

**Краснодар 2024**

УДК 004.043 (075.8)  
ББК 32.973.1-018я73  
П96

Рецензенты: доктор физико-математических наук, профессор кафедры «Математика и информатика» Калайдин Е.Н., кандидат педагоги физико-математических наук, доцент кафедры «Математика и информатика» Кирий В.А.

Пьянкова Н.Г. Рабочая программа дисциплины «Алгоритмы и структуры данных в языке Python» для обучающихся по направлению 01.03.02 Прикладная математика и информатика, профиль «Анализ данных и принятие решений в экономике и финансах». – Краснодар: Краснодарский филиал Финуниверситета, кафедра «Математика и информатика», 2024. – 53 с.

Дисциплина «Алгоритмы и структуры данных в языке Python» относится к модулю общепрофессиональных дисциплин по направлению подготовки 01.03.02-Прикладная математика и информатика.

В рабочей программе дисциплины определены ее цель, требования к результатам освоения дисциплины, содержание программы, тематика аудиторных занятий, формы самостоятельной работы, оценочные средства для текущего контроля и промежуточной аттестации, учебно-методическое и информационное обеспечение.

Рабочая программа дисциплины алгоритмы и структуры данных в языке Python  
*Учебное издание*

*Формат 60\*90/16. Гарнитура Times New Roman*

*Усл. п.л. 2,0. Изд. № \_от.*

*Тираж 100 экз.*

*Заказ № .*

*Отпечатано в Краснодарском филиале Финуниверситета*

© Пьянкова Н.Г.  
© Краснодарский филиал Финуниверситета, 2024

## Содержание

1. Наименование дисциплины .....	4
2. Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине .....	4
3. Место дисциплины в структуре образовательной программы .....	5
4. Объем дисциплины в зачетных единицах и в академических часах с выделением объема аудиторной (лекции, семинары) и самостоятельной работы обучающихся .....	6
5. Содержание дисциплины, структурированное по темам (разделам) дисциплины с указанием их объемов (в академических часах) и видов учебных занятий .....	7
5.1 Содержание дисциплины .....	7
5.2 Учебно-тематический план .....	9
5.3 Содержание семинаров, практических занятий .....	13
6. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине .....	15
6.1 Перечень вопросов, отводимых на самостоятельное освоение дисциплины, формы внеаудиторной самостоятельной работы .....	15
6.2 Перечень вопросов, заданий, тем для подготовки к текущему контролю .....	16
7. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине .....	20
7.1 Описание показателей и критериев оценивания компетенций .....	20
7.2 Задания или иные материалы, необходимые для оценки знаний и умений, характеризующих формирование компетенций в процессе освоения ОП ВО .....	25
7.3 Методические материалы, определяющие процедуры оценивания знаний и умений, характеризующих степень сформированности компетенций .....	39
7.4 Типовые контрольные задания или иные материалы, необходимые для оценки индикаторов достижения компетенций, умений и знаний .....	40
7.5 Теоретические вопросы для подготовки к экзаменам .....	42
8. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины .....	47
9. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины .....	47
10. Методические указания для обучающихся по освоению дисциплины .....	49
11. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень необходимого программного обеспечения и информационных справочных систем .....	50
11.1 Комплект лицензионного программного обеспечения: .....	50
11.2. Современные профессиональные базы данных и информационные справочные системы ....	50
11.3 Сертифицированные программные и аппаратные средства защиты информации .....	51
12. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине .....	52

## 1. Наименование дисциплины

Дисциплина Б1.О.03.01 «Алгоритмы и структуры данных в языке Python»

## 2. Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине

Дисциплина «Алгоритмы и структуры данных в языке Python» обеспечивает инструментарий формирования следующих компетенций: ПКН-2, ПКН-4.

Код компетенции	Наименование компетенции	Индикаторы достижения компетенции	Результаты обучения (владения, умения и знания), соотнесенные с компетенциями/ индикаторами достижения компетенции
ПКН-2	Способен с помощью математической модели решать поставленную теоретическую или прикладную задачу, реализовывая алгоритм решения в виде программного модуля	1. Демонстрирует знание базовых математических моделей, применяемых в различных предметных областях.	<b>Знать:</b> существующие базовые математические модели. алгоритмы решения теоретических или прикладных задач. <b>Уметь:</b> решать прикладные задачи в предметных областях с применением математических моделей.
		2. Адаптирует и применяет существующие математические модели для решения поставленной прикладной или теоретической задачи.	<b>Знать:</b> математические модели для решения конкретных прикладных задач. подходы к применению математических моделей. <b>Уметь:</b> адаптировать и применять математические модели для решения прикладных задач.
		3. Владеет методологией математического моделирования для решения профессиональных задач.	<b>Знать:</b> современные системы математического моделирования. <b>Уметь:</b> реализовывать алгоритмы решения профессиональных задач с применением современных систем математического моделирования.
ПКН -4	Способен	1. Демонстрирует	<b>Знать:</b>

	проектировать и реализовывать прикладные программные системы в соответствии с анализом задачи	базовые знания о существующих математических методах и системах программирования.	существующие математические методы; существующие системы программирования. <b>Уметь:</b> выполнять выбор математических методов и систем программирования необходимый для решения конкретной задачи.
		2. Использует и адаптирует существующие математические методы и системы программирования для решения прикладных задач. Владеет навыками обеспечения	<b>Знать:</b> существующие математические методы и системы программирования. <b>Уметь:</b> уметь адаптировать существующие математические методы и системы программирования для решения прикладных задач
		3. Владеет навыками проектирования и разработки компонентов программного обеспечения на основе современных парадигм, технологий и языков программирования.	<b>Знать</b> Современные языки программирования, современные технологии программирования <b>Уметь:</b> Выполнять разработку прикладного программного обеспечения на основе современных парадигм, технологий и
		4. Применяет методы машинного обучения для решения прикладных задач анализа данных.	<b>Знать:</b> методы машинного обучения для решения прикладных задач анализа данных. <b>Уметь:</b> применять методы машинного обучения для решения прикладных задач анализа данных.

### 3. Место дисциплины в структуре образовательной программы

Дисциплина «Алгоритмы и структуры данных в языке Python» относится к модулю общепрофессиональных дисциплин по направлению подготовки 01.03.02. Прикладная математика и информатика, профиль «Анализ данных и принятие решений в экономике и финансах».

**4.Объем дисциплины в зачетных единицах и в академических часах с выделением объема аудиторной (лекции, семинары) и самостоятельной работы обучающихся**

**Очная форма обучения**

<b>Вид учебной работы по дисциплине</b>	<b>Всего (в з/ед. и часах)</b>	<b>Семестр 1 (в часах)</b>	<b>Семестр 2 (в часах)</b>
Общая трудоемкость дисциплины	<b>8 зач.ед. / 288 час.</b>	<b>144 час.</b>	<b>144 час.</b>
Контактная работа - Аудиторные занятия	<b>68</b>	<b>34</b>	<b>34</b>
Лекции	32	16	16
Семинары, практические занятия	36	18	18
Самостоятельная работа	<b>220</b>	<b>110</b>	<b>110</b>
В семестре	172	86	86
Контроль	48	24	24
Вид промежуточной аттестации	экзамен, зачет	экзамен	экзамен
Вид текущего контроля	Контрольные работы	Контрольная работа	Контрольная работа

**Очно-заочная форма обучения**

<b>Вид учебной работы по дисциплине</b>	<b>Всего (в з/ед. и часах)</b>	<b>Семестр 1 (в часах)</b>	<b>Семестр 2 (в часах)</b>
Общая трудоемкость дисциплины	<b>8 зач.ед. / 288 час.</b>	<b>144 час.</b>	<b>144 час.</b>
Контактная работа - Аудиторные занятия	<b>48</b>	<b>24</b>	<b>24</b>
Лекции	16	8	8
Семинары, практические занятия	32	16	16
Самостоятельная работа	<b>240</b>	<b>120</b>	<b>120</b>
В семестре	204	120	84
Контроль	36		36
Вид промежуточной аттестации	зачет, экзамен	экзамен	экзамен
Вид текущего контроля	Контрольные работы	Контрольная работа	Контрольная работа

## **5.Содержание дисциплины, структурированное по темам (разделам) дисциплины с указанием их объемов (в академических часах) и видов учебных занятий**

### **5.1 Содержание дисциплины**

#### **Тема 1. Введение в программирование на Python**

Общая информация о языке Python. История языка программирования, его связь с другими языками программирования, распространенность Python и основные сферы его применения. Знакомство с первыми примерами кода на Python. Философия Python.

Базовая информация о языке Python. Основные типы данных. Основные числовые типы данных и операции над ними. Математические операции над числовыми типами данных. Преобразование типов данных. Переменные и специфика их объявления. Статическая и динамическая типизация. Работа с переменными. Управление памятью и сборка мусора в Python. Именованые переменных.

Работа со строками: создание строк, специальные символы. Индексирование строк, получение срезов строк. Основные функции для работы со строками. Вывод на экран (работа с функцией print) и форматирование строк. Различные подходы к форматированию строк, форматирование с помощью f- строк. Расширенное форматирование в Python.

#### **Тема 2. Управляющие конструкции, списки и кортежи**

Управляющие конструкции в Python. Булев тип: объявление и операции. Операции сравнения в Python. Условные операторы в Python. Реализация задачи case в Python.

Циклы в Python: while, for. Специфика циклов в Python. Функции range и enumerate и их использование в циклах.

Списки и кортежи в Python. Специфика списков и их отличие от массивов. Создание списка, оперирование вложенными списками, копирование списков, операции над списками: индексация и срезы; изменение списка; поиск, сортировка и обход; изменение списка. Кортежи в Python: синтаксис, специфика использования.

#### **Тема 3. Словари, множества и выражения-генераторы**

Словари Python. Словари: семантика, синтаксис создания, операции над словарями, перебор элементов словаря.

Множества в Python. Множества: семантика, синтаксис создания, операции над словарями, перебор элементов словаря. Специфика операций с множествами в Python.

Выражения-генераторы в Python. Выражения-генераторы для списков: семантика и синтаксис. Пример: задача приведения списка к "плоскому" виду. Выражения-генераторы для множеств и словарей. Кейсы использования и производительность решений с использованием выражений-генераторов.

#### **Тема 4. Функции**

Функции в Python: общая семантика. Создание функции и ее вызов. Расположение определений функций. Анонимные функции в Python. Необязательные параметры функций и сопоставление по ключам. Возвращение нескольких значений из функции. Распаковка и упаковка параметров функции. Аннотации и документирование функций. Глобальные и локальные переменные.

### **Тема 5. Работа с файлами и обработка исключительных ситуаций**

Обработка исключений в Python: кейсы для использования. Инструкция `try ... except ... else ... finally`. Классы встроенных исключений. Создание пользовательских исключений. Инструкция `assert`.

Работа с файлами в Python. Концепция файла в современных ОС и языках программирования. Операции с файлами: открытие/закрытие файла, чтение и записи и другие методы для работы с файлами. Инструкция `with ... as` и ее использование для файлов.

Сохранение объектов в файл с помощью модуля `pickle` и `shelve`. Модуль `CSV`.

### **Тема 6. Модули и пакеты**

Модули и пакеты в Python: подход к структурированию программного кода с помощью модулей и пакетов. Синтаксис импортирования в Python. Создание и работа с пакетами в Python. Повторная загрузка модулей.

Написание и запуск скриптов на Python. Установка модулей из глобального репозитория.

