка;

СРО – самостоятельная работа обучающегося задания (Зад), опрос (О), тестирование (Т)

### 3.2. Содержание дисциплины

#### **Тема 1. Общая характеристика языка Python. Типы данных и операции. Интегрированные среды разработки**

История языка, дзен языка. Типы данных языка. Интегрированные среды разработки. Anaconda, Ipython, Spyder, Jupiter Notebook. Организация работы в средах разработки. Арифметические операции. Операторы языка. Условные операторы. Операторы циклов. Понятие исключений. Списки, словари, кортежи, множества.

#### **Тема 2. Функции, модули и пакеты**

Построение функций языка. Объектно-ориентированное программирование. Классы в Python, Иерархия классов. Понятие модуля, библиотеки. Основы программирования модулей. Графические библиотеки. Построение графических объектов. Библиотека math, random. Библиотека numpy. Библиотека pandas. Понятие серии и набора данных. Работа с наборами данных.

### 4. Материалы текущего контроля успеваемости обучающихся

#### 4.1. Формы и методы текущего контроля успеваемости обучающихся

В ходе реализации дисциплины «Программирование на языке Python» используются следующие методы текущего контроля успеваемости обучающихся:

Таблица 4.1

Тема (раздел)	Формы (методы) текущего контроля успеваемости
Тема 1. Общая характеристика языка Python. Типы данных и операции. Интегрированные среды разработки	О/Зад/Т
Тема 2. Функции, модули и пакеты	О/Зад/Т

В дисциплине используются следующие активные и интерактивные методы обучения:

- обсуждение предложенных оценочных материалов;

- выполнение и защита задания;

- интерактивная работа по решению практических задач на компьютерах в компьютерном классе с текущим обсуждением хода и результатов решения задачи, использованию современных программных средств аналитики, data mining, интегрированных средств разработки;

- выполнение тестирования;

Признаками данных методов являются:

- активизация мышления студентов, причем учащийся вынужден быть активным;
- длительное время активности — учащийся работает не эпизодически, а в течение всего учебного процесса. Поэтому данные методы в основном реализуются на занятиях семинарского типа;
- самостоятельность в выработке и поиске решений поставленных задач;
- мотивированность к обучению путем использовать балльно-рейтинговой системы оценивания.

#### 4. 2. Типовые материалы текущего контроля успеваемости обучающихся.

##### Типовые оценочные материалы по теме 1

##### Типовые вопросы для опроса по теме 1

1. Дайте общую характеристику языка.
2. Как установить язык и интегрированную среду разработки.
3. Назовите интегрированные среды разработки. Какие возможности Jupyter notebook?
4. Почему язык приобрел такую популярность?
5. В чем отличается функциональное и объектно-ориентированное программирование на python?
6. Приведите примеры выполнения арифметических операций.
7. Назовите основные типы данных языка.
8. Что означает язык динамической типизации?
9. Приведите примеры использования управляющих конструкций языка

##### Типовые задания по теме 1

**Задание 1.** Напишите функцию, которая принимает 2 числа, сравнивает между собой и возвращает наименьшее.

**Задание 2.** Напишите функцию, которая принимает 3 числа, сравнивает между собой и возвращает количество совпадающих чисел. Соответственно, программа может возвращать одно из трех чисел: 3 - если совпадают все, 2 - если совпадают 2 числа, 0 - если все числа различны.

**Задание 3.** Даны два целых числа. Выведите значение наименьшего из них.

Подставьте "Входные данные" в свою программу и сравните результат с выходными данными.

- Входные данные: -13, -8. Выходные данные: -13

- Входные данные: 15, -8. Выходные данные: -8

- Входные данные: 3, 12. Выходные данные: 3

- Входные данные: 12, 2. Выходные данные: 2

**Задание 4.** С помощью оператора цикла Допишите функцию, которая принимает массив из 10 чисел и возвращает их сумму.

- Подставьте "Входные данные" в свою программу и сравните результат с выходными данными.

