

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ  
ПО ДИСЦИПЛИНЕ**  
Программирование на языке Python

**Код модуля**  
1159962

**Модуль**  
Алгоритмы и программирование

**Екатеринбург**

Оценочные материалы составлены автором(ами):

<b>№ п/п</b>	<b>Фамилия, имя, отчество</b>	<b>Ученая степень, ученое звание</b>	<b>Должность</b>	<b>Подразделение</b>
1	Детков Александр Александрович	кандидат экономических наук, без ученого звания	Доцент	анализа систем и принятия решений
2	Тарасьев Александр Александрович	кандидат экономических наук, без ученого звания	Доцент	анализа систем и принятия решений

**Согласовано:**

Управление образовательных программ

И.Ю. Русакова

**Авторы:**

- Детков Александр Александрович, Доцент, анализа систем и принятия решений
- Тарасьев Александр Александрович, Доцент, анализа систем и принятия решений

**1. СТРУКТУРА И ОБЪЕМ ДИСЦИПЛИНЫ Программирование на языке Python**

1.	Объем дисциплины в зачетных единицах	3	
2.	Виды аудиторных занятий	Лекции Лабораторные занятия	
3.	Промежуточная аттестация	Зачет	
4.	Текущая аттестация	Контрольная работа	1
		Домашняя работа	1

**2. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ (ИНДИКАТОРЫ) ПО ДИСЦИПЛИНЕ МОДУЛЯ Программирование на языке Python**

Индикатор – это признак / сигнал/ маркер, который показывает, на каком уровне обучающийся должен освоить результаты обучения и их предъявление должно подтвердить факт освоения предметного содержания данной дисциплины, указанного в табл. 1.3 РПМ-РПД.

Таблица 1

Код и наименование компетенции	Планируемые результаты обучения (индикаторы)	Контрольно-оценочные средства для оценивания достижения результата обучения по дисциплине
1	2	3
ПК-4 -Способен использовать информацию, методы и программные средства ее сбора, обработки и анализа для информационно-аналитической поддержки принятия управленческих решений	Д-1 - Демонстрировать целеустремленность, внимательность и ответственность при обработке информации Д-2 - Демонстрировать творческий подход и способность критического анализа результатов деятельности З-1 - Основные методы интеллектуального анализа данных П-1 - Опыт проведения регрессионного и факторного анализа данных	Домашняя работа Зачет Контрольная работа Лабораторные занятия Лекции

	<p>П-3 - Опыт решения проведения интеллектуального анализа данных с помощью языка Python (R)</p> <p>У-1 - Проводить статистический и интеллектуальный анализ данных в процессе решения поставленных задач экономики и управления</p>	
<p>ПК-13 -Способен организовать управление требованиями к программному обеспечению (ПО), продукту, средству, программно-аппаратному комплексу, автоматизированной информационной системе или автоматизированной системе управления</p>	<p>Д-1 - Личные качества: системное мышление, коммуникабельность, настойчивость в достижении цели</p> <p>З-1 - Знает технологии управления требованиями к продуктам информационной сферы</p> <p>П-2 - Опыт организации управления требованиями к продуктам информационной сферы</p> <p>У-1 - Умеет организовать управление требованиями к программному обеспечению (ПО), продукту, средству, программно-аппаратному комплексу, автоматизированной информационной системе или автоматизированной системе управления</p>	<p>Домашняя работа</p> <p>Зачет</p> <p>Контрольная работа</p> <p>Лабораторные занятия</p> <p>Лекции</p>
<p>ПК-17 -Способен оказывать консультационные услуги в сфере применения информационных технологий для повышения эффективности деятельности организации</p>	<p>Д-1 - Личные качества: коммуникабельность, внимательность, креативность</p> <p>З-2 - Методы применения информационных технологий для повышения эффективности деятельности организаций</p> <p>П-1 - Опыт консультационной деятельности в сфере применения информационных технологий</p> <p>П-2 - Опыт повышения эффективности деятельности организации</p> <p>У-1 - Применять информационные технологии</p>	<p>Домашняя работа</p> <p>Зачет</p> <p>Контрольная работа</p> <p>Лабораторные занятия</p> <p>Лекции</p>

