

Федеральное государственное образовательное  
бюджетное учреждение высшего образования  
**«ФИНАНСОВЫЙ УНИВЕРСИТЕТ  
ПРИ ПРАВИТЕЛЬСТВЕ РОССИЙСКОЙ ФЕДЕРАЦИИ»**  
(Финансовый университет)

**Краснодарский филиал Финуниверситета**

Кафедра «Математика и информатика»

СОГЛАСОВАНО

ООО «Портал-Юг»  
Генеральный директор



Е.В. Мостовой

«21» февраля 2024 г.

УТВЕРЖДАЮ

Краснодарский филиал  
Финуниверситета

Директор



Э.В. Соболев

«21» февраля 2024 г.

Пьянкова Н.Г.

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ  
АЛГОРИТМЫ И СТРУКТУРЫ ДАННЫХ В ЯЗЫКЕ PYTHON**

для студентов, обучающихся по направлению подготовки

38.03.05 — Бизнес-информатика, ОП «Цифровая трансформация управления  
бизнесом», профиль «Бизнес-аналитика» (очно-заочная форма обучения)

*Рекомендовано Ученым советом Краснодарского филиала Финуниверситета  
(протокол № 12 от 20.02.2024)*

*Одобрено кафедрой «Математика и информатика»  
(протокол № 13 от 27.02.2024)*

**Краснодар 2024**

УДК 004.043 (075.8)  
ББК 32.973.1-018я73  
П96

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

Пьянкова Н.Г. «Алгоритмы и структуры данных в языке Python». Рабочая программа дисциплины для студентов, обучающихся по направлению подготовки 38.03.05 «Бизнес-информатика» – Краснодар: Краснодарский филиал Финуниверситета, кафедра «Математика и информатика», 2024.

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.

*Учебное издание*

**Пьянкова Н.Г.**

**АЛГОРИТМЫ И СТРУКТУРЫ ДАННЫХ В ЯЗЫКЕ PYTHON**

**Рабочая программа дисциплины**

*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.



## Содержание

1. Наименование дисциплины .....	5
2. Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине .....	5
3. Место дисциплины в структуре образовательной программы .....	6
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
5.3. Содержание семинаров, практических занятий .....	11
6. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине .....	14
6.1. Перечень вопросов, отводимых на самостоятельное освоение дисциплины, формы внеаудиторной самостоятельной работы .....	14
6.2. Перечень вопросов, заданий, тем для подготовки к текущему контролю .....	15
7. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине .....	17
8. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины .....	29
9. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины .....	30
10. Методические указания для обучающихся по освоению дисциплины .....	31
11.1 Комплект лицензионного программного обеспечения: .....	33
11.2. Современные профессиональные базы данных и информационные справочные системы .....	33
11.3 Сертифицированные программные и аппаратные средства защиты информации .....	34
12. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине .....	34

## 1. Наименование дисциплины

Дисциплина Б1.О.02.07 «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: УК-4, УК-10.

Код компетенции	Наименование компетенции	Индикаторы достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикатором достижения компетенции
УК-4	Способность использовать прикладное программное обеспечение при решении профессиональных задач	1. Использует основные методы и средства получения, представления, хранения и обработки данных.	<u>Знать</u> основные методы получения, представления, хранения и обработки данных на языке программирования Python. <u>Уметь</u> применять основные методы получения, представления, хранения и обработки данных на практике, используя средства и методы языка Python.
		2. Демонстрирует владение профессиональными пакетами прикладных программ.	<u>Знать</u> перечень и функции основных пакетов средств разработки программ. <u>Уметь</u> выполнять разработку и отладку прикладных программ, используя интегрированную среду разработки.
		3. Выбирает необходимо прикладное программное обеспечение в зависимости от решаемой задачи.	<u>Знать</u> перечень и функции необходимого прикладного ПО. <u>Уметь</u> выбирать необходимо прикладное ПО в зависимости от решаемой задачи.
		4. Использует прикладное программное обеспечение для решения конкретных прикладных задач.	<u>Знать</u> алгоритмы решения прикладных задач <u>Уметь</u> реализовать алгоритмы решения прикладных задач на прикладном ПО.
УК-10	Способность осуществлять поиск, критически анализировать, обобщать и систематизировать информацию, использовать системный подход для решения	1. Четко описывает состав и структуру требуемых данных и информации, грамотно реализует процессы их сбора, обработки и интерпретации.	<u>Знать</u> состав и структуру требуемых данных и информации языка Python <u>Уметь</u> грамотно реализовать процессы их сбора, обработки и интерпретации, используя средства и методы языка Python
		2. Обосновывает сущность происходящего, выявляет закономерности, понимает природу вариабельности.	<u>Знать</u> природу вариабельности. <u>Уметь</u> обосновывать сущность происходящего, выявить закономерности.
		3. Формулирует признак	<u>Знать</u> признаки классификации