### **Тема 7. Введение в объектно-ориентированное программирование**

Предпосылки и история появления ООП. Объекты и классы в ООП. Принципы и основные механизмы ООП. Логика работы абстракции, инкапсуляции, наследования и полиморфизма.

Python как объектно-ориентированный язык программирования. Базовые возможности ООП в Python: создание классов и объектов; наследование и полиморфизм; функция `super()`; проверка принадлежности к классу. Базовые типы в Python.

### **Тема 8. Объектно-ориентированное программирование в Python**

Методы классов и статические переменные и методы в Python. Управление доступом к атрибутам класса в Python. Динамические операции с атрибутами и интроспекция в Python. Использование специальных методов для расширенного функционала пользовательских классов. Кейс построения иерархии классов.

### **Тема 9. Введение в функциональное программирование**

Парадигмы и идиомы программирования, общая концепция функционального программирования. Функциональные языки программирования.

Функции "граждане первого класса", функции высшего порядка, замыкания, функции без побочных эффектов, рекурсия, хвостовая рекурсия. неизменяемые



структуры данных. Идиомы, распространенные в функциональных языках программирования: итераторы, последовательности, ленивые вычисления, сопоставление с образцом, монады.

Элементы функционального программирования в Python: функции - граждане первого класса; глобальные и локальные переменные в Python; вложенные функции и замыкания в Python.

Декораторы в Python: использование и создание собственных декораторов.

### **Тема 10. Функциональное программирование в Python**

Подход: map, filter, reduce. Реализация функций map, filter, reduce в Python.

Итераторы в Python, итерируемый тип данных. Модуль itertools.

Функции-генераторы и выражения-генераторы в Python.

### **Тема 11 Структуры данных: массивы, стеки, очереди, списки.**

Введение в анализ сложности алгоритмов.

Массивы и их отличие от списков в Python. Динамические массивы, сложность операций работы с динамическими массивами.

Стек, операции со стеком. Реализации стека. Очередь, операции с очередью. Реализация очереди. Связные списки, варианты связанных списков.

### **Тема 12. Алгоритмы поиска и сортировки**

Поиск в списках/массивах, бинарный поиск. Сортировка и ее использование в прикладных задачах.

Простые методы сортировки: обменные сортировки (с различными вариациями); сортировка выбором (извлечением); сортировка включением (вставками).

Эффективные методы сортировки: быстрая сортировка; сортировка Шелла; сортировка слиянием. Сравнение различных сортировок.

## **5.2 Учебно-тематический план**

### **очная форма обучения**

№ п/п	Наименование тем (разделов) дисциплины	Трудоемкость в часах					Формы текущего контроля успеваемости
		Всего	Аудиторная работа			Самостоя- тельная работа	
			Общая, в т.ч.:	Лекци и	Семинары, практиче- ские занятия		
1.	Введение в программирование на Python	18	11	2	2	9	Самостоятельное решение задач (программирование), выступление на семинаре
2.	Управляющие конструкции,	20	10	2	3	12	Самостоятельное решение задач (программирование),

	списки и кортежи						выступление на семинаре
3.	Словари, множества и выражения-генераторы	19	14	2	5	10	Самостоятельное решение задач (программирование), выступление на семинаре
4.	Функции	21	12	4	3	11	Самостоятельное решение задач (программирование), выступление на семинаре
5.	Работа с файлами и обработка исключительных ситуаций	22	9	2	2	10	Самостоятельное решение задач (программирование), выступление на семинаре
6.	Модули и пакеты	24	12	4	3	10	Самостоятельное решение задач (программирование), выступление на семинаре
	Контроль	24				24	
	Итого за 1 семестр	144	68	16	18	86	Контрольная работа
7.	Введение в объектно-ориентированное программирование	16	12	4	3	10	Самостоятельное решение задач (программирование), выступление на семинаре
8.	Объектно-ориентированное программирование в Python	22	9	2	2	10	Самостоятельное решение задач (программирование), выступление на семинаре
9.	Введение в функциональное программирование	18	12	4	3	11	Самостоятельное решение задач (программирование), выступление на семинаре
10.	Функциональное программирование в Python	24	14	2	5	10	Самостоятельное решение задач (программирование), выступление на семинаре
11.	Структуры данных: массивы, стеки, очереди, списки	18	10	2	3	12	Самостоятельное решение задач (программирование), выступление на семинаре

12.	Алгоритмы поиска и сортировки	24	11	2	2	9	Самостоятельное решение задач (программирование), выступление на семинаре
Контроль		24				24	
Итого за 2 семестр		144	68	16	18	86	Контрольная работа
В целом по дисциплине		288	136	32	36	172	Контрольные работы
Итого в %		100	47	11	13	30	

#### очно-заочная форма обучения

№ п/п	Наименование тем (разделов) дисциплины	Трудоемкость в часах					Формы текущего контроля успеваемости
		Всего	Аудиторная работа			Самостоя- тельная работа	
			Общая, в т.ч.:	Лекци и	Семинары, практиче- ские занятия		
1.	Введение в программиро- вание на Python	24	4	2	2	20	Самостоятельное решение задач (программирование), выступление на семинаре
2.	Управляющие конструкции, списки и кор- тежи	23	3	1	2	20	Самостоятельное решение задач (программирование), выступление на семинаре
3.	Словари, мно- жества и вы- ражения-гене- раторы	23	3	1	2	20	Самостоятельное решение задач (программирование) выступление на семинаре
4.	Функции	25	5	1	4	20	Самостоятельное решение задач (программирование), выступление на семинаре
5.	Работа с фай- лами и обра- ботка исклю- чительных си- туаций	24	4	2	2	20	Самостоятельное решение задач (программирование), выступление на семинаре

6.	Модули и пакеты	25	5	1	4	20	Самостоятельное решение задач (программирование), выступление на семинаре
	Итого за 1 семестр	<b>144</b>	<b>24</b>	<b>8</b>	<b>16</b>	<b>120</b>	Контрольная работа
7.	Введение в объектно-ориентированное программирование	19	4	2	4	13	Самостоятельное решение задач (программирование), выступление на семинаре
8.	Объектно-ориентированное программирование в Python	17	4	1	2	13	Самостоятельное решение задач (программирование), выступление на семинаре
9.	Введение в функциональное программирование	17	4	2	2	13	Самостоятельное решение задач (программирование), выступление на семинаре
10.	Функциональное программирование в Python	17	4	2	2	13	Самостоятельное решение задач (программирование), выступление на семинаре
11.	Структуры данных: массивы, стеки, очереди, списки	18	4	1	2	18	Самостоятельное решение задач (программирование), выступление на семинаре
12.	Алгоритмы поиска и сортировки	20	4	2	4	14	Самостоятельное решение задач (программирование), выступление на семинаре
Контроль		<b>36</b>				<b>36</b>	
Итого за 2 семестр		<b>144</b>	<b>24</b>	<b>8</b>	<b>16</b>	<b>120</b>	Контрольная работа
В целом по дисциплине		<b>288</b>	<b>52</b>	<b>20</b>	<b>32</b>	<b>236</b>	Контрольные работы
Итого в %		<b>100</b>	<b>18</b>	<b>38</b>	<b>62</b>	<b>82</b>	

### 5.3 Содержание семинаров, практических занятий

Наименование тем (разделов) дисциплины	Перечень вопросов для обсуждения на семинарских, практических занятиях, рекомендуемые источники из разделов 8, 9 (указывается раздел и порядковый номер источника)	Формы проведения занятий
1. Введение в программирование на Python	Установка Python, установка дистрибутива Anaconda. Работа в интерактивном режиме интерпретатора. Интерактивная оболочка IPython notebook: принципы работы и применения. Рекомендуемая литература: 8 (1, 2); 9 (1-4)	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий (не менее 30% времени на интерактивные технологии)
2. Управляющие конструкции, списки и кортежи	Базовые числовые типы, строки, списки, словари, переменные, базовые операторы. Рекомендуемая литература: 8 (1, 2, 3, 4); 9 (1-10)	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий (не менее 30% времени на интерактивные технологии)
3. Словари, множества и выражения генераторы	Множества в Python. Множества: семантика, синтаксис создания, операции над словарями, перебор элементов словаря. Специфика операций с множествами в Python. Выражения-генераторы в Python. Выражения-генераторы для списков: семантика и синтаксис. Рекомендуемая литература: 8 (1, 2); 9 (1-4)	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий (не менее 30% времени на интерактивные технологии)
4. Функции	Создание функций, область видимости переменной, передача аргументов в функцию. Лямбда- функции. Рекомендуемая литература 8 (1, 2, 3, 4); 9 (1-11)	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий (не менее 30% времени на интерактивные технологии)
5. Работа с файлами и обработка исключительных ситуаций	Исключения. Инструкция try... except... else... finally. Классы встроенных исключений. Создание пользовательских исключений. Инструкция assert. Работа с файлами в Python, операции с файлами: открытие/закрытие файла, чтение и записи и другие методы для работы с файлами. Инструкция with ... as и ее использование для файлов. Рекомендуемая литература: 8 (1, 2, 3, 4); 9 (1-11)	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий (не менее 30% времени на интерактивные технологии)
6. Модули и пакеты	Устройство модулей и пакетов, инструкции import и from. Создание собственных модулей и пакетов. Рекомендуемая литература: 8 (1, 2, 3, 4); 9 (1-10)	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий (не менее 30% времени на интерактивные технологии)

		технологии)
7. Введение в объектно-ориентированное программирование	<p>Базовые возможности ООП в Python: создание классов и объектов; наследование и полиморфизм; функция super().</p> <p>Рекомендуемая литература: 8 (1, 2, 3, 4); 9 (1-11)</p>	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий (не менее 30% времени на интерактивные технологии)
8.Объектно-ориентированное программирование в Python	<p>Методы классов и статические переменные и методы в Python. Управление доступом к атрибутам класса в Python. Динамические операции с атрибутами и интроспекция в Python. Использование специальных методов для расширенного функционала пользовательских классов. Кейс построения иерархии классов.</p> <p>Рекомендуемая литература: 8 (1, 2, 3, 4); 9 (1-10)</p>	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий (не менее 30% времени на интерактивные технологии)
9. Введение в функциональное программирование	<p>Элементы функционального программирования в Python: функции - граждане первого класса; глобальные и локальные переменные в Python; вложенные функции и замыкания в Python. Декораторы в Python: использование и создание собственных декораторов.</p> <p>Рекомендуемая литература: 8 (1, 2, 3, 4); 9 (1-10)</p>	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий (не менее 30% времени на интерактивные технологии)
10.Функциональное программирование в Python	<p>Реализация функций map, filter, reduce в Python. Итераторы в Python, итерируемый тип данных. Модуль itertools.</p> <p>Рекомендуемая литература: 8 (1, 2, 3, 4); 9 (1-11)</p>	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий (не менее 30% времени на интерактивные технологии)
11. Структуры данных: массивы, стеки, очереди, списки	<p>Массивы и их отличие от списков в Python. Динамические массивы, сложность операций работы с динамическими массивами. Стек, операции со стеком. Реализации стека. Очередь, операции с очередью. Реализация очереди. Связные списки, варианты связанных списков.</p> <p>Рекомендуемая литература: 8 (1, 2, 3, 4); 9 (1-10)</p>	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий (не менее 30% времени на интерактивные технологии)
12 Алгоритмы поиска и сортировки	<p>Реализация на Python простых методов сортировки: обменные сортировки (с различными вариациями); сортировка выбором (извлечением); сортировка включением (вставками).</p>	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий (не менее 30%