- Входные данные: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. Выходные данные: 45
- Входные данные: 2, 1, 1, 1, 1, 1, 1, 1, 1. Выходные данные: 11
- Входные данные: 8, 4, 5, 3, 9, 2, 3, 4, 6, 1. Выходные данные: 45
- Входные данные: 34, 22, 13, 18, 32, 20, 11, 14, 19, 29. Выходные данные: 212

**Задание 5.** Задан массив из n чисел. Напишите программу, которая считает и выводит количество чисел равных нулю.

**Задание 6.** Дана строка: 'Информатика'

Требуется:

1. Вывести третий символ этой строки.
2. Вывести предпоследний символ этой строки.
3. Вывести первые пять символов этой строки.
4. Вывести строку, кроме последних двух символов.
5. Вывести все символы с четными индексами (считайте, что 0 - четный индекс).
6. Вывести все символы с нечетными индексами.
7. Вывести все символы в обратном порядке.
8. Вывести все символы строки через один в обратном порядке, начиная с последнего.
9. Вывести длину данной строки.

**Задание 7.** Имеется список компаний

```
companies = ["Microsoft", "Google", "Oracle", "Apple"]
```

С помощью операторов циклов распечатать их названия. Использовать оператор for и оператор while.

**Задание 8.** Преобразовать список в словарь

```
users_list = [  
    "+111123455", "Tom",  
    "+384767557", "Bob",  
    "+958758767", "Alice"  
]
```

**Задание 9.**

1. Любимые вещи Создайте список своих любимых развлечений и сохраните его в переменной games. Теперь создайте список любимых лакомств, сохранив его в переменной foods. Объедините два этих списка, сохранив результат в переменной favorites, и напечатайте значение этой переменной.
2. Приветствие Создайте две переменные: пусть одна хранит ваше имя, а другая фамилию. Теперь с помощью строки с метками %s напечатайте приветствие вроде такого: «Привет, Владимир Наумов!».

### **Тест по теме 1**

**Задание 1.** Какая строка будет напечатана при выполнении кода

```
users = ["Tom", "Bob", "Alice", "Sam", "Bill"]  
users.sort()  
users.reverse()  
print(users)
```

Ответ: ['Tom', 'Sam', 'Bob', 'Bill', 'Alice']

**Задание 2.** Какой метод нужно использовать, чтобы добавить в конец списка строку, если список имеет вид

```
users1 = ["Tom", "Bob", "Alice"]  
ответ: append()
```

**Задание 3.** Чем отличаются списки от кортежей:



5. нет отличий кроме обозначения
6. Кортеж изменяем, список нет
7. Список изменяем, кортеж нет

**Задание 4.** Какой из операторов производит немедленный выход из цикла?

8. pass
9. continue
10. break

**Задание 5.** Что будет выведено на экран? `print(1^8)`

- 9
- 3
- 4
- 0
- 7

**Задание 6.** Что будет выведено на экран? `print(1|3|4)`

11. 1
12. 2
13. 8
14. 4
15. 7

**Задание 7.** Что вернет срез `'Python'[:]`

`'Pytho'`

`','`

`'Pyth`

`'Python'`

**Задание 8.** Какой из вариантов подключения модулей правильный?

16. `import random`
17. `import random from random`
18. `from random import as rand`

**Задание 9.** Какие из перечисленных выражений создадут список ровно из трех элементов

`print('a b c'.split())`

`print(list(range(3)))`

`print('asd'.split())`

`print('a= ',1,2,3)`

**Задание 10.** Что необходимо добавить на месте пропущенной строки

```
def find_max(nums):
```

```
    max_num=float('-inf')
```

```
    for num in nums:
```

```
        if num>max_num:
```

```
#пропущенная строка
```

```
    return max_num
```

```
find_max([1,2,3,4,6,3,19,2])
```

19. `max_num+=num`

20. `max_num=num`

21. `max_mum+=1`

22. `num=max_num`

**Задание 11.** Что напечатает следующий код

```
for i in range(4):
```

```
    if i<1:
        print(i)
    else:
        print(i)
        break
```

- 23. 0
- 24. ошибка
- 25. ничего
- 26. 0 1 2
- 27. 0 1

**Задание 12.** Какая из переменных в коде локальная, а какая глобальная?

```
def square(a,b):
```

```
    s=a*b
```

```
    return s
```

```
m=int(input('Введите число m:'))
```

```
n=int(input('Введите число n:'))
```

```
print(square(m,n))
```

- s - глобальная, m - локальная
- s - глобальная, n - локальная
- s, m - локальные, n - глобальная
- **s - локальная, m, n - глобальные**