	поставленных задач	классификации, выделяет соответствующие ему группы однородных «объектов», идентифицирует общие свойства элементов этих групп, оценивает полноту результатов классификации, показывает прикладное назначение классификационных групп.	<u>Уметь</u> выделять соответствующие ему группы однородных «объектов», идентифицировать общие свойства элементов этих групп, оценивать полноту результатов классификации, показывать прикладное назначение классификационных групп.
		4. Грамотно, логично, аргументировано формирует собственные суждения и оценки. Отличает факты от мнений, интерпретаций, оценок и т.д. в рассуждениях других участников деятельности.	<u>Знать</u> отличия фактов от мнений, интерпретаций и оценок <u>Уметь</u> формировать собственные суждения и оценки, отличать факты от мнений, интерпретаций, оценок.
		5. Аргументировано и логично представляет свою точку зрения посредством и на основе системного описания.	<u>Знать</u> основные принципы системного описания <u>Уметь</u> представить свою точку зрения посредством и системного описания.

### 3. Место дисциплины в структуре образовательной программы

Дисциплина «Алгоритмы и структуры данных в языке Python» относится к циклу математики и информатики обязательной части по направлению подготовки: 38.03.05 Бизнес-информатика, ОП «Цифровая трансформация управления бизнесом» (очно-заочная форма обучения).

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

Вид учебной работы по дисциплине	Всего (в з/ед. и часах)	Семестр 2 (в часах)
Общая трудоемкость дисциплины	6 зач.ед. / 216 час.	6/216 час.
Контактная работа - Аудиторные занятия	38	38
Лекции	12	12

Семинары, практические занятия	26	26
Самостоятельная работа	178	178
В семестре	142	142
Контроль	36	36
Вид промежуточной аттестации	зачет, экзамен	экзамен
Вид текущего контроля	Контрольные работы	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**

№ п/п	Наименование тем (разделов) дисциплины	Трудоемкость в часах					Формы текущего контроля успеваемости
		Всего	Аудиторная работа			Самостоятель ная работа	
			Общая, в т.ч.:	Лекци и	Семинары, практическ		

					не занятия		
1.	Введение в программирование на Python	18	4	1	3	19	Самостоятельное решение задач (программирование), выступление на семинаре
2.	Управляющие конструкции, списки и кортежи	18	4	1	3	19	Самостоятельное решение задач (программирование), выступление на семинаре
3.	Словари, множества и выражения-генераторы	18	3	1	2	14	Самостоятельное решение задач (программирование) выступление на семинаре
4.	Функции	18	3	1	2	14	Самостоятельное решение задач (программирование), выступление на семинаре
5.	Работа с файлами и обработка исключительных ситуаций	18	3	1	2	14	Самостоятельное решение задач (программирование), выступление на семинаре
6.	Модули и пакеты	18	3	1	2	14	Самостоятельное решение задач (программирование), выступление на семинаре
7.	Введение в объектно-ориентированное программирование	18	3	1	2	14	Самостоятельное решение задач (программирование), выступление на семинаре
8.	Объектно-ориентированное программирование в Python	18	3	1	2	14	Самостоятельное решение задач (программирование), выступление на семинаре
9.	Введение в функциональное программирование	18	3	1	2	14	Самостоятельное решение задач (программирование), выступление на семинаре
10.	Функциональное программирование	18	3	1	2	14	Самостоятельное решение задач (программирование),