	Реализация на Python эффективных методов сортировки: быстрая сортировка; сортировка Шелла; сортировка слиянием. Рекомендуемая литература: 8 (1, 2, 3, 4); 9 (1-11)	времени на интерактивные технологии)
--	---	--------------------------------------

## 6. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине

### 6.1 Перечень вопросов, отводимых на самостоятельное освоение дисциплины, формы внеаудиторной самостоятельной работы

Наименование тем (разделов) дисциплины	Перечень вопросов, отводимых на самостоятельное освоение	Формы внеаудиторной самостоятельной работы
1. Введение в программирование на Python	Среда программирования. Использование документации.	Индивидуальное выполнение заданий с использованием Jupyter Notebook.
2. Управляющие конструкции, списки и кортежи	Оперирование вложенными списками, копирование списков, некоторые операции над списками.	Индивидуальное выполнение заданий с использованием Jupyter Notebook.
3. Словари, множества и выражения-генераторы	Выражения-генераторы в Python. Выражения-генераторы для словарей и множеств: семантика, синтаксис и практическое использование.	Индивидуальное выполнение заданий с использованием Jupyter Notebook.
4. Функции	Возвращение нескольких значений из функции. Распаковка и запаковка параметров функции.	Индивидуальное выполнение заданий с использованием Jupyter Notebook.
5. Работа с файлами и обработка исключительных ситуаций	Сохранение объектов в файл с помощью модуля pickle и shelve. Модуль CSV.	Индивидуальное выполнение заданий с использованием Jupyter Notebook.
6. Модули и пакеты	Написание и запуск скриптов на Python. Установка модулей из глобального репозитория.	Индивидуальное выполнение заданий с использованием Jupyter Notebook.
7. Введение в объектно-ориентированное программирование	Базовые типы в Python: взгляд с точки зрения ООП. Методы базовых типов.	Индивидуальное выполнение заданий с использованием Jupyter Notebook.
8. Объектно-ориентированное программирование в Python	Проверка принадлежности к классу и интроспекция в Python.	Индивидуальное выполнение заданий с использованием Jupyter Notebook.
9. Введение в функциональное программирование	Реализация декораторов с параметрами в Python.	Индивидуальное выполнение заданий с использованием Jupyter Notebook.

10 Функциональное программирование в Python	Функции-генераторы и выражения-генераторы в Python.	Индивидуальное выполнение заданий с использованием Jupyter Notebook.
11. Структуры данных: массивы, стеки, очереди, списки	Реализация различных вариантов связанных списков на Python.	Индивидуальное выполнение заданий с использованием Jupyter Notebook.
12 Алгоритмы поиска и сортировки	Сравнение различных сортировок (простых и эффективных) с использованием их реализаций на Python.	Индивидуальное выполнение заданий с использованием Jupyter Notebook.

## 6.2 Перечень вопросов, заданий, тем для подготовки к текущему контролю

### Примерные вопросы к контрольной работе для 1 семестра

1. Операции над основными числовыми типами данных.
2. Операции над булевыми переменными.
3. Динамическая типизация в Python.
4. Преобразование типов в Python.
5. Создание строк в Python.
6. Организация и пример цикла while в Python.
7. Организация и пример цикла for в Python.
8. Операции над словарями в Python.
9. Выражения-генераторы для списков в Python.
10. Необязательные параметры функций в Python.
11. Заpackовка и распаковка параметров в Python.
12. Аннотации и документирование функций.
13. Создание объектов в Python.
14. Управление доступом к атрибутам класса в Python.
15. Выполнение интроспекции в Python.
16. Замыкания в Python.
17. Использование декораторов в Python.
18. Создание собственных декораторов в Python.
19. Компилятор и интерпретатор. Достоинства и недостатки.
20. Назовите и дайте краткую характеристику основных классов языков программирования.
21. Встроенные числовые типы языка Python.
22. Списки. Создание, основные операции.
23. Основные методы списка.
24. Кортежи. Создание, основные методы и операции.
25. Словари. Создание, основные операции.
26. Методы для работы со словарями.
27. Множества. Создание, основные методы и операции.
28. Переменные. Правила именования переменных.
29. Динамическая типизация.
30. Операторы сравнения и логические операторы.
31. Инструкция if...else.



32. Инструкция цикла while. 33. Инструкция цикла for.  
34. Создание и вызов функции. 35. Передача аргументов функцию.  
36. Функции-генераторы.  
37. Лямбда-функции.  
38. Модули. Инструкции import и from.

### Примеры заданий к контрольной работе для 1 семестра

1. В строке содержащей последовательность слов, разделенных запятыми удалить все нечетные слова. Ответ представить в виде строки. Пример: строка 'SIX,SEVEN,EIGHT,NINE,TEN' будет преобразована в: 'SIX,EIGHT,TEN'.

2. Из списка списков элементами которого являются текстовые символы собрать строку, в которой вложенные списки объединены в слова, а слова через запятую объединены в строку. Пример список вида [['E', 'e', 'n', 'y'], ['m', 'e', 'e', 'n', 'y'], ['m', 'i', 'n', 'e', 'y'], ['m', 'o', 'e']] будет преобразован в строку 'Еeny,meeny,miney,мое'.

3. Используя генератор словарей (и не используя код вне него) инвертировать словарь, т.е. сделать ключи словаря, его значениями и наоборот. Значения, которые в исходном словаре повторяются не добавлять в итоговый словарь. Пример: {'a':1, 'b':3, 'c':4, 'd':3} -> {1:'a', 4:'c'}

4. Задана строка, в которой через запятую перечислены имена людей (с заглавной буквы) и их текущие занятия (со строчной буквы) в произвольном порядке (например, "Иван ест, поет Оля" и т.д.). С помощью генераторов создать словарь, в котором ключами будут имена, а значениями – занятия. Решить задачу в одну строку. Например: "Маша гуляет, Коля работает, дома Ваня, закупаются Женя" представить в виде {'Ваня': 'дома', 'Женя': 'за-купается', 'Коля': 'работает', 'Маша': 'гуляет'}.

5. Написать скрипт, который заменяет в текстовом файле все слова из определенного перечня на определенные для этих слов слова-заменители. Скрипт принимает через командную строку 2 параметра:

первый параметр - имя файла из которого берется текст (файл имеет кодировку UTF8; нет переносов слов на другую строку);

второй параметр - имя файла, являющегося файлом CSV (с разделителем ;) содержащего перечень слов для замены и соответствующих им заменителей (файл не имеет заголовка столбцов; в первом столбце слова подлежащие замене, во втором слова-заменители) (файл имеет кодировку UTF8). Результат замены слов сохраняется в новом файле с именем совпадающем с исходным файлом с текстом, но имеющим префикс подчеркивание ( \_ ). Скрипт должен иметь вид replace\_Ivanov.py (вместо Ivanov - ваша фамилия). Функции связанные с обработкой файлов и заменой в тексте должны находиться в отдельном модуле modul\_Ivanov.py (вместо Ivanov - ваша фамилия).

6. Реализовать функцию summate для расчета накопленных двойных сумм (квадратов произведений). Функция принимает одно или более числовое значение (количество параметров заранее не определено). На основе этих значений рассчитываются накопленные суммы, которые сохраняются в списке, список возвращается как результат функции. Необязательный булевский параметр mul должен

позволять заменять суммирование умножением.

### **Примерные вопросы к контрольной работе для 2 семестра**

1. Базовые принципы объектно-ориентированного программирования.
2. Класс, метод класса, атрибут класса. Определение класса и создание экземпляра класса.
3. Конструктор и деструктор.
4. Наследование.
5. Абстрактные методы класса.
6. Статические методы класса.
7. Свойства класса.
8. Исключения. Обработка исключений.
9. Пользовательские исключения.
10. Вложенные функции и замыкания, специфика реализации в Python.
11. Функции высшего порядка и декораторы в Python.
12. Реализация map/filter/reduce в Python и пример их использования.
13. Итераторы в Python.
14. Встроенные функции для работы с итераторами и возможности модуля itertools.
15. Специфика массивов, как структур данных.
16. Абстрактная структура данных стек и очередь: базовые и расширенные операции, их сложность.
17. Реализация основных операций в очереди на базе массива и связанного списка.
18. Связанные списки: однонаправленные и двунаправленные – принцип реализации.
19. Алгоритм обменной сортировки.
20. Алгоритм сортировки выбором.
21. Алгоритм сортировки вставками.
22. Алгоритм быстрого поиска в отсортированном массиве.
23. Алгоритм сортировки Шелла.
24. Алгоритм быстрой сортировки.
25. Алгоритм сортировки слиянием.
26. Структуры данных: деревья
27. Реализация двоичных деревьев в виде связанных объектов. Различные реализации рекурсивного обхода двоичных деревьев.
28. Двоичная куча, реализации основных операций.
29. Абстрактный тип данных - ассоциативный массив и принцип его реализации на основе хэш-таблиц и хэш-функций.
30. Хэш-функции multiply-add-and-divide. Принцип работы хэш-функции multiply-add-and-divide.
31. Полиномиальная хэш-функция.

## Примеры заданий к контрольной работе для 2 семестра

1. Создать иерархию классов для фруктов, продающихся в магазине. Иерархия должна содержать не менее 3 классов. Объекты должны содержать не менее 2-х атрибутов и 2-х методов. Реализовать механизм автоматического подсчета количества всех созданных фруктов и автоматического присвоения каждому фрукту уникального идентификатора. Необходимо заполнить список представителями всех классов (всего не менее 10 объектов) и продемонстрировать работу созданного механизма.

2. Создайте класс Length (Длина), имеющий свойства:

- value (значение),
- unit (единица измерения).

При изменении единицы измерения значение должно соответственно меняться. Например, при переходе от сантиметров к метрам значение

должно уменьшаться в 100 раз. Допустимые значения свойства unit: 'см', 'м', 'км'.

Организируйте эту проверку. Протестируйте работу с классом.

3. Создайте класс Заказ(Order), у которого есть свойства код\_товара(code), цена(price), количество(count) и методы init и str. Создайте 2 класса-потомка: Опт(Opt) и Розница(Retail). В этих классах создайте методы init, \_\_str\_\_ и сумма\_заказа(summa), позволяющий узнать стоимость заказа. Для опта стоимость единицы товара составляет 95% от цены, а при покупке более 500 штук – 90% цены. В розницу стоимость единицы товара составляет 100% цены. Стоимость заказа равна произведению цены на количество. Протестируйте работу с классами, создав необходимые объекты и обратившись к их свойствам и методам.

4. Написать функцию-генератор my\_func\_2(lst), которая принимает объект, поддерживающий итерации с произвольным уровнем вложенности, и возвращает все элементы по одному.

5. С помощью механизма map/filter/reduce (хотя бы одна из этих функций должна быть использована в решении) посчитать в тексте количество слов, состоящих не менее, чем из 3-х букв. Слова в тексте разделены пробелами. Написать реализацию в одну строку. Оформить решение в виде функции my\_func\_3(text), т.е. шаблон таков: строка с import, если необходимо def my\_func\_3(text): return # однострочная реализация задания.