	<p>для повышения эффективности деятельности организации</p> <p>У-2 - Консультировать по применению информационных технологий для повышения эффективности деятельности организации</p>	
<p>ПК-22 -Способен проводить исследования в области экономики, управления и финансов с применением информационных технологий, и методов интеллектуального анализа данных</p>	<p>Д-1 - Личные качества: развитый интеллект, системное мышление, креативность</p> <p>З-2 - Информационные технологии, применяемые в области экономики, управления и финансов</p> <p>З-3 - Методы интеллектуального анализа данных</p> <p>П-1 - Навыками применения информационных технологий в области экономики, управления и финансов</p> <p>П-2 - Навыками интеллектуального анализа данных</p> <p>У-2 - Применять информационные технологии для проведения исследований в области экономики, управления и финансов</p> <p>У-3 - Применять методы интеллектуального анализа данных</p>	<p>Домашняя работа</p> <p>Зачет</p> <p>Контрольная работа</p> <p>Лабораторные занятия</p> <p>Лекции</p>
<p>ПК-23 -Способен разрабатывать модели и методики проведения научных исследований в области экономики, управления и финансов с применением информационных технологий, и методов интеллектуального анализа данных</p>	<p>Д-1 - Развитый интеллект, системное мышление, креативность</p> <p>З-1 - Методики проведения научных исследований в области экономики, управления и финансов</p> <p>З-3 - Информационные технологии интеллектуального анализа данных</p> <p>П-1 - Навыками разработки моделей и методик проведения научных исследований в области экономики, управления и финансов</p> <p>У-1 - Разрабатывать модели и методики проведения научных исследований в области</p>	<p>Домашняя работа</p> <p>Зачет</p> <p>Контрольная работа</p> <p>Лабораторные занятия</p> <p>Лекции</p>

	экономики, управления и финансов	
--	----------------------------------	--

### 3. ПРОЦЕДУРЫ КОНТРОЛЯ И ОЦЕНИВАНИЯ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ В РАМКАХ ТЕКУЩЕЙ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ МОДУЛЯ В БАЛЬНО-РЕЙТИНГОВОЙ СИСТЕМЕ (ТЕХНОЛОГИЧЕСКАЯ КАРТА БРС)

#### 3.1. Процедуры текущей и промежуточной аттестации по дисциплине

<b>1. Лекции: коэффициент значимости совокупных результатов лекционных занятий – 0.6</b>		
<b>Текущая аттестация на лекциях</b>	<b>Сроки – семестр, учебная неделя</b>	<b>Максимальная оценка в баллах</b>
<i>контрольная работа</i>	17	30
<i>контроль лекций</i>	17	70
<b>Весовой коэффициент значимости результатов текущей аттестации по лекциям – 0.4</b>		
<b>Промежуточная аттестация по лекциям – зачет</b>		
<b>Весовой коэффициент значимости результатов промежуточной аттестации по лекциям – 0.6</b>		
<b>2. Практические/семинарские занятия: коэффициент значимости совокупных результатов практических/семинарских занятий – не предусмотрено</b>		
<b>Текущая аттестация на практических/семинарских занятиях</b>	<b>Сроки – семестр, учебная неделя</b>	<b>Максимальная оценка в баллах</b>
<b>Весовой коэффициент значимости результатов текущей аттестации по практическим/семинарским занятиям – не предусмотрено</b>		
<b>Промежуточная аттестация по практическим/семинарским занятиям – нет</b>		
<b>Весовой коэффициент значимости результатов промежуточной аттестации по практическим/семинарским занятиям – не предусмотрено</b>		
<b>3. Лабораторные занятия: коэффициент значимости совокупных результатов лабораторных занятий – 0.4</b>		
<b>Текущая аттестация на лабораторных занятиях</b>	<b>Сроки – семестр, учебная неделя</b>	<b>Максимальная оценка в баллах</b>
<i>выполнение лабораторных работ</i>	17	70
<i>домашняя работа</i>	17	30
<b>Весовой коэффициент значимости результатов текущей аттестации по лабораторным занятиям – 1</b>		
<b>Промежуточная аттестация по лабораторным занятиям – нет</b>		
<b>Весовой коэффициент значимости результатов промежуточной аттестации по лабораторным занятиям – не предусмотрено</b>		
<b>4. Онлайн-занятия: коэффициент значимости совокупных результатов онлайн-занятий – не предусмотрено</b>		

Текущая аттестация на онлайн-занятиях	Сроки – семестр, учебная неделя	Максимальная оценка в баллах
Весовой коэффициент значимости результатов текущей аттестации по онлайн-занятиям -не предусмотрено		
Промежуточная аттестация по онлайн-занятиям –нет Весовой коэффициент значимости результатов промежуточной аттестации по онлайн-занятиям – не предусмотрено		