**Задание 13.** Что напечатает следующий код?

```
a = [1, 2, 4]
```

```
a[2] = 3
```

```
print(a)
```

- 1,2,4
- 1,2,2
- 1,2,2,3
- **1,2,3**

**Задание 14.** Что из нижеперечисленного относится к кортежам?

28. (1,2,4,5)

29. [2,3,4]

30. {'a':1,'b':3}

31. 'cortege'

**Задание 15.** Что вернет срез Python[-2:]:

32. Ошибка

33. th

34. on

35. Py

**Задание 16.** Какой результат выражения 'Python'[:3]\*2?

- 'PytPyt'
- 'onon'
- 'Python'
- ошибка

**Задание 17.** Какой из операторов возвращает данные после выполнения функции?

36. return

37. break

38. continue

39. exit

40. оператор присваивания

**Задание 18.** Каков результат выполнения следующего кода?

```
var=10
```

```
print(Var)
```

- 10
- Var
- ошибка

**Задание 19.** Чем отличается set от frozenset?

41. Ничем;

42. set неизменяемое множество, frozenset –изменяемое

43. set изменяемое множество, frozenset –неизменяемое

44. в set важен порядок элементов в отличие от frozenset

**Задание 20.** Допишите функцию, чтобы после ее выполнения выдавался результат

```
def func_square(n):
```

```
    n=n**2
```

```
    #пропущенная строка
```

```
print(func_square(2))
```

Ответ: def func\_square(n):

```
    n=n**2
```

```
    #пропущенная строка
```

```
    return n
```

```
print(func_square(2))
```

**Задание 21.** Чему равен результат выполнения кода?

```
x=35
```

```
def func(x):
```

```
    x+=25
```

```
    return x
```

```
func(x)
```

```
print(x)
```

- 45. 35
- 46. 60
- 47. 25
- 48. 50

**Задание 22.** Какой или какие фрагменты кода создадут словари (dictionary):

- `tel = {'qwe', 'asd', 'zxc'}`
- `tel = {'qwe' : 1, 'asd' : 2, 'zxc' : 3}`
- `dict('asd', 'qwe', 'zxc')`
- ни один из перечисленных

### Типовые оценочные материалы по теме 2

#### Типовые вопросы для опроса по теме 2

1. Как установить модуль (библиотеку) языка?
2. Дайте общую характеристику модуля `numpy`. Какие типы данных используются в данном модуле?
3. Дайте общую характеристику модуля `pandas`. Какие типы данных используются в данном модуле?
4. Что такое класс языка? Приведите примеры построения классов.
5. Назовите основные типы диаграмм. Приведите пример построения диаграмм
6. Сгенерируйте случайную последовательность, исследуйте статистические характеристики полученной выборочной совокупности.

#### Типовые задания по теме 2

**Задание 1.** Имеются два списка. Построить диаграмму динамики ввп по годам наблюдения.

```
years=[1950,1960,1970,1980,1990,2000,2010]
gdp=[300.2,543.3,1075.9,2862.5,5979.6,10289.7,14958.3]#ВВП
```

**Задание 2.** Построить функцию нахождения корней квадратного уравнения. Проверить правильность ее построения для различных ситуаций.

**Задание 3.** Построить график функции  $y=2x+3$ , если аргумент изменяется от -100 до 100.

**Задание 4.** Задать функцию

$$f(x) = \begin{cases} -x - 1, & x < -1 \\ 0, & x \in [-1, 2] \\ x + 2, & x > 2 \end{cases}$$

С помощью функции `linspace` задать 100 значений аргумента в диапазоне от -2 до 2. Построить график функции.

49. Задание 5. Модуль `numpy`

1. Сгенерировать последовательность равномерно распределенных случайных чисел, в диапазоне от 0 до 10. Размер последовательности 100. Аналогично сгенерировать еще одну последовательность целых чисел, в диапазоне от 0 до 20 также состоящую из 100 чисел. Используя функцию `linspace()` сгенерировать последовательность из 100 чисел, заключенную от 0 до 10. Используя оператор цикла `for` и итератор `i` сформировать последовательность из 100 чисел от -1 до 9. Построить точечные, линейные диаграммы для полученных последовательностей.
2. Построить матричную диаграмму, используя стиль `matlab`. Задавать панели с помощью `subplot`. Применить средства управления цветом, типом линии, маркерами.