Написать декоратор с параметром my\_decorator(n). Декоратор превращает функцию, возвращающую поддерживающий итерации объект (далее "последовательность"), в функцию-генератор. Если декорируемая функция возвращает что-то другое, а не последовательность, то декоратор должен вернуть этот результат вызова функции без изменений. Проверку объекта можно организовать при помощи условия import collections if isinstance(item, collections.Iterable). Параметром декоратора может быть целое положительное число n, тогда получившаяся функция-декоратор должна генерировать по одному значению из последовательности, повторенной n раз. Также параметр может принимать строковое значение 'inf', тогда функция-декоратор генерирует по одному значению из последовательности, повторенной бесконечное число раз (зацикливает генерирование результата). Подсказка: сначала реализовать случай со значением аргумента 'inf', а затем модифицировать для целочисленного

значения аргумента.

### *Критерии балльной оценки различных форм текущего контроля успеваемости*

Критерии балльной оценки различных форм текущего контроля успеваемости содержатся в соответствующих методических рекомендациях кафедры «Математика и информатика».

## **7. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине**

Перечень компетенций с указанием индикаторов их достижения в процессе освоения образовательной программы содержится в разделе 1. «Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине».

### **Паспорт фонда оценочных средств**

Оценочные средства предназначены для контроля и оценки образовательных достижений обучающихся, освоивших программу учебной дисциплины «Алгоритмы и структуры данных в языке PYTHON».

Фонд оценочных средств включает контрольные материалы для проведения текущего контроля и промежуточной аттестации.

### **7.1 Описание показателей и критериев оценивания компетенций**

Планируемые результаты освоения компетенции (индикатора достижения компетенции)	Уровень освоения				Оценочное средство
	«неудовлетворительно»	«удовлетворительно»	«хорошо»	«отлично»	
<b>ПKN-2 Способен с помощью математической модели решать поставленную теоретическую или прикладную задачу, реализовывая алгоритм решения в виде программного модуля</b>					
Демонстрирует знание базовых математических моделей, применяемых в различных предметных областях.					
Знать: - существующие базовые математические модели. - алгоритмы решения теоретических или прикладных	Фрагментарное представление о существующих базовых математических моделях и алгоритмах решения теоретически	Неполные представления о существующих базовых математических моделях и алгоритмах решения теоретически или	Сформированные, но содержащие отдельные пробелы представления о существующих базовых математических моделях и	Сформированные систематические представления о существующих базовых математических моделях и алгоритмах	Вопросы для оценки знаний и умений, задания в виде расчетных задач, тестовые задания

Планируемые результаты освоения компетенции (индикатора достижения компетенции)	Уровень освоения				Оценочное средство
	«неудовлетворительно»	«удовлетворительно»	«хорошо»	«отлично»	
задач.	х или прикладных задач	прикладных задач	алгоритмах решения теоретических или прикладных задач	решения теоретических или прикладных задач	
Уметь: решать прикладные задачи в предметных областях с применением математических моделей.	Фрагментарное умение решать прикладные задачи в предметных областях с применением математических моделей.	Несистематическое применение решать прикладные задачи в предметных областях с применением математических моделей.	В целом успешное, но содержащее отдельные пробелы умение решать прикладные задачи в предметных областях с применением математических моделей.	Сформированное умение решать прикладные задачи в предметных областях с применением математических моделей.	Вопросы для оценки знаний и умений, задания в виде расчетных задач, тестовые задания
Адаптирует и применяет существующие математические модели для решения поставленной прикладной или теоретической задачи.					
Знать: - математические модели для решения конкретных прикладных задач. - подходы к применению математических моделей.	Фрагментарное представление о математических моделях для решения конкретных прикладных задач, подходах к применению математических моделей	Неполные представления о математических моделях для решения конкретных прикладных задач, подходах к применению математических моделей	Сформированные, но содержащие отдельные пробелы представления о математических моделях для решения конкретных прикладных задач, подходах к применению математических моделей	Сформированные систематические представления о математических моделях для решения конкретных прикладных задач, подходах к применению математических моделей	Вопросы для оценки знаний и умений, задания в виде расчетных задач, тестовые задания
Уметь: адаптировать и применять математические модели для	Фрагментарное умение адаптировать и применять математических	Несистематическое применение умений адаптировать	В целом успешное, но содержащее отдельные пробелы	Сформированное умение адаптировать и применять математических	Вопросы для оценки знаний и умений, задания в виде расчетных

Планируемые результаты освоения компетенции (индикатора достижения компетенции)	Уровень освоения				Оценочное средство
	«неудовлетворительно»	«удовлетворительно»	«хорошо»	«отлично»	
решения прикладных задач.	ие модели для решения прикладных задач	и применять математическое модели для решения прикладных задач.	умение адаптировать и применять математическое модели для решения прикладных задач.	ие модели для решения прикладных задач.	задач, тестовые задания
Владеет методологией математического моделирования для решения профессиональных задач.					
Знать: - современные системы математического моделирования.	Фрагментарное представление о современных системах математического моделирования	Неполные представления о современных системах математического моделирования	Сформированные, но содержащие отдельные пробелы представления о современных системах математического моделирования	Сформированные систематические представления о современных системах математического моделирования	Вопросы для оценки знаний и умений, задания в виде расчетных задач, тестовые задания
Уметь: - реализовывать алгоритмы решения профессиональных задач с применением современных систем математического моделирования.	Фрагментарное умение реализовывать алгоритмы решения профессиональных задач с применением современных систем математического моделирования	Несистематическое применение умений реализовывать алгоритмы решения профессиональных задач с применением современных систем математического моделирования.	В целом успешное, но содержащее отдельные пробелы умение реализовывать алгоритмы решения профессиональных задач с применением современных систем математического моделирования.	Сформированное умение реализовывать алгоритмы решения профессиональных задач с применением современных систем математического моделирования.	Вопросы для оценки знаний и умений, задания в виде расчетных задач, тестовые задания
<b>ПКН -4 Способен проектировать и реализовывать прикладные программные системы в соответствии с анализом задачи и требований к ним</b>					
Демонстрирует базовые знания о существующих математических методах и системах программирования.					

Планируемые результаты освоения компетенции (индикатора достижения компетенции)	Уровень освоения				Оценочное средство
	«неудовлетворительно»	«удовлетворительно»	«хорошо»	«отлично»	
Знать: - существующие математические методы; - существующие системы программирования.	Фрагментарное представление о существующих математических методах, системах программирования	Неполные представления о существующих математических методах, системах программирования	Сформированные, но содержащие отдельные пробелы представления о существующих математических методах, системах программирования	Сформированные систематические представления о существующих математических методах, системах программирования	Вопросы для оценки знаний и умений, задания в виде расчетных задач, тестовые задания
Уметь: - выполнять выбор математических методов и систем программирования необходимый для решения конкретной задачи.	Фрагментарное умение выполнять выбор математических методов и систем программирования необходимый для решения конкретной задачи.	Несистематическое применение выполнять выбор математических методов и систем программирования необходимый для решения конкретной задачи.	В целом успешное, но содержащее отдельные пробелы умение выполнять выбор математических методов и систем программирования необходимый для решения конкретной задачи.	Сформированное умение выполнять выбор математических методов и систем программирования необходимый для решения конкретной задачи.	Вопросы для оценки знаний и умений, задания в виде расчетных задач, тестовые задания
Использует и адаптирует существующие математические методы и системы программирования для решения прикладных задач. Владеет навыками обеспечения					
Знать: - существующие математические методы и системы программирования.	Фрагментарное представление о существующих математических методах и системах программирования	Неполные представления о существующих математических методах и системах программирования	Сформированные, но содержащие отдельные пробелы представления о существующих математических методах и	Сформированные систематические представления о существующих математических методах и	Вопросы для оценки знаний и умений, задания в виде расчетных задач, тестовые задания

Планируемые результаты освоения компетенции (индикатора достижения компетенции)	Уровень освоения				Оценочное средство
	«неудовлетвори тельно»	«удовлетвори тельно»	«хорошо»	«отлично»	
			системах программиров ания	программиров ания	
Уметь: - адаптировать существующи е математическ ие методы и системы программиров ания для решения прикладных задач	Фрагментарно е умение адаптировать существующи е математическ ие методы и системы программиров ания для решения прикладных задач	Несистематич еское применение умений адаптировать существующи е математическ ие методы и системы программиров ания для решения прикладных задач	В целом успешное, но содержащее отдельные пробелы умение адаптировать существующи е математическ ие методы и системы программиров ания для решения прикладных задач	Сформирован ное умение адаптировать существующи е математическ ие методы и системы программиров ания для решения прикладных задач	Вопросы для оценки знаний и умений, задания в виде расчетных задач, тестовые задания
Владеет навыками проектирования и разработки компонентов программного обеспечения на основе современных парадигм, технологий и языков программирования.					
Знать: - современные языки программиров ания, современные технологии программиров ания	Фрагментарно е представлени е о современных языках программиров ания, современных технологиях программиров ания	Неполные представлени я о современных языках программиров ания, современных технологиях программиров ания	Сформирован ные, но содержащие отдельные пробелы представлени я о современных языках программиров ания, современных технологиях программиров ания	Сформирован ные систематичес кие представлени я о современных языках программиров ания, современных технологиях программиров ания	Вопросы для оценки знаний и умений, задания в виде расчетных задач, тестовые задания
Уметь: - выполнять разработку прикладного программного обеспечения на основе	Фрагментарно е умение выполнять разработку прикладного программного обеспечения	Несистематич еское применение умений выполнять разработку прикладного	В целом успешное, но содержащее отдельные пробелы умение выполнять	Сформирован ное умение выполнять разработку прикладного программного обеспечения	Вопросы для оценки знаний и умений, задания в виде расчетных задач, тестовые задания



Планируемые результаты освоения компетенции (индикатора достижения компетенции)	Уровень освоения				Оценочное средство
	«неудовлетворительно»	«удовлетворительно»	«хорошо»	«отлично»	
современных парадигм и технологий	на основе современных парадигм и технологий	программного обеспечения на основе современных парадигм и технологий	разработку прикладного программного обеспечения на основе современных парадигм и технологий	на основе современных парадигм и технологий	
Применяет методы машинного обучения для решения прикладных задач анализа данных.					
Знать: - методы машинного обучения для решения прикладных задач анализа данных.	Фрагментарное представление о методах машинного обучения для решения прикладных задач анализа данных	Неполные представления о методах машинного обучения для решения прикладных задач анализа данных	Сформированные, но содержащие отдельные пробелы представления о методах машинного обучения для решения прикладных задач анализа данных	Сформированные систематические представления о методах машинного обучения для решения прикладных задач анализа данных	Вопросы для оценки знаний и умений, задания в виде расчетных задач, тестовые задания
Уметь: - применять методы машинного обучения для решения прикладных задач анализа данных.	Фрагментарное применение методов машинного обучения для решения прикладных задач анализа данных.	Несистематическое применение методов машинного обучения для решения прикладных задач анализа данных.	В целом успешное, но содержащее отдельные пробелы умение применять методы машинного обучения для решения прикладных задач анализа данных.	Сформированное применение методов машинного обучения для решения прикладных задач анализа данных.	Вопросы для оценки знаний и умений, задания в виде расчетных задач, тестовые задания

## 7.2 Задания или иные материалы, необходимые для оценки знаний и умений, характеризующих формирование компетенций в процессе освоения ОП ВО