	ние в Python						выступление на семинаре
11.	Структуры данных: массивы, стеки, очереди, списки	18	3	1	2	14	Самостоятельное решение задач (программирование), выступление на семинаре
12.	Алгоритмы поиска и сортировки	18	3	1	2	14	Самостоятельное решение задач (программирование), выступление на семинаре
Контроль		36				36	
Итого за 2 семестр		216	38	12	26	178	Контрольная работа
В целом по дисциплине		216	38	12	26	178	Контрольные работы
Итого в %		100	22	9	13	56	

### 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
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.	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий (не менее 30% времени на интерактивные технологии)

	Рекомендуемая литература: 8 (1, 2); 9 (1-4)	
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	<p>Устройство модулей и пакетов, инструкции <code>import</code> и <code>from</code>. Создание собственных модулей и пакетов.</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)
7. Введение в объектно-ориентированное программирование	<p>Базовые возможности ООП в Python: создание классов и объектов; наследование и полиморфизм; функция <code>super()</code>.</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 <code>map</code>, <code>filter</code>, <code>reduce</code> в Python. Итераторы в Python, итерируемый тип данных. Модуль <code>itertools</code>.</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,</p>	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий (не менее 30% времени на интерактивные технологии)

	<p>варианты связанных списков.</p> <p>Рекомендуемая литература: 8 (1, 2, 3, 4);9 (1-10)</p>	
12 Алгоритмы поиска и сортировки	<p>Реализация на Python простых методов сортировки: обменные сортировки (с различными вариациями); сортировка выбором (извлечением); сортировка включением (вставками).</p> <p>Реализация на Python эффективных методов сортировки: быстрая сортировка; сортировка Шелла; сортировка слиянием.</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)

## 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 контрольным работам

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.
1. Организация и пример цикла for в Python.
2. Операции над словарями в Python.
3. Выражения-генераторы для списков в Python.
4. Необязательные параметры функций в Python.
5. Заpackовка и распаковка параметров в Python.
6. Аннотации и документирование функций.
7. Создание объектов в Python.



8. Управление доступом к атрибутам класса в Python.
9. Выполнение интроспекции в Python.
10. Замыкания в Python.
11. Использование декораторов в Python.
12. Создание собственных декораторов в Python.
13. Компилятор и интерпретатор. Достоинства и недостатки.
14. Назовите и дайте краткую характеристику основных классов языков программирования.
15. Встроенные числовые типы языка Python.
16. Списки. Создание, основные операции.
17. Основные методы списка.
18. Кортежи. Создание, основные методы и операции.
19. Словари. Создание, основные операции.
20. Методы для работы со словарями.
21. Множества. Создание, основные методы и операции.
22. Переменные. Правила именования переменных.
23. Динамическая типизация.
24. Операторы сравнения и логические операторы.
25. Инструкция if...else.
26. Инструкция цикла while.
27. Инструкция цикла for.
28. Создание и вызов функции.
29. Передача аргументов функцию.
30. Функции-генераторы.
31. Лямбда-функции.
32. Модули. Инструкции import и from.
33. Базовые принципы объектно-ориентированного программирования.
34. Класс, метод класса, атрибут класса. Определение класса и создание экземпляра класса.
35. Конструктор и деструктор.
36. Наследование.
37. Абстрактные методы класса.
38. Статические методы класса.
39. Свойства класса.
40. Исключения. Обработка исключений.
41. Пользовательские исключения.

### **Примеры заданий контрольных работ**

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. Создать иерархию классов для фруктов, продающихся в магазине. Иерархия должна содержать не менее 3 классов. Объекты должны содержать не менее

2х атрибутов и 2х 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.