### **Задание 6. Модуль pandas**

1. Загрузить набор данных. Файл данных хранится в `moodle`.  

```
sp500=pd.read_csv(filepath_or_buffer="спецификация файла",sep=',',
                 usecols=['Symbol','Sector','Price','Book Value'],
                 index_col='Symbol')
sp500.head()
```

Провести анализ файла.
2. С помощью метода `iloc` отобразить только 10,20,30,40,50, 60 и 70 элементы
3. Отобразить только первые два столбца набора данных
4. С помощью метода `sample` выбрать 20 кампаний.  
50.

### **Задание 7. Создание функций и классов**

1. Создайте функцию, которая решает систему алгебраических уравнений методом обратной функции, если определитель матрицы не равен нулю. В противном случае выдается сообщение о том, что система уравнений не определена. Проверить решение на нескольких вариантах исходных данных. Сравнить с результатами решения, полученными вручную.
2. Постройте иерархию классов «Автомобиль». В иерархию классов включить классы: грузовые автомобили; легковые автомобили, гоночные автомобили, транспорт. В класс легковых автомобилей включить классы: седан, лимузин, купе, хетчбек, универсал, фургон, микроавтобус. Задать атрибуты «модель», «год выпуска». Задать свойства и методы для седана. Создать экземпляры седана

### **Тест по теме2**

**Задание 1.** Решение системы алгебраических уравнений. Система алгебраических уравнений задач скриптом

```
a = np.array([[0, 1,2], [2, 3,-1],[1,2,-1]])
v = np.array([0, 1,2], dtype=np.float64)
```

Решить данную систему. Варианты ответов.

1. система несовместна
2. [5. 1. 0.]
3. [5 1. 0]
4. [5 1 1]
5. [-1 2 1]

**Задание 2.** Двумерный массив. Сколько строк будет содержать массив, созданный с

помощью данного скрипта?

```
grid = np.arange(1, 13).reshape((4, 3)).
```

Ответ дать числом

**Задание 3.** Массив. Массив задан с помощью скрипта

```
np.random.seed(0) # seed for reproducibility
```

```
x1 = np.random.randint(10, size=6) # One-dimensional array
```

Чему равно сумма первых чисел данного одномерного массива?

**Задание 4.** модуль вектора

Чему равен модуль вектора, полученного из списка

```
s=[1,2,3,4,6]# числовой список. Ответ дать с точностью до целых в формате int
```

**Задание 5.** Обратная матрица. Чему равен определитель обратной матрицы, имеющей

вид

```
A = np.array([
    [1,2,3,5],
    [3,4,4,3],
    [2,5,6,4],
    [1,-1,2,1]
])
```

Ответ дать с точностью до двух знаков после запятой

**Задание 6.** Одномерный массив. Одномерный массив задан скриптом

```
x = np.arange(10)
```

Сколько элементов содержит его срез  $x[4:7]$ ? Ответ дать числом

**Задание 7.** Операции с массивами. Массив задан следующим образом:

```
x = np.arange(4)
```

Чему равен второй элемент массива, если каждый его элемент умножен на 2 и полученный результат возведен в квадрат?

**Задание 8.** Определитель матрицы. Чему равен определитель матрицы, полученной в результате произведения двух матриц  $B^T A$

```
A = np.array([[3,2], [-1,0], [0.5,0.5]])
```

```
B = np.array([[-5,0.5], [-10,10], [11,12]])
```

Ответ дать с точностью до целых

**Задание 9.** Срез. Одномерный массив задан с помощью скрипта.  

```
np.random.seed(1234) # seed for reproducibility
```

Сколько элементов содержит срез данного массива  $x1[::3]$ . Ответ дать числом0

## 5. Оценочные материалы промежуточной аттестации.

**Зачет проводится с применением следующих методов (средств):**

Зачет проводится в компьютерном классе в устной форме. Во время зачета проверяется уровень знаний по учебной дисциплине, а также уровень умений решать простые учебные задачи по программированию на python с использованием программных приложений. К зачету студенты должны решить задания по всем темам учебной дисциплины.

51. Промежуточная аттестация может проводиться устно в ДОТ/письменно с

прокторингом/ тестирование с прокторингом. Для успешного освоения курса учащемуся рекомендуется ознакомиться с литературой, размещенной в разделе 6, и материалами, выложенными в ДОТ.