*Вопросы для оценки знаний и умений, характеризующих формирование компетенций*

Шифр компетенции	Вопросы	Правильный ответ
<b>ПКН-2</b>	1. Какие числовые типы данных используются в Python?	int, float, и complex
	2. Какой оператор используется для конкатенации строк в языке Python, заданных по обе стороны от оператора?	+
	3. Как задается оператор повторения в языке Python, который объединяет несколько копий одной и той же строки?	*
	4. Может ли имя переменной в языке Python содержать пробелы?	нет
	5. Может ли имя переменной в языке Python содержать символ нижнего подчеркивания?	да
	6. Какая математическая функция в языке Python может быть использована для нахождения остатка от деления (в том числе для чисел с плавающей точкой)?	fmod()
	7. Какая математическая функция в языке Python может быть использована для нахождения значения $e^x$ ?	exp()
	8. Может ли быть индекс списка в языке Python отрицательным?	да
	9. Какую функцию нужно использовать, чтобы перевести строку, содержащую двоичный код, в целое число в языке Python?	int()
	10. Значение индекса первого элемента в кортеже в языке Python?	0
	11. Какая функция возвращает количество повторений элемента в кортеже в языке Python?	count()
	12. Вызов встроенной помощи в языке Python	help()
	13. Арифметический оператор Python, определяющий остаток от деления	%
	14. Оператор, который возвращает True если последовательность присутствует в объекте (в языке Python)	in
	15. Функция для сортировки списка в языке Python	sorted()
	16. Как определить длину строки str в языке Python?	len(str)
	17. Как создать пустой список в переменной s в языке Python?	s = []
	18. Как вставить на i-ый элемент списка list значение x в языке Python?	list.insert(i, x)
	19. Как называется цикл, который проверяет истинность некоторого условия, и если условие истинно, то выполняет инструкции цикла (в языке Python)?	while

	20. Какой оператор осуществляет выход из цикла в языке Python?	break
ПКН-4	1. Какую библиотеку необходимо использовать, для того чтобы получить доступ к модулю, написанному на Python из C	Python/C API
	2. Python – это императивный или декларативный язык?	Императивный
	3. В файле с каким расширением находятся директории для поиска модулей Python?	pth
	4. Как называется интегрированный набор компонентов и инструментов, который предоставляет готовое решение для решения определенной задачи в языке Python?	Фреймворк
	5. Как записывается в Python, пустой оператор?	pass
	6.С помощью какой функции можно посмотреть все методы и атрибуты, связанные с определенным объектом в Python?	dir()
	7.Как называется коллекция модулей или функций, предназначенных для использования другими приложениям? Данная коллекция содержит набор готовых решений для различных задач и обеспечивает функциональность, которую можно использовать в своём приложении.	Библиотека
	8 Как называется механизм, который позволяет создавать изолированные окружения для установки и использования пакетов Python?	Виртуальное окружение
	9. Как называется инструмент, который позволяет управлять установкой, обновлением и удалением библиотек и зависимостей в проектах на языке Python?	Менеджер пакетов
	10.Как называется приложение, если в нем каждый процесс имеет свой собственный набор ресурсов, включая память, открытые файлы, сетевые соединения и другие системные ресурсы?	Микропроцессорное
	11. Как называется приложение, если в нем несколько потоков выполняются в рамках одного процесса?	Многопоточное
	12.Какая встроенная функция возвращает словарь, содержащий все глобальные переменные, доступные в текущей области видимости в языке Python?	globals()
	13. Какая встроенная функция возвращает словарь, содержащий все локальные переменные, определенные в текущей области видимости в языке Python?	locals()
	14. Что формируется из исходного кода для ускорения работы интерпретатора Python?	байт-код

15. Как называется файл, содержащий код Python, который может быть повторно использован в других программах?	Модуль
16. Как называется директория, которая может содержать один или несколько модулей в Python?	Пакет
17. Как в Python называется способ извлечения определенной части последовательности (например, строки, списка, кортежа) с использованием индексации?	Слайс
18. Как называется код, который выполняется последовательно, один за другим, и блокирует выполнение других задач до его завершения?	Синхронный код
19. Как называется подход к написанию кода, который позволяет выполнять несколько задач одновременно в рамках одного процесса.	Асинхронный код
20. Как называются классы, которые определяют поведение других классов?	Метаклассы

### ***Практико-ориентированные задания***

Шифр компетенции	Практико-ориентированные задания	Правильный ответ
<b>ПКН-2</b>	1. Дан фрагмент программы в Python: <pre>for i in 1, 2, 3, 'one', 'two', 'three':     print(i)</pre> Какого типа значение примет переменная <i>i</i> при первых трех итерациях цикла?	int
	2. Дан фрагмент программы в Python: <pre>for i in 1, 2, 3, 'one', 'two', 'three':     print(i)</pre> Какого типа значение примет переменная <i>i</i> при четвертой, пятой и шестой итерациях цикла?	str
	3. Дан фрагмент программы в Python: <pre>A = 0 n = 5 for i in range(1, n + 1):     A += i print(A)</pre> Какая операция будет выполняться в цикле при нахождении значения переменной <i>A</i>	Сумма
	4. Дана условная инструкция if <i>Условие</i> : <i>Блок инструкций I</i> else:	Неполное ветвление

Блок инструкций 2		
Как называется условная инструкция <i>if</i> , если в ней отсутствует слово <i>else</i> и последующий за ней блок инструкций?		
5. Напишите оператор сравнения двух операндов на неравенство в языке Python	!=	
6. Какое значения получится, если преобразовать логическую переменную, имеющую значение <i>True</i> к типу <i>int</i> в языке Python?	1	
7. Какое значения получится, если преобразовать логическую переменную, имеющую значение <i>False</i> к типу <i>int</i> в языке Python?	0	
8. Напишите название функции, которая скрывается под знаком «?» ?(x) = 1, если x > 0, ?(x) = -1, если x < 0, ?(x) = 0, если x = 0.	sign	
9. Какого типа будет значение, если разделить два значения типа <i>int</i> (язык программирования Python)?	float	
10. Укажите символы строки <i>str</i> , начиная с индекса 5 и до конца (язык программирования Python)?	str[5:]	
11. Равны ли множества A = {1, 2, 3} и B = {3, 2, 3, 1}?	Да	
12. С помощью какой команды задается множество в языке программирования Python?	set()	
13. Что выведет программа <code>print(17 // 5)</code> , написанная на языке Python?	3	
14. Что выведет данный код, написанный на языке Python? <code>a = 'Пюре'</code> <code>print(a[::-1])</code>	ерюП	
15. Какое значение получит переменная <i>b</i> в результате работы кода в Python? <code>a, b = 0, 1</code>	1	
16. Что появится на экране в результате действия кода в Python? <code>a = 1</code> <code>b = 2</code> <code>a, b = b, a</code> <code>print(a, b)</code>	2 1	
17. Укажите последние два символа строки <i>str</i> (язык программирования Python)?	str[-2:]	
18. Написать операцию проверки на языке Python: содержится ли элемент <i>x</i> в списке <i>A</i> ?	<code>x in A</code>	
19. Результат работы программы на языке Python: <code>print([0, 1] * 3)</code>	[0, 1, 0, 1, 0, 1]	
20. Результат работы программы на языке Python: <code>a = ['red', 'green']</code>	red green	

	<code>print(' '.join(a))</code>	
<b>ПКН-4</b>	1. Результат работы программы на языке Python: <pre>def factorial(n):     res = 1     for i in range(1, n + 1):         res *= i     return res print(factorial(3))</pre>	6
	2. Результат работы программы на языке Python: <pre>def max(a, b):     if a &gt; b:         return a     else:         return b print(max(3, 5))</pre>	5
	3. Результат работы программы на языке Python: <pre>def max(a, b):     if a &gt; b:         return a     else:         return b  def max3(a, b, c):     return max(max(a, b), c)  print(max3(3, 5, 4))</pre>	5
	4. Результат работы программы на языке Python: <pre>def max(*a):     res = a[0]     for val in a[1:]:         if val &gt; res:             res = val     return res print(max(6, 5, 4))</pre>	6
	5. Результат работы программы на языке Python: <pre>def f():     global a     a = 1     print(a)  a = 0 f()</pre>	1
	6. Как называется прием, который используется в подпрограмме данной программы, написанной на языке Python:  <pre>def factorial(n):     if n == 0:         return 1</pre>	Рекурсия

	<pre> else:     return n * factorial(n - 1)  print(factorial(5)) </pre>	
	<p>7. Результат работы кода на языке Python:</p> <pre> sum = lambda a, b: a + b  print(sum(4, 5)) </pre>	9
	<p>8. Результат работы кода на языке Python:</p> <pre> a = [[1, 2, 3], [4, 5, 6]] b = a[0] print(b) </pre>	[1, 2, 3]
	<p>9. Результат работы кода на языке Python:</p> <pre> a = [[1, 2, 3, 4], [5, 6], [7, 8, 9]] s = 0 for i in range(len(a)):     for j in range(len(a[i])):         s += a[i][j] print(s) </pre>	45
	<p>10. Результат работы кода на языке Python:</p> <pre> a = [[1, 2, 3, 4], [5, 6], [7, 8, 9]] s = 0 for row in a:     for elem in row:         s += elem print(s) </pre>	45
	<p>11. Какими значениями будет заполнена первая строка двумерного массива в результате работы кода, написанного на языке Python?</p> <pre> n = 4 a = [[0] * n for i in range(n)] for i in range(n):     for j in range(n):         if i &lt; j:             a[i][j] = 0         elif i &gt; j:             a[i][j] = 2         else:             a[i][j] = 1 for row in a:     print(' '.join([str(elem) for elem in row])) </pre>	1 0 0 0
	<p>12. Результат работы кода на языке Python:</p> <pre> A = {1, 2, 3} print(1 in A, 4 in A) </pre>	True False
	<p>13. Результат работы кода на языке Python:</p> <pre> A = {1, 2, 3} print(5 in A, 4 in A) </pre>	False False
	<p>14. Результат работы кода на языке Python:</p> <pre> A = {1, 2, 3} B = {3, 2, 3, 1} print(A == B) </pre>	True

	15. Какая запись будет в четвертой строке, полученной таблицы (язык Python)? A = dict(zip('abcdef', list(range(6)))) for key in A: print(key, A[key])	d 3
	16. Результат работы программы на языке Python: res = 1 for i in range(1, 4): res *= i print(res)	6
	17. Результат работы программы на языке Python: a = [[1, 2, 3], [4, 5, 6]] print(a[1][2])	6
	18. Какая запись будет во второй строке, полученной таблицы (язык Python)? A = dict(zip('abcdef', list(range(6)))) for key, val in A.items(): print(key, val)	b 1
	19. Как подключить математическую библиотеку в языке Python?	import math
	20. Что появится на экране в результате действия кода, написанного на языке Python? s = 'ab12c59p7dq' digits = [] for symbol in s: if '1234567890'.find(symbol) != -1: digits.append(int(symbol)) print(digits)	[1, 2, 5, 9, 7]

### Тесты

Шифр компетенции	Тестовые задания	Правильный ответ
<b>ПKN-2</b>	1. Равны ли множества $A = \{1, 2, 3\}$ и $B = \{3, 2, 3, 1\}$ ?  1. Нет 2. Да 3. Второе множество больше первого 4. Первое множество больше второго	2
	2. Какие операции для целых чисел возвращают вещественное число (float) в языке Python?  1. +, - 2. //, * 3. +, * 4. **, /	4
	3. Какая математическая функция может быть использована для нахождения остатка от деления ( в том числе для чисел с плавающей точкой) в языке Python?	1