### 3.2. Процедуры текущей и промежуточной аттестации курсовой работы/проекта

Текущая аттестация выполнения курсовой работы/проекта	Сроки – семестр, учебная неделя	Максимальная оценка в баллах
Весовой коэффициент текущей аттестации выполнения курсовой работы/проекта– не предусмотрено		
Весовой коэффициент промежуточной аттестации выполнения курсовой работы/проекта– защиты – не предусмотрено		

## 4. КРИТЕРИИ И УРОВНИ ОЦЕНИВАНИЯ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ МОДУЛЯ

4.1. В рамках БРС применяются утвержденные на кафедре/институте критерии (признаки) оценивания достижений студентов по дисциплине модуля (табл. 4) в рамках контрольно-оценочных мероприятий на соответствие указанным в табл.1 результатам обучения (индикаторам).

Таблица 4

### Критерии оценивания учебных достижений обучающихся

Результаты обучения	Критерии оценивания учебных достижений, обучающихся на соответствие результатам обучения/индикаторам
Знания	Студент демонстрирует знания и понимание в области изучения на уровне указанных индикаторов и необходимые для продолжения обучения и/или выполнения трудовых функций и действий, связанных с профессиональной деятельностью.
Умения	Студент может применять свои знания и понимание в контекстах, представленных в оценочных заданиях, демонстрирует освоение умений на уровне указанных индикаторов и необходимых для продолжения обучения и/или выполнения трудовых функций и действий, связанных с профессиональной деятельностью.
Опыт /владение	Студент демонстрирует опыт в области изучения на уровне указанных индикаторов.
Другие результаты	Студент демонстрирует ответственность в освоении результатов обучения на уровне запланированных индикаторов. Студент способен выносить суждения, делать оценки и формулировать выводы в области изучения. Студент может сообщать преподавателю и коллегам своего уровня собственное понимание и умения в области изучения.

4.2 Для оценивания уровня выполнения критериев (уровня достижений обучающихся при проведении контрольно-оценочных мероприятий по дисциплине модуля) используется универсальная шкала (табл. 5).

Таблица 5

**Шкала оценивания достижения результатов обучения (индикаторов) по уровням**

<b>Характеристика уровней достижения результатов обучения (индикаторов)</b>				
<b>№ п/п</b>	<b>Содержание уровня выполнения критерия оценивания результатов обучения (выполненное оценочное задание)</b>	<b>Шкала оценивания</b>		
		<b>Традиционная характеристика уровня</b>		<b>Качественная характеристика уровня</b>
1.	Результаты обучения (индикаторы) достигнуты в полном объеме, замечаний нет	Отлично (80-100 баллов)	Зачтено	Высокий (В)
2.	Результаты обучения (индикаторы) в целом достигнуты, имеются замечания, которые не требуют обязательного устранения	Хорошо (60-79 баллов)		Средний (С)
3.	Результаты обучения (индикаторы) достигнуты не в полной мере, есть замечания	Удовлетворительно (40-59 баллов)		Пороговый (П)
4.	Освоение результатов обучения не соответствует индикаторам, имеются существенные ошибки и замечания, требуется доработка	Неудовлетворительно (менее 40 баллов)	Не зачтено	Недостаточный (Н)
5.	Результат обучения не достигнут, задание не выполнено	Недостаточно свидетельств для оценивания		Нет результата

**5. СОДЕРЖАНИЕ КОНТРОЛЬНО-ОЦЕНОЧНЫХ МЕРОПРИЯТИЙ ПО ДИСЦИПЛИНЕ МОДУЛЯ**

**5.1. Описание аудиторных контрольно-оценочных мероприятий по дисциплине модуля**

**5.1.1. Лекции**

Самостоятельное изучение теоретического материала по темам/разделам лекций в соответствии с содержанием дисциплины (п. 1.2. РПД)

**5.1.2. Лабораторные занятия**

Примерный перечень тем

1. Компилируемые и интерпретируемые языки программирования: преимущества и недостатки. История разработки языка Python, общая информация, лицензия.

2. Python IDLE: особенности, интерактивный режим работы. Другие системы разработки. Использование встроенных функций help() и dir().