Таблица 3

Код компетенции	Наименование компетенции	Код этапа освоения компетенции	Наименование этапа освоения компетенции
ОПК-1	Способен проводить моделирование, анализ и совершенствование бизнес-процессов и информационно-технологической инфраструктуры предприятия в интересах достижения его стратегических целей с использованием современных методов и программного инструментария	ОПК-1.1	Способен демонстрировать понимание стратегических целей развития предприятия

**Показатели и критерии оценивания компетенций с учетом этапа их формирования**

Таблица 4.3

Код компонента компетенции	Показатель оценивания	Критерий оценивания
ОПК-1.1	Демонстрирует методы программирования, основы синтаксиса языка Python;	Студент реализует алгоритмы на языке Python; анализирует результаты решения задач программирования для поддержки принятия решений; кодирует на языке программирования Python, использовать основные библиотеки

**Типовые вопросы, выносимые на зачет:**

1. Язык программирования Python: описание и философия. Дзен языка.
2. Исходный код на языке Python: кодировка, физические и логические строки, блоки кода.
3. Типы данных. Числа, строки.
4. Встроенные типы: целочисленный, вещественный, комплексный, логический
5. Арифметические операторы.
6. Операторы работы со строками.

## 7. Управляющие конструкции: операторы выбора и цикла.

### 8. Обработка исключений

9. Функции в языке Python. Лямбда-выражения.
10. Последовательности. Кортежи.
11. Списки. Срезы.
12. Словари.
13. Множества и операции над ними
14. Файлы и операции над ними
15. Стиль программирования: описание и назначение.
16. Модули и пакеты.
17. Графические библиотеки. Построение графических объектов.
18. Библиотека numpy. Объект ndarray
19. Библиотека random.
20. Понятие класса. Объектно-ориентированное программирование.
21. Библиотека pandas.
22. Понятие серии. Построение объекта «серия». Работа с сериями.
23. Понятие набора данных. Построение и работа с датафреймами.
24. Агрегирование данных и работа с групповыми данными

### Шкала оценивания.

Оценка результатов производится на основе балльно-рейтинговой системы (БРС). Использование БРС осуществляется в соответствии с приказом от 06 сентября 2019 г. №306 «О применении балльно-рейтинговой системы оценки знаний обучающихся».

Схема расчетов сформирована в соответствии с учебным планом направления, согласована с руководителем научно-образовательного направления, утверждена деканом факультета.

Схема расчетов доводится до сведения студентов на первом занятии по данной дисциплине, является составной частью рабочей программы дисциплины и содержит информацию по изучению дисциплины, указанную в Положении о балльно-рейтинговой системе оценки знаний обучающихся в РАНХиГС.

В случае если студент в течение семестра не набирает минимальное число баллов, необходимое для сдачи промежуточной аттестации, то он может заработать дополнительные баллы, отработав соответствующие разделы дисциплины, получив от преподавателя компенсирующие задания.

В случае получения на промежуточной аттестации неудовлетворительной оценки студенту предоставляется право повторной аттестации в срок, установленный для ликвидации академической задолженности по итогам соответствующей сессии.

Обучающийся, набравший в ходе текущего контроля в семестре от 51 до 70 баллов, по его желанию может быть освобожден от промежуточной аттестации.

Шкала перевода оценки из многобалльной в систему «зачтено»/«не зачтено»:

от 0 по 50 баллов	«не зачтено»
от 51 по 100 баллов	«зачтено»

## 6. Методические материалы по освоению дисциплины

Рабочей программой дисциплины предусмотрены следующие виды аудиторных занятий: лекции, практические занятия. На лекциях рассматриваются наиболее сложный



материал дисциплины. Для развития у магистрантов креативного мышления и логики в каждой теме учебной дисциплины предусмотрены теоретические положения, инструментальные средства, а также примеры их использования при решении задач предиктивной аналитики. Кроме того, часть теоретического материала предоставляется на самостоятельное изучение по рекомендованным источникам для формирования навыка самообучения.

Практические занятия предназначены для самостоятельной работы магистрантов по решению конкретных задач. Каждое практическое занятие сопровождается заданиями, выдаваемыми магистрантам для решения во внеаудиторное время.