1. fmod() 2. factorial() 3. fabs() 4. frexp()	
4. Команда print (11 % 3), написанная на языке Python, выдаст результат:  1. 3 2. 2 3. 3.6666666666666665	2
5. Что покажет данный код (язык программирования Python)? for i in range(5): if i % 2 == 0: continue print(i)  1.ошибку 2.Числа 1, 3 и 5 3.Числа 2 и 4 4.Числа 1и 3	4
6. Чему будет равна переменная d после выполнения программы, написанной на языке Python? a = 10 b = 3 b = a // b d = a % (b - 2)  1.3 2.1 3.10 4. 0	4
7. Что выведет этот код, написанный на языке Python? a = 'Пюре' print(a[::-1])  1. е 2. ерюП 3. ерюп 4. П	2
8. Равны ли множества $A = \{1, 2, 3\}$ и $B = \{3, 2, 3, 1\}$ ?  1. Да 2. Второе множество больше первого 3 Первое множество больше второго	1
9. Какие ошибки допущены в коде, написанном на языке Python? def factorial(n): if n == 0: return 1 else:	4

<pre>return n * factorial(n - 1) print(factorial(5))</pre> <p>1. Функция не может вызывать сама себя  2. Необходимо указать тип возвращаемого значения  3. Функция всегда будет возвращать 1  4. В коде нет никаких ошибок</p>	
<p>10. Какое значение получит переменная <i>b</i> в результате работы кода, написанного на языке Python?  a, b = 0, 1</p> <p>1. 1  2. 0  3. 0,1</p>	1
<p>11. Значение индекса первого элемента в кортеже в языке Python?</p> <p>1. 0  2. 1  3. -1</p>	1
<p>12. Выберите оператор сравнения двух операндов на неравенство в языке Python:</p> <p>1. !=  2. &lt;&gt;  3. ≠</p>	1
<p>13. Какого типа будет значение, если разделить два значения типа <i>int</i> в языке Python?</p> <p>1. complex  2. int  3. float</p>	3
<p>14. Арифметический оператор Python, определяющий остаток от деления:</p> <p>1. %  2. //  3. /  4. ^</p>	1
<p>15. Арифметический оператор Python, определяющий деление нацело?</p> <p>1. %  2. //  3. /  4. ^</p>	2
<p>16. Результат работы программы, написанной на языке Python:</p> <pre>def factorial(n):     res = 1     for i in range(1, n + 1):         res *= i     return res</pre>	2

<pre>print(factorial(3))</pre> <p>1.9 2.6 3.24 4.120</p>	
<p>17. В результате работы программы, написанной на языке Python:</p> <pre>A = {1, 2, 3, 4, 5} B = {4, 5, 6, 7, 8}</pre> <pre>print(A   B)</pre> <p>на экране появится:</p> <p>1. {1, 2, 3, 4, 5, 6, 7, 8} 2. {4, 5} 3. {1, 2, 3}</p>	1
<p>18. В результате работы программы, написанной на языке Python:</p> <pre>A = {1, 2, 3, 4, 5} B = {4, 5, 6, 7, 8}</pre> <pre>print(A - B)</pre> <p>на экране появится:</p> <p>1. {1, 2, 3, 4, 5, 6, 7, 8} 2. {4, 5} 3. {1, 2, 3}</p>	3
<p>19. В результате работы программы, написанной на языке Python:</p> <pre>A = {1, 2, 3, 4, 5} B = {4, 5, 6, 7, 8}</pre> <pre>print(A &amp; B)</pre> <p>на экране появится:</p> <p>1. {1, 2, 3, 4, 5, 6, 7, 8} 2. {4, 5} 3. {1, 2, 3}</p>	2
<p>20. В результате работы программы, написанной на языке Python:</p> <pre>A = {1, 2, 3, 4, 5} B = {4, 5, 6, 7, 8}</pre> <pre>print(A ^ B)</pre> <p>на экране появится:</p>	1

	1. {1, 2, 3, 6, 7, 8} 2. {4, 5} 3. {1, 2, 3}	
<b>ПКН-4</b>	1. Сколько библиотек можно импортировать в один проект в языке Python?  1. Не более 5 2. Не более 3 3. Не более 10 4. Неограниченное количество	4
	2. Что покажет этот код, написанный на языке Python? for j in 'Hi! I\'m mister Robert': if j == '\': print("Найдено") break else: print ("Готово")  1 Ошибку в коде 2. "Найдено" и "Готово" 3. "Готово" 4. "Найдено"	4
	3. Какая функция выводит что-либо на консоль в языке Python?  1. out() 2. write() 3. print() 4. log()	3
	4. Какая библиотека отвечает за время в языке Python?  1. clock 2. time 3. localtime	2
	5. Как получить данные от пользователя в языке Python?  1. Использовать метод readLine() 2. Использовать метод input() 3. Использовать метод get() 4. Использовать метод read()	2
	6. Что будет показано в результате действия программы, написанной на языке Python? name = "John" print('Hi, %s' % name)  1. "Hi, name" 2. "Ошибка" 3. "Hi, John" 4."Hi, "	3
	7. Какая запись будет во второй строке, полученной таблицы	2

<p>в результате действия кода, написанного на языке Python?</p> <pre>A = dict(zip('abcdef', list(range(6)))) for key, val in A.items():     print(key, val)</pre> <p>1. a 2 2. b 1 3. a 1 4. b 2</p>	
<p>8 Какое ключевое слово применяется для определения функции в Python?</p> <p>1. func 2. def 3. procedure 4. define</p>	2
<p>9. Выберите правильные утверждения для языка Python?</p> <p>1. Метод – это параметр, применяемый к объекту 2. Метод – это функция, применяемая к объекту 3. Метод задается Имя_объекта.Имя_метода((параметры) 4. Метод задается Имя_метода.Имя_объекта ((параметры)</p>	2 и 3
<p>10. Какая может возникнуть ошибка при неправильной записи среза на языке Python?</p> <p>1. SyntaxError 2. NameError 3. IndexError 4. ValueError</p>	3
<p>11. Для чего применяется import в языке Python?</p> <p>1. Применяется с целью подключить пакет или модуль 2. Ключевое слово, целью которого изменение значения переменной за пределами её текущей области видимости 3. Функция предназначенная для добавления элемента в массив</p>	1
<p>12. Какие функции входят в библиотеку math в языке Python?</p> <p>1. int(), str(), input() 2. plot(), sell(), log() 3. factorial(), sqrt(), log() 4. fat(), cut(), car()</p>	3
<p>13. Выберите функцию, которая не относится к функциям библиотеки math в языке Python:</p> <p>1. radians()</p>	4

2. <code>sqrt()</code> 3. <code>exp()</code> 4. <code>len(x)</code>	
14. Переменные объявленные вне функции в языке Python, но доступные внутри неё называются...  1. Вещественные 2. Логические 3. Глобальные 4. Локальные	3
15. Какой метод возвращает индекс последнего вхождения данной подстроки в данной строке в языке Python?  1. <code>str.rfind()</code> 2. <code>str.replace()</code> 3. <code>str.find()</code>	1
16. Какой метод возвращает индекс первого вхождения данной подстроки в данной строке в языке Python?  1. <code>str.rfind()</code> 2. <code>str.replace()</code> 3. <code>str.find()</code>	3
17. Результат работы программы, написанной на языке Python:  <pre>a = [[1, 2, 3], [4, 5, 6]] print(a[1][2])</pre> 1. 3 2. 1 3. 6 4. 4	3
18. Результат работы кода, написанного на языке Python:  <pre>A = {1, 2, 3} B = {3, 2, 3, 1} print(A == B)</pre> 1. <i>True</i> 2. <i>False</i> 3. {1, 2, 3} 4. {3, 2, 3, 1}	1
19. Результат работы кода, написанного на языке Python:  <pre>A = {1, 2, 3} print(5 in A, 4 in A)</pre> 1. <i>False False</i> 2. <i>True False</i> 3. 1 2 3 4. 5 4	1

20. Как записывается в Python, пустой оператор?	2
1. break	
2. pass	
3. dir	
4. self	

### 7.3 Методические материалы, определяющие процедуры оценивания знаний и умений, характеризующих степень сформированности компетенций

Форма промежуточной аттестации по дисциплине – *зачет, экзамен.*

#### Критерии оценивания знаний и умений, характеризующих степень сформированности компетенций:

##### Критерии оценки знаний на зачете/экзамене

Оценка «**отлично**» выставляется обучающемуся (86 -100 баллов), который обладает всесторонними, систематизированными и глубокими знаниями материала учебной программы, умеет свободно выполнять задания, предусмотренные учебной программой, усвоил основную и ознакомился с дополнительной литературой, рекомендованной учебной программой. Как правило, оценка «отлично» выставляется обучающемуся усвоившему взаимосвязь основных положений и понятий дисциплины в их значении для приобретаемой специальности, проявившему творческие способности в понимании, изложении и использовании учебного материала, правильно обосновывающему принятые решения, владеющему разносторонними навыками и приемами выполнения практических работ.

Оценка «**хорошо**» выставляется обучающемуся (70 - 85 баллов), обнаружившему полное знание материала учебной программы, успешно выполняющему предусмотренные учебной программой задания, усвоившему материал основной литературы, рекомендованной учебной программой. Как правило, оценка «хорошо» выставляется обучающемуся, показавшему систематизированный характер знаний по дисциплине, способному к самостоятельному пополнению знаний в ходе дальнейшей учебной и профессиональной деятельности, правильно применяющему теоретические положения при решении практических вопросов и задач, владеющему необходимыми навыками и приемами выполнения практических работ.

Оценка «**удовлетворительно**» выставляется обучающемуся (50 – 69 баллов), который показал знание основного материала учебной программы в объеме, достаточном и необходимым для дальнейшей учебы и предстоящей работы по специальности, справился с выполнением заданий, предусмотренных учебной программой, знаком с основной литературой, рекомендованной учебной программой. Как правило, оценка «удовлетворительно» выставляется обучающемуся, допустившему погрешности в ответах на экзамене или выполнении экзаменационных заданий, но обладающему необходимыми знаниями под руководством преподавателя

для устранения этих погрешностей, нарушающему последовательность в изложении учебного материала и испытывающему затруднения при выполнении практических работ.

Оценка «**неудовлетворительно**» выставляется обучающемуся (менее 50 баллов), не знающему основной части материала учебной программы, допускающему принципиальные ошибки в выполнении предусмотренных учебной программой заданий, неуверенно с большими затруднениями выполняющему практические работы. Как правило, оценка «неудовлетворительно» выставляется обучающемуся, который не может продолжить обучение или приступить к деятельности по специальности по окончании университета без дополнительных занятий по соответствующей дисциплине.

#### **7.4 Типовые контрольные задания или иные материалы, необходимые для оценки индикаторов достижения компетенций, умений и знаний**

<b>Наименование компетенции</b>	<b>Наименование индикаторов достижения компетенции</b>	<b>Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции</b>	<b>Типовые контрольные задания</b>
<b>ПКН-2</b> Способен с помощью математической модели решать поставленную теоретическую или прикладную задачу, реализовывая алгоритм решения в виде программного модуля	1. Демонстрирует знание базовых математических моделей, применяемых в различных предметных областях	<b>Знать:</b> существующие базовые математические модели. алгоритмы решения теоретических или прикладных задач. <b>Уметь:</b> решать прикладные задачи в предметных областях с применением математических моделей.	<b>Задание 1</b> Напишите скрипт на Python обрабатывающий текстовый файл так, что в нем все заглавные буквы преобразуются в строчные.  <b>Задание 2</b> Написать программный код на Python, реализующий решение следующей задачи: В одномерном массиве, состоящем из n вещественных элементов, вычислить: - Сумму положительных элементов. - Произведение элементов, расположенных между максимальным и минимальным элементами. - Упорядочить элементы массива по возрастанию, используя алгоритм пузырька.