3. 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 'Eeny,meeny,miney,moe'

4. Реализовать декоратор с именем `not_none`, который генерирует исключительную ситуацию если декорируемая функция вернула значения `None`. (20 баллов)

5. 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 итоговую словарь. Пример: {'a':1, 'b':3, 'c':4, 'd':3} -> {1:'a', 4:'c'}

6. Реализовать декоратор с именем `print_type`, выводящий на печать тип значения, возвращаемого декорируемой функцией.

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

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
	«неудовлетвори тельно»	«удовлетвори тельно»	«хорошо»	«отлично»	
<b>УК-4 Способность использовать прикладное программное обеспечение при решении профессиональных задач</b>					
Использует основные методы и средства получения, представления, хранения и обработки данных					
Знать: -	Фрагментарно е	Неполные представлени	Сформирован ные, но	Сформирован ные	Вопросы для оценки знаний

Планируемые результаты освоения компетенции (индикатора достижения компетенции)	Уровень освоения				Оценочное средство
	«неудовлетворительно»	«удовлетворительно»	«хорошо»	«отлично»	
существующие базовые математические модели. - алгоритмы решения теоретических или прикладных задач.	представление об основных методах получения, представления, хранения и обработки данных на языке программирования Python	об основных методах получения, представления, хранения и обработки данных на языке программирования Python	содержащие отдельные пробелы представления об основных методах получения, представления, хранения и обработки данных на языке программирования Python	систематические представление об основных методах получения, представления, хранения и обработки данных на языке программирования Python	и умений, задания в виде расчетных задач, тестовые задания
Уметь: решать прикладные задачи в предметных областях с применением математических моделей.	Фрагментарное умение применять основные методы получения, представления, хранения и обработки данных на практике, используя средства и методы языка Python	Несистематическое умение применять основные методы получения, представления, хранения и обработки данных на практике, используя средства и методы языка Python	В целом успешное, но содержащее отдельные пробелы умение применять основные методы получения, представления, хранения и обработки данных на практике, используя средства и методы языка Python	Сформированное умение применять основные методы получения, представления, хранения и обработки данных на практике, используя средства и методы языка Python	Вопросы для оценки знаний и умений, задания в виде расчетных задач, тестовые задания
<b>Демонстрирует владение профессиональными пакетами прикладных программ</b>					
Знать: - перечень и функции основных пакетов средств разработки программ	Фрагментарное представление о перечне и функциях основных пакетов средств разработки программ	Неполные представления о перечне и функциях основных пакетов средств разработки программ	Сформированные, но содержащие отдельные пробелы представления о перечне и функциях основных пакетов	Сформированные систематические представления о перечне и функциях основных пакетов средств	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»	
			средств разработки программ	разработки программ	
Уметь: выполнять разработку и отладку прикладных программ, используя интегрированная среда разработки	Фрагментарное умение выполнять разработку и отладку прикладных программ, используя интегрированная среда разработки	Несистематическое применение умений выполнять разработку и отладку прикладных программ, используя интегрированная среда разработки	В целом успешное, но содержащее отдельные пробелы умение выполнять разработку и отладку прикладных программ, используя интегрированная среда разработки	Сформированное умение выполнять разработку и отладку прикладных программ, используя интегрированная среда разработки	Вопросы для оценки знаний и умений, задания в виде расчетных задач, тестовые задания
Выбирает необходимо прикладное программное обеспечение в зависимости от решаемой задачи					
Знать: - перечень и функции необходимого прикладного ПО	Фрагментарное представление о перечне и функциях необходимого прикладного ПО	Неполные представления о перечне и функциях необходимого прикладного ПО	Сформированные, но содержащие отдельные пробелы представления о перечне и функциях необходимого прикладного ПО	Сформированные систематические представления о перечне и функциях необходимого прикладного ПО	Вопросы для оценки знаний и умений, задания в виде расчетных задач, тестовые задания
Уметь: - выбирать необходимо прикладное ПО в зависимости от решаемой задачи	Фрагментарное умение выбирать необходимо прикладное ПО в зависимости от решаемой задачи	Несистематическое применение умений выбирать необходимо прикладное ПО в зависимости от решаемой задачи	В целом успешное, но содержащее отдельные пробелы умение выбирать необходимо прикладное ПО в зависимости от решаемой задачи	Сформированное умение выбирать необходимо прикладное ПО в зависимости от решаемой задачи	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»	
Выбирает прикладное программное обеспечение для решения конкретных прикладных задач					
Знать: - алгоритмы решения прикладных задач	Фрагментарное представление об алгоритмах решения прикладных задач	Неполные представления об алгоритмах решения прикладных задач	Сформированные, но содержащие отдельные пробелы представления об алгоритмах решения прикладных задач	Сформированные систематические представления об алгоритмах решения прикладных задач	Вопросы для оценки знаний и умений, задания в виде расчетных задач, тестовые задания
Уметь: - реализовать алгоритмы решения прикладных задач на прикладном ПО	Фрагментарное умение реализовать алгоритмы решения прикладных задач на прикладном ПО	Несистематическое применение умений реализовать алгоритмы решения прикладных задач на прикладном ПО	В целом успешное, но содержащее отдельные пробелы умение реализовать алгоритмы решения прикладных задач на прикладном ПО	Сформированное умение реализовать алгоритмы решения прикладных задач на прикладном ПО	Вопросы для оценки знаний и умений, задания в виде расчетных задач, тестовые задания
УК-10 Способность осуществлять поиск, критически анализировать, обобщать и систематизировать информацию, использовать системный подход для решения поставленных задач					
Четко описывает состав и структуру требуемых данных и информации, грамотно реализует процессы их сбора, обработки и интерпретации					
Знать: - состав и структуру требуемых данных и информации языка Python	Фрагментарное представление о составе и структуре требуемых данных и информации языка Python	Неполные представления о составе и структуре требуемых данных и информации языка Python	Сформированные, но содержащие отдельные пробелы представления о составе и структуре требуемых данных и информации языка Python	Сформированные систематические представления о составе и структуре требуемых данных и информации языка Python	Вопросы для оценки знаний и умений, задания в виде расчетных задач, тестовые задания
Уметь: - грамотно	Фрагментарное умение	Несистематическое	В целом успешное, но	Сформированное умение	Вопросы для оценки знаний