Для работы с печатными и электронными ресурсами СЗИУ имеется возможность доступа к электронным ресурсам. Организация работы магистрантов с электронной библиотекой указана на сайте института (странице сайта – «Научная библиотека»).

### **Методические указания для обучающихся по освоению дисциплины**

Обучение по дисциплине «Введение в специальность» предполагает изучение курса на аудиторных занятиях (лекции, практические работы) и самостоятельной работы обучающихся. Семинарские занятия дисциплины «Введение в специальность» предполагают их проведение в различных формах с целью выявления полученных знаний, умений, навыков и компетенций с проведением контрольных мероприятий. С целью обеспечения успешного обучения обучающийся должен готовиться к лекции, поскольку она является важнейшей формой организации учебного процесса, поскольку:

- знакомит с новым учебным материалом;
- разъясняет учебные элементы, трудные для понимания;
- систематизирует учебный материал;
- ориентирует в учебном процессе.

Подготовка к лекции заключается в следующем:

- внимательно прочитайте материал предыдущей лекции;
- узнайте тему предстоящей лекции (по тематическому плану, по информации лектора);
- ознакомьтесь с учебным материалом по рекомендуемой литературе;
- постарайтесь уяснить место изучаемой темы в своей профессиональной подготовке;
- запишите возможные вопросы, которые вы зададите лектору на лекции.

Подготовка к практическим занятиям:

- внимательно прочитайте материал лекций, относящихся к данному семинарскому занятию, ознакомьтесь с учебным материалом;
- ответьте на контрольные вопросы по семинарским занятиям, готовьтесь дать развернутый ответ на каждый из вопросов;
- уясните, какие учебные элементы остались для вас неясными и постарайтесь получить на них ответ заранее (до семинарского занятия) во время текущих консультаций преподавателя;
- готовиться можно индивидуально, парами или в составе малой группы, последние являются эффективными формами работы;
- рабочая программа дисциплины в части целей, перечню знаний, умений, терминов и учебных вопросов может быть использована вами в качестве ориентира в организации обучения.

Выполнение задания:

- выберите набор данных (временной ряд, временные ряды) для выполнения задания;
- выполните анализ используемых признаков (целевого признака);
- проанализируйте качество исходных данных;

- выполните выбор инструментов предобработки для улучшения качества исходных данных, а также формулировки предварительных гипотез;
- решите задачу прогнозирования уровней временного ряда;
- исследуйте возможность извлечения признаков временного ряда;
- решите задачу анализа выявленных признаков;
- оформите отчет по результатам выполнения задания.

## **7. Учебная литература и ресурсы информационно-телекоммуникационной сети "Интернет", включая перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине**

### **7.1. Основная литература**

1. Городняя, Лидия Васильевна. Парадигма программирования? 2021. -231 с. Текст : электронный. - URL: <https://e.lanbook.com/book/151660> (дата обращения: 23.12.2020). - Режим доступа: для авториз. пользователей.
2. Сузи, Роман Арвиевич Python [Электронный ресурс] – СПб.: БХВ-Петербург, 2020. -350 с. Текст : электронный. - URL: <http://www.iprbookshop.ru/97589.html> (дата обращения: 16.09.2020). - Режим доступа: для авторизир. пользователей
3. Федоров, Дмитрий Юрьевич. Программирование на языке высокого уровня python [Электронный ресурс]- М.:Юрайт, 2021. Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/478098> (дата обращения: 26.07.2021)
4. Чернышев, Станислав Андреевич. Основы программирования на python. -2021. -286 с. Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/477353> (дата обращения: 04.06.2021)

Все источники основной литературы взаимозаменяемы.

### **7.2.Дополнительная литература**

1. Бринк Х., Ричардс Дж., Феверолф М. Машинное обучение. – СПб.: Питер, 2018..
2. Грас Дж. Data Science. Наука о данных с нуля. – СПб.: БХВ-Петербург, 2018.
3. Лутц М. Изучаем Python. – СПб.: Символ-Плюс, 2019.
4. Маккинли Уэс. Python и анализ данных. – М.: ДМК Пресс, 2015. -482с.
5. Мюллер А., Гвилло С. Введение в машинное обучение с помощью Python. Руководство для специалистов по работе с данными. – М., СПб: ООО Диалектика, 2019.-480с.
6. Наумов В.Н. Анализ данных и машинное обучение. Методы и инструментальные средства. – СПб.:СЗИУ, 2020.
7. Плас В. Python для сложных задач: наука о данных и машинное обучение. – СПб.: Питер, 2018, -576 с.
8. Хендрик Б., Джозеф Р., Феверолф М. Машинное обучение. – СПб.: Питер, 2018.
9. Шолле Ф. Глубокое обучение на Python. – СПб. :Питер, 2018.