	2. Адаптирует и применяет существующие математические модели для решения поставленной прикладной или теоретической задачи.	<p><b>Знать:</b> математические модели для решения конкретных прикладных задач. подходы к применению математических моделей.</p> <p><b>Уметь:</b> адаптировать и применять математические модели для решения прикладных задач.</p>	<p><b>Задание 1</b> Используя программу Jupyter Notebook со- ставьте код на Python, определяющий существование треугольника по трем заданным дли- нам сторон. Если суще-ствует, то определить его тип. Используйте биб- лиотеку math.</p> <p><b>Задание 2</b> Выберите библиотекуPython для работы с мас-сивами. Выполните про- граммный код бинарногопоиска данных.</p>
	3. Владеет методологией математического моделирования для решения профессиональных задач.	<p><b>Знать:</b> современные системы математического моделирования.</p> <p><b>Уметь:</b> реализовывать алгоритмы решения профессиональных задач с применением современных систем математического моделирования.</p>	<p><b>Задание 1</b> Выберите библиотеку Python для хэширования данных. Выполните программный код быстрого поиска текстовых строк.</p> <p><b>Здание 2</b> Реализуйте программу на Python которой в качестве аргумента команд- ной строки передается имя CSV-файла, в первом столбце находятся числа,которые необходимо от- сортировать. Программа создает новый файл, в ко- тором первый столбец отсортирован.</p>
	ПКН -4 Способен проектировать и реализовывать прикладные программные системы в соответствии с анализом задачи	<p>1. Демонстрирует базовые знания о существующих математических методах и системах программирования.</p> <p><b>Знать:</b> Существующие математические методы; существующие системы программирования.</p> <p><b>Уметь:</b> выполнять выбор математических методов и систем программирования необходимый для решения конкретной задачи.</p>	<p><b>Задание 1</b> Выполните обработку текстового файла, в кото ром зафиксированы факты продаж, необходимого для подсчета суммарных продаж по раз- личным географическим подразделениям фирмы. На основе полученных данных составьте CSVфайл, хранящий полученные результаты.</p> <p><b>Задание 2</b> Опишите структуру CSVфайла фиксирующего факты продаж, необходимого для подсчета суммарных продаж по раз- личным географическим подразделениям фирмы</p>
	2. Использует и адаптирует	<b>Знать:</b> существующие	<b>Задание 1</b> Опишите закономерности в

	существующие математические методы и системы программирования для решения прикладных задач. Владеет навыками обеспечения	математические методы и системы программирования. <b>Уметь:</b> уметь адаптировать существующие математические методы и системы программирования для решения прикладных задач	синтаксических конструкциях Python для работы со списками и словарями. <b>Задание 2</b> Определите количество элементов списка, значения которых совпадают с ключами заданного словаря и количество элементов списка, значения которых равны значениям в словаре.
	3. Владеет навыками проектирования и разработки компонентов программного обеспечения на основе современных парадигм, технологий и языков программирования.	<b>Знать</b> Современные языки программирования, современные технологии программирования  <b>Уметь:</b> Выполнять разработку прикладного программного обеспечения на основе современных парадигм, технологий и языков программирования	<b>Задание 1</b> Опишите подходы, используемые в Python для группировки функций, ориентированных на обработку однотипных объектов и решение близких задач.  <b>Задание 2</b> Выполните классификацию с использованием бинарного дерева. Оцените количество операций
	4. Применяет методы машинного обучения для решения прикладных задач анализа данных.	<b>Знать:</b> методы машинного обучения для решения прикладных задач анализа данных. <b>Уметь:</b> применять методы машинного обучения для решения прикладных задач анализа данных.	<b>Задача 1</b> Сформируйте собственные суждения о применимости Python для решения задач машинного обучения.  <b>Задача 2</b> Выполните задачу построения бинарного дерева двумя способами: на основе списков и на основе узлов и ссылок. Оцените эффективность работы каждого способа

## 7.5 Теоретические вопросы для подготовки к экзаменам

*Примерные вопросы для подготовки к зачету в I семестре*

### Тема 1. Введение в программирование на Python

1. Присвоение по ссылке и по значению. Специфика создания объектов и присвоения в Python, особенности Python в связи с распространенностью использования неизменяемых типов.

2. Специфика типизации в языках программирования (различные аспекты типизации). Реализация типизации в Python.

### **Тема 2. Управляющие конструкции, списки и кортежи**

3. Организация ветвлений в Python. Инструкция if...else. Множественный выбор.

4. Циклы в Python, работа и устройство цикла for, типичное применение range и enumerate в цикле for.

5. Циклы в Python, работа и устройство цикла while. Организация бесконечного цикла, аварийное завершение цикла.

6. Списки в Python. Обращение к элементам списка и создание срезов. Обход списка и поиск элементов в списке.

7. Ключевые операции, проводящие к изменению списка и порождающие измененные списки, копирование списков.

### **Тема 3. Словари, множества и выражения-генераторы**

8. Словари в Python. Итерирование по словарям, преобразование между словарями и списками в Python. Операции с представлениями словарей.

9.

10. Операции со словарями, учитывающие возможное отсутствие ключа. Операции многоэлементного изменения словарей. Операции поэлементного извлечения из словаря и их использование.

11. Множества в Python. Основные способы создания, получения и изменения значений. Обход множеств. Выполнение основных операций с парой множеств в Python.

12. Кортежи в Python. Отличия кортежей от списков. Распаковка и частичная распаковка кортежей.

13. Выражения генераторы и генераторы списков в Python. Использование условий в генераторах.

14. Функции стандартной библиотеки для работы с контейнерами.

### **Тема 4. Функции**

15. Объявление и вызов функции в Python. Параметры функции со значением по умолчанию и комментирование функции.

16. Функция. Получение информации о функции. Способы передачи параметров при вызове функции.

17. Передача **переменного** количества параметров (именованных и не именованных) в функции Python.

18. Вызов функции с позиционными параметрами, находящимися в списке, и именованными параметрами, находящимися в словаре.

### **Тема 5. Работа с файлами и обработка исключительных ситуаций**

19. Синтаксис и семантика обработки исключительных ситуаций в Python.

20. Создание пользовательских исключений и инструкция assert.

21. Базовые операции для работы с файлами в Python. Использование инструкции with ... as на примере работы с файлами.

22. CSV файлы. Методы и функции работы с CSV файлами.

23. Бинарные файлы. Методы и функции работы с бинарными файлами.

## **Тема 6. Модули и пакеты**

24. Модули в Python и их отличие от скриптов Python. Варианты синтаксиса импорта модуля и объектов модуля.

25. Применение импортированных объектов. Порядок поиска модулей и специфика их загрузки. Загрузка модулей из глобального репозитория.

*Примерные вопросы для подготовки к экзамену во 2 семестре*

## **Тема 7-8. Объектно-ориентированное программирование**

1. Концепция класса и объекта. Принципы и механизмы ООП.

2. Объявление класса, конструктор, создание объектов и одиночное наследование в Python. Управление доступом к атрибутам класса в Python.

3. Полиморфизм и утиная типизация и проверка принадлежности объекта к классу в языке Python.

4. Методы классов и статические переменные и методы в Python. Специальные методы для использования пользовательских классов со стандартными операторами и функциями.

## **Тема 9-10. Функциональное программирование**

5. Основные возможности, поддерживаемые функциональными языками программирования. Поддержка элементов функционального программирования в Python.

6. Концепция «функции – граждане первого класса» в языке программирования, поддержка этой концепции в Python. Специфика лямбда-функций в Python их возможности и ограничения. Типичные сценарии использования лямбда-функций в Python.

7. Глобальные и локальные переменные в функциях на примере Python. Побочные эффекты вызова функций и их последствия.

8. Вложенные функции и замыкания, специфика реализации в Python.

9. Функции высшего порядка и декораторы в Python.

10. Концепция map/filter/reduce. Реализация map/filter/reduce в Python и пример их использования.

11. Итераторы в Python: встроенные итераторы, создание собственных итераторов, типичные способы обхода итераторов и принцип их работы. Встроенные функции для работы с итераторами и возможности модуля itertools.

12. Функции генераторы и выражения генераторы: создание и применение в Python.

## **Тема 11. Структуры данных: массивы, стеки, очереди, списки.**

13. Специфика массивов, как структур данных. Динамические массивы – специфика работы, сложность операций. Специфика работы с array в Python.

14. Абстрактная структура данных стек и очередь: базовые и расширенные операции, их сложность.

15. Специфика реализации и скорости основных операций в очереди на базе массива и связанного списка.

16. Связанные списки: однонаправленные и двунаправленные – принцип реализации. Сравнение скорости выполнения основных операций в связанных списках и в динамическом массиве.

17. Деревья, создание деревьев с помощью структуры список, узлов и ссылок. Операции с деревьями. Обход дерева

### **Тема 12. Алгоритмы поиска и сортировки**

18. Алгоритм обменной сортировки, сложность сортировки и возможности по ее улучшению.

19. Алгоритм сортировки выбором, сложность сортировки и возможности по ее улучшению.

20. Алгоритм сортировки вставками, его сложность. Алгоритм быстрого поиска в отсортированном массиве. Сложность поиска в отсортированном и не отсортированном массиве.

21. Алгоритм сортировки Шелла, сложность сортировки и возможности по ее улучшению.

22. Алгоритм быстрой сортировки, сложность сортировки и возможности по ее улучшению.

23. Алгоритм сортировки слиянием, сложность сортировки.

## Пример экзаменационных билетов

### ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ БЮДЖЕТНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

#### ФИНАНСОВЫЙ УНИВЕРСИТЕТ

#### ПРИ ПРАВИТЕЛЬСТВЕ РОССИЙСКОЙ ФЕДЕРАЦИИ

(Финансовый университет)

Краснодарский филиал	Кафедра «Математика и информатика»
Дисциплина «Алгоритмы и структуры данных в языке Python»	Форма обучения очная
Курс 1	Семестр 2
Направление 1.03.02 Прикладная математика и информатика	Профиль «Анализ данных и принятие решений в экономике и финансах»

### ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №

	Задания экзаменационного билета	Балы БРС
	Теоретические вопросы:	
1	Алгоритм быстрой сортировки, сложность сортировки и возможности по ее улучшению.	20
2	Концепция класса и объекта. Принципы и механизмы ООП	20
	Практическое задание	
3	Дана матрица $A[3 \times 3]$ . Найти максимум из минимальных элементов столбцов матрицы. Вывести значение.	20
	<b>ИТОГО</b>	<b>60</b>

Подготовил:

Н.Г. Пьянкова

На основе перечня теоретических вопросов и практико-ориентированных заданий, утвержденного на заседании кафедры (протокол № \_\_\_ от \_\_\_\_\_ 2022г.).

Утверждаю:

Заведующий кафедрой

А.С.Молчан

«\_\_\_\_\_» \_\_\_\_\_ 2022 г.