Планируемые результаты освоения компетенции (индикатора достижения компетенции)	Уровень освоения				Оценочное средство
	«неудовлетворительно»	«удовлетворительно»	«хорошо»	«отлично»	
реализовать процессы их сбора, обработки и интерпретации, используя средства и методы языка Python	реализовать процессы их сбора, обработки и интерпретации, используя средства и методы языка Python	применение реализовать процессы их сбора, обработки и интерпретации, используя средства и методы языка Python	содержащее отдельные пробелы умение реализовать процессы их сбора, обработки и интерпретации, используя средства и методы языка Python	грамотно реализовать процессы их сбора, обработки и интерпретации, используя средства и методы языка Python	и умений, задания в виде расчетных задач, тестовые задания
Обосновывает сущность происходящего, выявляет закономерности, понимает природу variability					
Знать: - природу variability	Фрагментарное представление о природе variability	Неполное представление о природе variability	Сформированные, но содержащие отдельные пробелы представления о природе variability	Сформированные систематические представления о природе variability	Вопросы для оценки знаний и умений, задания в виде расчетных задач, тестовые задания
Уметь: - обосновывать сущность происходящего, выявить закономерности	Фрагментарное умение обосновывать сущность происходящего, выявить закономерности	Несистематическое применение умений обосновывать сущность происходящего, выявить закономерности	В целом успешное, но содержащее отдельные пробелы умение обосновывать сущность происходящего, выявить закономерности	Сформированное умение обосновывать сущность происходящего, выявить закономерности	Вопросы для оценки знаний и умений, задания в виде расчетных задач, тестовые задания
Формулирует признак классификации, выделяет соответствующие ему группы однородных «объектов», идентифицирует общие свойства элементов этих групп, оценивает полноту результатов классификации, показывает прикладное назначение классификационных групп					