3. Числовой тип данных: целые, комплексные, с плавающей точкой. Системы счисления. Особенности каждого типа.
  4. Строковый тип данных: особенности. Основные методы класса STR.
  5. Лексическая структура Python: комментарии, литералы, идентификаторы. Синтаксическая структура: особенности. Блоки кода.
  6. Правила ввода литералов в Python. Строки и многострочный ввод. Числа. Динамическая типизация.
  7. Обработка исключений в программах. Встроенные типы исключений.
  8. Определение пользовательских функций. Передача параметров и возврат результатов.
  9. Значения аргументов функции по умолчанию.
  10. Произвольный набор аргументов. Именованные аргументы. Правила видимости.
  11. Работа с матрицами и векторами: основные операции с матрицами, вычислений обратной, собственных чисел и собственных значений.
  12. Построение графиков с помощью библиотеки Matplotlib.
  13. Основы функционального программирования. Генераторы списков.
  14. Арифметические операторы, операторы присваивания, логические операторы.
  15. Списки, словари, кортежи, множества в Python: ключевые особенности. Основные методы по работе со списками.
  16. Поиск и установка модулей из PIP. Создание собственного модуля. Пакеты.
  17. Создание веб-приложений с помощью модуля flask.
  18. Анализ массивов данных с использованием модуля Pandas.
- LMS-платформа – не предусмотрена

## **5.2. Описание внеаудиторных контрольно-оценочных мероприятий и средств текущего контроля по дисциплине модуля**

Разноуровневое (дифференцированное) обучение.

### **Базовый**

#### **5.2.1. Контрольная работа**

Примерный перечень тем

1. Задачи на булевый тип данных
2. Задание на срезы
3. Работа с информацией
4. Работа с текстом

Примерные задания

Задачи на булевый тип данных. Ответьте на вопрос, что выведет код для каждого случая:

```
1) a == 50
print(a == True)
print(True - 2)
print(False + True)
if 30:
print(True)
```

```
else:  
print(False)  
2) b = 0  
print(b == False)  
a = True + True  
if -1:  
print(False)  
else:  
print(True)
```

Задание на срезы. Напишите результат действия каждого из срезов:

```
test = „123456“  
test[5:6]  
test[: -3]  
test[1: -1]  
test[-5:3]  
test[-6:0: -1]
```

С помощью срезов извлеките из строки `b = „Я люблю Python“` подстроку `„Python“`.

Сумма цифр числа 498578324323 равна 58. У 58 сумма цифр — 13, а у 13 — 4. На вход вашей программе строка, содержащая число. "Сократите" его указанным образом до 1 цифры.

Большинство сайтов предоставляет возможность оставить комментарий, поэтому необходимо вести их учет, иногда можно увидеть такую запись: «Комментарии (28)». Давайте составим программу, которая будет записывать слово "комментарий" в нужной форме, например: «24 комментария». На вход вашей программе подается число, необходимо вывести слово "комментарий" в нужной форме.

Напишите программу, которая принимает на вход строку текста и вычисляет количество букв (кириллица, латиница в любом регистре), цифр и специальных символов. При выводе в первой строке указывается количество букв, во второй - количество цифр, в третьей — количество специальных символов.

Написать программу для сжатия строки, в которой алгоритм работает следующим образом: `string = 'xxxxtttcyyaaa'` преобразуется в `'x4t3c1y2a3'`, то есть последовательность одинаковых символов строки заменяется на этот символ и количество его повторений в текущей позиции строки.

Напишите функцию, которая будет возвращать самое длинное слово в предложении. Если найдено более одного слова, то функция возвращает первое.

Напишите функцию, которая возвращает самую длинную не повторяющуюся подстроку для входной строки. Если несколько подстрок совпадают по длине, функция возвращает ту, которая встречается первой.

```
xxxxx -> x  
abcdefa -> abcdef
```

Веб-сайт требует, чтобы пользователи вводили пароль для регистрации, соответствующий определенным требованиям. Напишите программу для проверки правильности ввода пароля пользователями.

Ниже приведены критерии проверки пароля:

1. Минимум 1 буква латинского алфавита в нижнем регистре [az]
2. Минимум 1 число от [0–9]

3. Минимум 1 буква латинского алфавита в верхнем регистре [AZ]
4. Минимум 1 специальный символ
5. Минимальная длина пароля : 6
6. Максимальная длина пароля: 12

Программа должна возвращать True или False.