## **8.Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины**

### **Основная литература**

1. Баяк, О. А. Практикум по анализу данных на языках Python и R : учебное пособие / О. А. Баяк, М. Р. Исаева, М. О. Самсонкин. - Москва : Прометей, 2023. - 100 с. - ISBN 978-5-00172-356-1. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/2124862>
2. Жуков, Р. А. Язык программирования Python: практикум : учебное пособие / Р.А. Жуков. — Москва : ИНФРА-М, 2023. — 216 с. + Доп. материалы [Электронный ресурс]. — (Высшее образование: Бакалавриат). — DOI 10.12737/textbook\_5cb5ca35aaa7f5.89424805. - ISBN 978-5-16-016971-2. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1915716> (дата обращения: 15.08.2023).

### **Дополнительная литература**

3. Гуриков, С. Р. Основы алгоритмизации и программирования на Python : учебное пособие / С.Р. Гуриков. — Москва : ИНФРА-М, 2023. — 343 с. — (Среднее профессиональное образование). - ISBN 978-5-16-016906-4. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/1927269>
4. Протодьяконов, А. В. Алгоритмы Data Science и их практическая реализация на Python : учебное пособие / А. В. Протодьяконов, П. А. Пылов, В. Е. Садовников. - Москва ; Вологда : Инфра-Инженерия, 2022. - 392 с. - ISBN 978-5-9729-1006-9. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1902689>

## **9.Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины**

1. Pylru 1.0.9 [Электронный ресурс]: сайт. - Режим доступа: <https://pypi.python.org/pypi/pylru>
2. Python Data Analysis Library [Электронный ресурс]: сайт. - Режим доступа: <http://pandas.pydata.org/>
3. Python Documentation [Электронный ресурс]: сайт. - Режим доступа: <http://python.org/doc/>
4. Python Standard Library [Электронный ресурс]: сайт. - Режим доступа: <https://docs.python.org/2/library/>
5. Scikit-learn Machine Learning in Python [Электронный ресурс]: сайт. - Режим доступа: <http://scikit-learn.org>
6. Официальный сайт продукта <https://www.python.org/>
7. Портал Финансового университета <http://www.fa.ru/>
8. Каталог курсов Интернет Университета Информационных Технологий <http://www.intuit.ru/>
10. The Python Tutorial // <https://docs.python.org/3/tutorial/index.html>.
11. The Python Standard Library //

- <https://docs.python.org/3/library/index.html>.
12. SciPy // <http://docs.scipy.org/doc/scipy/reference/>
  13. NumPy User Guide // <http://docs.scipy.org/doc/numpy/user/index.html>
  14. Электронная библиотека Финансового университета (ЭБ) <http://elib.fa.ru/>
  15. Электронно-библиотечная система BOOK.RU <http://www.book.ru>
  16. Электронно-библиотечная система «Университетская библиотека ОНЛАЙН» <http://biblioclub.ru/>
  17. Электронно-библиотечная система Znanium <http://www.znaniy.com>
  18. Электронно-библиотечная система издательства «ЮРАЙТ» <https://urait.ru/>
  19. Электронно-библиотечная система издательства Проспект <http://ebs.prospekt.org/books>
  20. Электронно-библиотечная система издательства «Лань» <https://e.lanbook.com/>
  21. Электронная библиотека Издательского дома «Гребенников» <https://grebennikon.ru/>
  22. Деловая онлайн-библиотека Alpina Digital <http://lib.alpinadigital.ru/>
  23. Научная электронная библиотека eLibrary.ru <http://elibrary.ru>
  24. Национальная электронная библиотека nlr.ru
  25. Финансовая справочная система «Финансовый директор» <http://www.1fd.ru/>
  26. Ресурсы информационно-аналитического агентства по финансовым рынкам Cbonds.ru <https://cbonds.ru/>
  27. СПАРК <https://spark-interfax.ru/>
  28. Academic Reference <http://ar.cnki.net/ACADREF>
  29. Bank Focus <http://library.fa.ru/resource.asp?id=527>
  30. Пакет баз данных компании EBSCO Publishing, крупнейшего агрегатора научных ресурсов ведущих издательств мира <http://search.ebscohost.com>
  31. Электронные продукты издательства Elsevier <http://www.sciencedirect.com>
  32. Emerald: Management eJournal Portfolio <https://www.emerald.com/insight/>
  33. Информационно-аналитическая база данных EMIS Global <https://www.emis.com/php/companies/overview/index>
  34. Реферативная база данных по математике MathSciNET <https://mathscinet.ams.org/mathscinet/>
  35. Oxford Scholarship Online <https://oxford.universitypressscholarship.com/>
  36. Коллекция научных журналов Oxford University Press <https://academic.oup.com/journals/>
  37. ProQuest: База данных Business Ebook Subscription на платформе Ebook Central <https://search.proquest.com/>
  38. ProQuest Dissertations & Theses A&I <https://search.proquest.com/>
  39. База данных RUSLANA компании Bureau van Dijk <https://ruslana.bvdep.com/>
  40. Scopus <https://www.scopus.com>
  41. Электронная коллекция книг издательства Springer: Springer eBooks <http://link.springer.com/>
  42. Интерактивная финансовая информационная система компании Bloomberg
  43. Система Thomson Reuters Eikon
  44. Web of Science <http://apps.webofknowledge.com>



## 10. Методические указания для обучающихся по освоению дисциплины

Лекционные занятия проводятся в соответствии с тематическим планом, при изложении материала рекомендуется использовать презентации в среде PowerPoint программный код из Jupyter Notebook и фрагменты печатных материалов по теме лекции.

В ходе интерактивных занятий следует проводить разбор конкретных примеров программного кода из Jupyter Notebook.

При изучении теоретического материала необходимо опираться на рабочую программу дисциплины и литературу из основного списка.

Кроме этого материала необходимо активно работать с Интернет-источниками и пособиями других авторов, помогающими усвоить материал отдельных разделов программы.

Практические занятия проводятся по следующей схеме:

Студенты в индивидуальном режиме выполняют задания в компьютерном классе. Существуют две основных формы заданий. В первом случае всем студентам предлагаются одинаковые исходные данные. После выполнения заданий устраивается обсуждение результатов в форме тематической дискуссии. Во втором случае каждому из студентов предлагается индивидуальное задание. В этом случае после выполнения задания организуется работа в группах: каждый из студентов анализирует и критически оценивает или же интерпретирует результаты одного или нескольких членов группы.

Контрольная работа выполняется студентами индивидуально и самостоятельно во внеаудиторное время. Каждый из студентов выбирает область экономических знаний, для которой можно найти данные для анализа в свободных источниках, выдвигает предположения о наличии скрытых закономерностей в отобранных данных, определяет вид этих закономерностей, выбирает соответствующие инструментальные средства интеллектуального анализа данных, осуществляет поиск закономерностей и делает выводы по возможности применения методов интеллектуального анализа данных в избранной области.

Тематика контрольной работы соответствует содержанию дисциплины, и определяется преподавателем.

Общие положения по выполнению контрольной работы

Контрольная работа по дисциплине «Алгоритмы и структуры данных в языке Python» предназначена для проверки степени усвоения студентами пройденного материала по данной дисциплине.

Контрольная работа выполняется после прослушивания студентами лекционного курса по дисциплине и их самостоятельной работы с рекомендованной преподавателем учебной литературой. Контрольная работа должна показать, что ее автор освоил фундаментальные знания в области программирования на языке Python.

Контрольная работа носит практический характер и дает дополнительный опыт самостоятельной работы над выбранной темой, заключающийся в подборе

необходимой литературы, письменном изложении материала на основе систематизации, обобщении, критическом анализе изученного материала и умении решения практических задач.

Контрольная работа должна включать следующие структурные элементы:

введение;

задачи, решение которых требует написание программного кода на языке Python

заключение;

список использованных источников.

Во введении обосновывается актуальность темы работы, определяются цели и задачи.

Решение задач, требующих финансовых вычислений, должно содержать:

условие задания;

перечень вводимых условных обозначений;

применяемые для расчета формулы;

программный код .на языке Python

В заключении делаются основные выводы по изученной теме.

Список информационных источников должен содержать не менее 5-7 наименований.

В методических указаниях по дисциплине «Алгоритмы и структуры данных в языке Python» представлены 10 вариантов контрольных работ. Номер варианта контрольной работы устанавливается в соответствии с последней цифрой номера зачетной книжки.

Выполненная и оформленная в соответствии с требованиями кафедры «Математика и информатика» контрольная работа прикрепляется в информационно-образовательном портале для проверки преподавателем.

## **11. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень необходимого программного обеспечения и информационных справочных систем**

### **11.1 Комплект лицензионного программного обеспечения:**

1. ОС Astra Linux,
2. LibreOffice
3. Антивирус Kaspersky

### **11.2. Современные профессиональные базы данных и информационные справочные системы**

1. Информационно-образовательный портал Финуниверситета. <http://portal.ufrf.ru>
2. Электронная энциклопедия <http://ru.wikipedia.org/wiki/Wiki>

3. Система комплексного раскрытия информации «СКРИН»  
<http://www.skrin.ru/>
4. Электронно-библиотечная система BOOK.RU <http://www.book.ru>
5. Электронно-библиотечная система «Университетская библиотека ОНЛАЙН»  
<http://biblioclub.ru/>
6. Электронно-библиотечная система Znanium <http://www.znaniy.com>
7. Электронно-библиотечная система «Лань» <http://e.lanbook.com>
8. Электронно-библиотечная система «Юрайт» [www.biblio-online.ru](http://www.biblio-online.ru)
9. ЭБС «Единое окно доступа к образовательным ресурсам»  
<http://window.edu.ru>

### **11.3 Сертифицированные программные и аппаратные средства защиты информации**

Не предусмотрены

Каждый обучающийся в течение всего обучения обеспечивается индивидуальным неограниченным доступом к электронно-библиотечной системе и электронной информационно-образовательной среде Краснодарского филиала Финансового университета.

Электронная информационно-образовательная среда Финансового университета обеспечивает:

– доступ к учебным планам, рабочим программам дисциплин, практик, к указаниям электронных библиотечных систем и электронным образовательным ресурсам, указанным в рабочих программах. По дисциплине «Корпоративные информационные системы на базе 1С Предприятие» в ИОП представлены следующие виды информационных ресурсов:

- аннотации дисциплины;
- видеолекции;
- мультимедийные презентации по всем темам курса; - методический материал:

- рабочие программы дисциплины.

Источник - <https://portal.fa.ru/Catalog?MenuId=Catalog>

- фиксацию хода образовательного процесса, результатов промежуточной аттестации;

– проведение всех видов занятий, процедур оценки результатов обучения, реализация которых осуществляется с применением электронного обучения.

## **12. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине**

Материально-техническая база Краснодарского филиала Финансового университета соответствует действующим противопожарным правилам и нормам, обеспечивает проведение всех видов дисциплинарной и междисциплинарной подготовки, практической и научно-исследовательской работ обучающихся, предусмотренных учебным планом.

Образовательный процесс обеспечивается специальными помещениями, которые представляют собой аудитории для проведения занятий лекционного типа, занятий семинарского типа, лабораторных работ, выполнения курсовых работ, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, помещения для самостоятельной работы студентов и помещения для хранения и профилактического обслуживания учебного оборудования.

Специальные помещения укомплектованы специализированной мебелью и техническими средствами обучения, обеспечивающими представление учебной информации большой аудитории, демонстрационным оборудованием.

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой, обеспечивающей доступ к сети Интернет и электронной информационно-образовательной среде Краснодарского филиала Финансового университета.

При изучении *теоретического* материала необходимо опираться на рабочую программу дисциплины и литературу из основного списка.

Кроме этого материала необходимо активно работать с Интернет-источниками и пособиями других авторов, помогающими усвоить материал отдельных разделов программы.