Планируемые результаты освоения компетенции (индикатора достижения компетенции)	Уровень освоения				Оценочное средство
	«неудовлетворительно»	«удовлетворительно»	«хорошо»	«отлично»	
Знать: - признаки классификации	Фрагментарное представление о признаках классификации	Неполные представления о признаках классификации	Сформированные, но содержащие отдельные пробелы представления о признаках классификации	Сформированные систематические представления о признаках классификации	Вопросы для оценки знаний и умений, задания в виде расчетных задач, тестовые задания
Уметь: - выделять соответствующие ему группы однородных «объектов», идентифицировать общие свойства элементов этих групп, оценивать полноту результатов классификации, показывать прикладное назначение классификационных групп	Фрагментарное умение выделять соответствующие ему группы однородных «объектов», идентифицировать общие свойства элементов этих групп, оценивать полноту результатов классификации, показывать прикладное назначение классификационных групп	Несистематическое применение умений выделять соответствующие ему группы однородных «объектов», идентифицировать общие свойства элементов этих групп, оценивать полноту результатов классификации, показывать прикладное назначение классификационных групп	В целом успешное, но содержащее отдельные пробелы умение выделять соответствующие ему группы однородных «объектов», идентифицировать общие свойства элементов этих групп, оценивать полноту результатов классификации, показывать прикладное назначение классификационных групп	Сформированное умение выделять соответствующие ему группы однородных «объектов», идентифицировать общие свойства элементов этих групп, оценивать полноту результатов классификации, показывать прикладное назначение классификационных групп	Вопросы для оценки знаний и умений, задания в виде расчетных задач, тестовые задания
Грамотно, логично, аргументировано формирует собственные суждения и оценки. Отличает факты от мнений, интерпретаций, оценок и т.д. в рассуждениях других участников деятельности					
Знать: - отличия фактов от мнений, интерпретаций и	Фрагментарное представление об отличиях фактов от мнений,	Неполные представления об отличиях фактов от мнений, интерпретации	Сформированные, но содержащие отдельные пробелы представления	Сформированные систематические представления об отличиях	Вопросы для оценки знаний и умений, задания в виде расчетных задач, тестовые

Планируемые результаты освоения компетенции (индикатора достижения компетенции)	Уровень освоения				Оценочное средство
	«неудовлетвори тельно»	«удовлетвори тельно»	«хорошо»	«отлично»	
оценок	интерпретаци й и оценок	й и оценок	я об отличиях фактов от мнений, интерпретаци й и оценок	фактов от мнений, интерпретаци й и оценок	задания
Уметь: - формировать собственные суждения и оценки, отличать факты от мнений, интерпретаци й, оценок	Фрагментарно е умение формировать собственные суждения и оценки, отличать факты от мнений, интерпретаци й, оценок	Несистематич еское умение формировать собственные суждения и оценки, отличать факты от мнений, интерпретаци й, оценок	В целом успешное, но содержащее отдельные пробелы умение формировать собственные суждения и оценки, отличать факты от мнений, интерпретаци й, оценок	Сформирован ное умение формировать собственные суждения и оценки, отличать факты от мнений, интерпретаци й, оценок	Вопросы для оценки знаний и умений, задания в виде расчетных задач, тестовые задания
Аргументировано и логично представляет свою точку зрения посредством и на основе системного описания					
Знать: - основные принципы системного описания	Фрагментарно е представлени е об основных принципах системного описания	Неполные представлени я об основных принципах системного описания	Сформирован ные, но содержащие отдельные пробелы представлени я об основных принципах системного описания	Сформирован ные систематичес кие представлени я о об основных принципах системного описания	Вопросы для оценки знаний и умений, задания в виде расчетных задач, тестовые задания
Уметь: - представить свою точку зрения посредством и системного описания	Фрагментарно е умение представить свою точку зрения посредством и системного описания	Несистематич еское умение представить свою точку зрения посредством и системного описания	В целом успешное, но содержащее отдельные пробелы умение представить свою точку зрения посредством и системного описания	Сформирован ное умение представить свою точку зрения посредством и системного описания	Вопросы для оценки знаний и умений, задания в виде расчетных задач, тестовые задания