Дан файл "titanic.csv" в формате .csv, который можно открыть командой `pd.read_csv("titanic.csv")`. Каждая строка набора данных - это отдельный человек, который либо выжил, либо не выжил (столбец "Survived"). В данной задаче необходимо определить один из статистических показателей: количество, максимум, минимум - по указанной группе пассажиров, по указанному признаку. Например, найти количество представителей выживших/не выживших пассажиров ("Survived") по столбцу "Pclass". На вход получаем: столбец по которому группируем, столбец для которого ищем статистику, название самой статистики (count, min, max). На выходе печатаем столько строк, сколько получилось после группировки по указанному столбцу. В каждой строке сначала название группы (уникальное значение столбца, по которому группировали), затем пробел, затем значение статистики для указанной группы.

Дан файл "titanic.csv" в формате .csv, который можно открыть командой `pd.read_csv("titanic.csv")`. Каждая строка набора данных - это отдельный человек, который либо выжил, либо не выжил (столбец "Survived"). В данной задаче необходимо определить долю выживших людей ("Survived" == 1) в наборе данных, которые удовлетворяют заданному составному фильтру:

1. Их возраст ("Age") строго больше заданного числа N
  2. Их Pclass строго равен заданному числу P
  3. Количество родственников ("Parch") строго равно заданному числу K
- LMS-платформа – не предусмотрена

### 5.2.2. Домашняя работа

Примерный перечень тем

1. Поиск информации
2. Работа с текстом

Примерные задания

Создайте программу, которая по почте пользователя, например, `user@myserver.com` узнает сервер этой почты.

На вход подается некоторая строка. Выведите ее с отступами в виде пробелов. Количество пробелов равно позиции символа в строке, начиная с 0.

В некоторых изданиях, например, NY Times ([www.nytimes.com](http://www.nytimes.com)) заголовки статей печатают так, чтобы каждое слово было с большой буквы. На вход вашей программе подается строка со словами, разделенными пробелом. Напишите ее как заголовок в NY Times. Может использоваться как кириллица, так и латиница.

Для входной строки вычислите символ, который встречается в ней чаще всего.

Числа Фибоначчи — элементы числовой последовательности 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765, 10946, 17711, ..., в которой первые два числа равны либо 1 и 1, либо 0 и 1, а каждое последующее число равно сумме двух предыдущих чисел. Будем нумеровать числа Фибоначчи начиная с нуля. Получая номер числа Фибоначчи, напечатайте само число.

Для строки вывести статистику по количеству входящих в нее символов (без учета регистра), сортируя по алфавиту. Игнорируйте всё, кроме букв латиницы и кириллицы. Вывод: символ, пробел, количество. Приоритет вывода у латиницы, вывод символов в нижнем регистре.

Написать программу, которая из исходной строки оставляет только уникальные слова (без учета регистра), но в том порядке, в котором они первый раз встретились. Слова разделены пробелом, вывод слов в нижнем регистре.

Дана строка текста (кириллица) со словами через пробел. Среди слов найти все пары анаграмм. Пары анаграмм вывести в алфавитном порядке, среди пар сортировка тоже по алфавиту. Каждая пара выводится в новой строке в нижнем регистре.

Создайте функцию, которая принимает переменное количество аргументов и находит среднее арифметическое ненулевых из них. Обратите внимание на формат вывода: целые числа должны выводиться без .0 .

```
1 2 3 ---> 2
```

```
2 0 0 2 2 ---> 2
```

```
2 0 2 1 1 ---> 1.5
```

Строка считается действительной, если все символы в строке встречаются одинаковое количество раз. Также допустимо, если для выполнения этого условия будет достаточно удалить 1 символ из строки. Напишите функцию, которая возвращает True, если строка действительна и False, если нет.

```
abc -> True
```

```
abcc -> True
```

На вход в программу поступает цвет CSS RGB(A), необходимо определить действителен ли его формат. Создайте функцию, которая принимает строку (например, «rgb(0, 0, 0)») и возвращает True, если формат правильный, в противном случае возвращает False. Данные могут поступать как в формате rgb, так и rgba.

Допустимые значения: rgb(0-255, 0-255, 0-255), rgb(0-100%, 0-100%, 0-100%), rgba(0-255, 0-255, 0-255, 0-1)

Возможные форматы ввода:

```
rgb(0%,50%,100%) ---> True
```

```
rgba(0,0,0,0) ---> True
```

```
rgb(255,255,255) ---> True
```

```
rgb(0,,0) ---> False
```

```
rgb(-1,0,0) ---> False
```

```
rgba(0,0,0,1.5) ---> False
```

```
rgba(0,0,0,0.5) ---> True
```

Дан файл "titanic.csv" в формате .csv, который можно открыть командой `pd.read_csv("titanic.csv")` Каждая строка набора данных - это отдельный человек, который либо выжил, либо не выжил (столбец "Survived"). В данной задаче необходимо определить количество людей в наборе данных возраст которых строго больше N. Где N - некое целое число от 0 до 100, которое подается на вход вашей функции. На выходе ожидается также целое число - количество пассажиров Титаника.