### **7.3. Учебно-методическое обеспечение самостоятельной работы.**

1. Положение об организации самостоятельной работы студентов федерального государственного бюджетного образовательного учреждения высшего образования «Российская академия народного хозяйства и государственной службы при Президенте Российской Федерации» (в ред. приказа РАНХиГС от 11.05.2016 г. № 01-2211);

2. Положение о курсовой работе (проекте) выполняемой студентами федерального государственного бюджетного образовательного учреждения высшего образования «Российская академия народного хозяйства и государственной службы при Президенте Российской Федерации» (в ред. приказа РАНХиГС от 11.05.2016 г. № 01-2211)

#### **7.4. Нормативные правовые документы.**

Не используются

#### **7.5. Интернет-ресурсы.**

СЗИУ располагает доступом через сайт научной библиотеки <http://nwapr.spb.ru/> к следующим подписным электронным ресурсам:

##### **Дополнительная литература и интернет-ресурсы**

1. Курс “Introduction to Python for Data Science”  
<https://www.datacamp.com/courses/intro-to-python-for-data-science>
2. Специализация “Python for Everybody”  
<https://www.coursera.org/specializations/python>
3. Тьютор на Python <https://pythontutor.ru/lessons/dicts/>
4. Платформа REpl.it <https://repl.it/repls/GraciousRemoteLaw#main.py>

3. Wes McKinney (2011). Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython.

4. Charles Severance (2016). Python for Everybody: Exploring Data in Python 3.

5. Anastasopoulos et al. (2017). Political image analysis with deep neural networks.

6. Sebastian Raschka (2015). Python Machine Learning.

7. Эдвард Тафти (1983). Визуальное представление больших объемов информации. 8. Guillermo Moncecchi, Raul Garreta (2013). Learning scikit-learn: Machine Learning in Python.

9. Aurélien Géron (2017). Hands-On Machine Learning with Scikit-Learn and TensorFlow.

10. Марк Лутц (2011). Изучаем Python.

11. Дж. Вандер Плас (2017). Python для сложных задач. Наука о данных и машинное обучение.

12. Benjamin Bengfort, Rebecca Bilbro, Tony Ojeda (2018). Applied Text Analysis with Python.

##### **Русскоязычные ресурсы**

- Электронные учебники электронно - библиотечной системы (ЭБС) «Айбукс»
- Электронные учебники электронно – библиотечной системы (ЭБС) «Лань»
- Рекомендуется использовать следующий интернет-ресурсы
- <http://serg.fedosin.ru/ts.htm>
- <http://window.edu.ru/resource/188/64188/files/chernyshov.pdf>

#### **7.6. Иные источники.**

Не используются.

### **8. Материально-техническая база, информационные технологии, программное обеспечение и информационные справочные системы**

Курс включает использование программного обеспечения, позволяющего разрабатывать простейшие программы на языке программирования python: Anaconda, Jupyter Lab, Jupyter notebook, IPython, IDLE Python.

Интернет-сервисы и электронные ресурсы (поисковые системы, электронная почта, профессиональные тематические чаты и форумы, системы аудио и видео конференций,

онлайн энциклопедии, справочники, библиотеки, электронные учебные и учебно-методические материалы).

Для организации дистанционного обучения используется система Moodle.

№ п/п	Наименование
1.	Компьютерные классы с персональными ЭВМ, объединенными в локальные сети с выходом в Интернет
2.	Office 365 с Teams
3.	Anaconda navigator
4.	SPSS, Power BI
5.	Gogle colab
6.	Мультимедийные средства в каждом компьютерном классе и в лекционной аудитории
7.	Браузер, сетевые коммуникационные средства для выхода в Интернет
8.	Система дистанционного обучения Moodle

Компьютерные классы из расчета 1 ПЭВМ для одного обучаемого. Каждому обучающемуся должна быть предоставлена возможность доступа к сетям типа Интернет в течение не менее 20% времени, отведенного на самостоятельную подготовку.