## 7. 2. Вопросы для оценки знаний и умений, характеризующих формирование компетенций

Шифр компетенции	Вопросы	Правильный ответ
<b>УК-4</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 ex?	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
<b>УК-10</b>	11. Какую библиотеку необходимо использовать, для того чтобы получить доступ к модулю, написанному на Python из C	Python/C API
	12. В файле с каким расширением находятся директории для поиска модулей Python?	pth
	13. Как называется интегрированный набор компонентов и инструментов, который предоставляет готовое решение для решения определенной задачи в языке Python?	Фреймворк
	14. Как записывается в Python, пустой оператор?	pass
	14.С помощью какой функции можно посмотреть все методы и атрибуты, связанные с определенным объектом в Python?	dir()
	15.Как называется коллекция модулей или функций, предназначенных для использования другими приложениям? Данная коллекция содержит набор готовых решений для различных задач и обеспечивает функциональность,	Библиотека

	которую можно использовать в своём приложении.	
	16. Как называется механизм, который позволяет создавать изолированные окружения для установки и использования пакетов Python?	Виртуальное окружение
	17. Как называется инструмент, который позволяет управлять установкой, обновлением и удалением библиотек и зависимостей в проектах на языке Python?	Менеджер пакетов
	18. Как называется приложение, если в нем каждый процесс имеет свой собственный набор ресурсов, включая память, открытые файлы, сетевые соединения и другие системные ресурсы?	Микропроцессорное
	19. Как называется приложение, если в нем несколько потоков выполняются в рамках одного процесса?	Многопоточное
	20. Какая встроенная функция возвращает словарь, содержащий все глобальные переменные, доступные в текущей области видимости в языке Python?	globals()

### 7.3. Практико-ориентированные задания

Шифр компетенции	Практико-ориентированные задания	Правильный ответ
<b>УК-4</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>	Неполное ветвление

	<p>else:  <i>Блок инструкций 2</i></p> <p>Как называется условная инструкция <i>if</i>, если в ней отсутствует слово <i>else</i> и последующий за ней блок инструкций?</p>	
	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
	<p>8. Напишите название функции, которая скрывается под знаком «?»</p> <p>?(x) = 1, если x &gt; 0,  ?(x) = -1, если x &lt; 0,  ?(x) = 0, если x = 0.</p>	sign
	9. Какого типа будет значение, если разделить два значения типа <i>int</i> (язык программирования Python)?	float
УК-10	<p>10. Результат работы программы на языке Python:</p> <pre>def factorial(n):     res = 1     for i in range(1, n + 1):         res *= i     return res print(factorial(3))</pre>	6
	<p>11. Результат работы программы на языке Python:</p> <pre>def max(a, b):     if a &gt; b:         return a     else:         return b print(max(3, 5))</pre>	5
	<p>12. Результат работы программы на языке Python:</p> <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
	<p>13. Результат работы программы на языке Python:</p> <pre>def max(*a):     res = a[0]     for val in a[1:]:</pre>	6

	<pre> if val &gt; res:     res = val return res  print(max(6, 5, 4)) </pre>	
	<p>14. Результат работы программы на языке Python:</p> <pre> def f():     global a     a = 1     print(a)  a = 0 f() </pre>	1
	<p>15. Как называется прием, который используется в подпрограмме данной программы, написанной на языке Python:</p> <pre> def factorial(n):     if n == 0:         return 1     else:         return n * factorial(n - 1)  print(factorial(5)) </pre>	Рекурсия
	<p>16. Результат работы кода на языке Python:</p> <pre> sum = lambda a, b: a + b  print(sum(4, 5)) </pre>	9
	<p>17. 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>18. 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