Дан файл "titanic.csv" в формате .csv, который можно открыть командой `pd.read_csv("titanic.csv")`. Каждая строка набора данных - это отдельный человек, который либо выжил, либо не выжил (столбец "Survived"). В данном наборе, в столбце "Age" есть

пропуски, то есть некоторые значения ячеек не цифры, а None. Для заполнения None в pandas есть специальный инструмент `df["Название столбца"].fillna("заполнитель")`. Где заполнителем может быть число, или строка, или другой объект, в зависимости от типа столбца. В столбце "Age" хранятся числовые объекты — float. Ваша задача принять из stdin значение (тип float) и заполнить им пропуски в столбце "Age" и вернуть среднее значение столбца "Age".

LMS-платформа – не предусмотрена

### **5.3. Описание контрольно-оценочных мероприятий промежуточного контроля по дисциплине модуля**

#### **5.3.1. Зачет**

Список примерных вопросов

1. Компилируемые и интерпретируемые языки программирования: преимущества и недостатки. История разработки языка Python, общая информация, лицензия.
2. Архитектурные особенности Python: динамическая типизация, автоматическое управление памятью, высокоуровневые структуры данных. Стили программирования на Python. Документы PEP.
3. Лексическая структура Python: комментарии, литералы, идентификаторы. Синтаксическая структура: особенности. Блоки кода.
4. Python IDLE: особенности, интерактивный режим работы. Другие системы разработки. Использование встроенных функций `help()` и `dir()`.
5. Модули Python. Способы импорта. Пространство имен программы. Примеры системных модулей.
6. Арифметические операторы, операторы присваивания, логические операторы.
7. Ветвление в Python. Операторы сравнения. Правила сравнения разных типов данных. Логический тип данных `Bool`.
8. Циклы в Python. Особенности. Принудительный выход из цикла и принудительное продолжение цикла.
9. Числовой тип данных: целые, комплексные, с плавающей точкой. Системы счисления. Особенности каждого типа.
10. Строковый тип данных: особенности. Основные методы класса `STR`.
11. Индексы и срезы в Python. Виды срезов. Примеры.
12. Правила ввода литералов в Python. Строки и многострочный ввод. Числа. Динамическая типизация.
13. Вывод содержимого на экран: оператор `R`, оператор форматирования `%`, метод `format` и `f`-строки.
14. Списки в Python: ключевые особенности. Основные методы по работе со списками: создание, изменение.
15. Кортежи в Python: ключевые особенности. Основные методы по работе с кортежами: создание, изменение.
16. Словари в Python: ключевые особенности. Основные методы по работе со словарями: создание, изменение.
17. Множества в Python: ключевые особенности. Основные методы по работе со множествами: создание, изменение.

18. Обработка исключений в программах. Встроенные типы исключений и определение новых исключений.
  19. Выполнение команд внутри контекста (with). Отличие от циклов while и for.
  20. Определение пользовательских функций. Передача параметров и возврат результатов. Значения аргументов функции по умолчанию. Произвольный набор аргументов. Именованные аргументы. Правила видимости.
  21. Основы функционального программирования. Генераторы списков. Оператор Lambda. Функции map, filter.
  22. Поиск и установка модулей из PIP. Создание собственного модуля. Пакеты.
  23. Модули Python: NumPy, Math. Основные функции, области применения.
  24. Построение графиков с помощью библиотеки Matplotlib.
  25. Анализ массивов данных с использованием модуля Pandas.
  26. Численные методы в Python: интегрирование, интерполирование.
  27. Работа с матрицами и векторами: основные операции с матрицами, вычислений обратной, собственных чисел и собственных значений.
  28. Создание веб-приложений с помощью модуля flask.
- LMS-платформа – не предусмотрена

#### 5.4 Содержание контрольно-оценочных мероприятий по направлениям воспитательной деятельности

Направление воспитательной деятельности	Вид воспитательной деятельности	Технология воспитательной деятельности	Компетенция	Результаты обучения	Контрольно-оценочные мероприятия
Профессиональное воспитание	целенаправленная работа с информацией для использования в практических целях	Технология самостоятельной работы	ПК-4	П-3	Домашняя работа Зачет Контрольная работа Лабораторные занятия Лекции