#### 7.4. Тесты

Шифр компетенции	Тестовые задания	Правильный ответ
<b>УК-4</b>	<p>1. Равны ли множества <math>A = \{1, 2, 3\}</math> и <math>B = \{3, 2, 3, 1\}</math>?</p> <p>1. Не равны 2. Равны 3. Второе множество больше первого 4. Первое множество больше второго</p>	2

	<p>2. Какие операции для целых чисел возвращают вещественное число (float) в языке Python?</p> <p>1. +, -  2. //, *  3. +, *  4. **, /</p>	4
	<p>3. Какая математическая функция может быть использована для нахождения остатка от деления ( в том числе для чисел с плавающей точкой) в языке Python?</p> <p>1. fmod()  2. factorial()  3. fabs()  4. frexp()</p>	1
	<p>4. Команда print (11 % 3), написанная на языке Python, выдаст результат:</p> <p>1. 3  2. 2  3. 3.6666666666666665</p>	2
	<p>5. Что покажет данный код (язык программирования Python)?</p> <pre>for i in range(5):     if i % 2 == 0:         continue     print(i)</pre> <p>1.ошибку  2.Числа 1, 3 и 5  3.Числа 2 и 4  4.Числа 1и 3</p>	4
	<p>6. Чему будет равна переменная <i>d</i> после выполнения программы, написанной на языке Python?</p> <pre>a = 10 b = 3 b = a // b d = a % (b - 2)</pre> <p>1.3  2.1  3.10  4. 0</p>	4
<b>УК-10</b>	<p>7. Сколько библиотек можно импортировать в один проект в языке Python?</p> <p>1. Не более 5  2. Не более 3  3. Не более 10  4. Неограниченное количество</p>	4

8. Что покажет этот код, написанный на языке Python? <pre>for j in 'Hi! I\'m mister Robert':     if j == '\':         print("Найдено")         break else:     print ("Готово")</pre> <p>1 Ошибку в коде  2. "Найдено" и "Готово"  3. "Готово"  4. "Найдено"</p>	4
9. Какая функция выводит что-либо на консоль в языке Python? <p>1. out()  2. write()  3. print()  4. log()</p>	3
10. Какая библиотека отвечает за время в языке Python? <p>1. clock  2. time  3. localtime</p>	2
11. Как получить данные от пользователя в языке Python? <p>1. Использовать метод readLine()  2. Использовать метод input()  3. Использовать метод get()  4. Использовать метод read()</p>	2
12. Что будет показано в результате действия программы, написанной на языке Python? <pre>name = "John" print('Hi, %s' % name)</pre> <p>1. "Hi, name"  2. "Ошибка"  3. "Hi, John"  4."Hi, "</p>	3

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

### Основная литература

1. Гуриков, С. Р. Основы алгоритмизации и программирования на Python : учебное пособие / С. Р. Гуриков. — Москва : ИНФРА-М, 2022. — 343 с. — (Высшее образование: Бакалавриат). — ЭБС ZNANIUM.com. — URL: <https://znanium.com/catalog/product/1356003> (дата обращения: 15.08.2023).

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. Маккинни, У. Python и анализ данных. Первичная обработка данных с применением pandas, NumPy и Jupiter / У. Маккинни ; пер. А. А. Слинкина. - 3-е изд. - Москва : ДМК Пресс, 2023. - 536 с. - ISBN 978-5-93700-174-0. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/2150530>.

4. Саммерфильд, М. Python на практике. Создание качественных программ с использованием параллелизма, библиотек и паттернов : практическое пособие / М. Саммерфильд ; пер. с англ. А. А. Слинкина. - 2-е изд. - Москва : ДМК Пресс, 2023. - 340 с. - ISBN 978-5-89818-322-6. - Текст : электронный. - URL: <https://znanium.com/catalog/product/2102609>.

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

1. Pyru 1.0.9 [Электронный ресурс]: сайт. - Режим доступа: <https://pypi.python.org/pypi/pyru>
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.znanium.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 h11p:i гэб.рф"
25. Финансовая справочная система «Финансовый директор» <http://www.lfd.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.znanium.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.