

**«Финансовый университет при Правительстве Российской Федерации»  
(Финуниверситет)**

**Калужский филиал Финуниверситета**

**Кафедра «Бизнес-информатика и высшая математика»**

**«УТВЕРЖДАЮ»**

**Директор Калужского филиала  
Финуниверситета**



*В.А. Матчинов*

**В.А. Матчинов**

**«27» июня 2024 г.**

**Костенко А.В.**

**АЛГОРИТМЫ И СТРУКТУРЫ ДАННЫХ В ЯЗЫКЕ PYTHON**

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**

для студентов, обучающихся по направлению подготовки

38.03.05 «Бизнес-информатика»

Образовательная программа бакалавриата: «Цифровая трансформация  
управления бизнесом»  
очная форма обучения

*Рекомендовано Ученым советом Калужского филиала Финуниверситета  
(протокол № 16 от 27.06. 2024 г.)*

Одобрено кафедрой «Бизнес – информатика и высшая математика»  
Калужского филиала Финуниверситета  
(протокол № 12 от 27 июня 2024 г.)


**КАЛУГА 2024**

Рабочая программа предназначена для преподавания дисциплины «Алгоритмы и структуры данных в языке Python» студентам, обучающимся по направлению подготовки 38.03.05 «Бизнес-информатика», образовательная программа бакалавриата: «Цифровая трансформация управления бизнесом» по очной форме обучения.

В рабочей программе излагаются планируемые результаты освоения дисциплины, содержание дисциплины, тематика и содержание семинаров и практических занятий, технологии их проведения. В рабочей программе дисциплины приводится перечень учебно-методического обеспечения для самостоятельной работы обучающихся, фонд оценочных средств для проведения промежуточной аттестации обучающихся, перечень основной и дополнительной литературы, а также ресурсов информационно-телекоммуникационной сети Интернет.

СОГЛАСОВАНО:


Заместитель директора  
по учебно-методической работе  
«27» июня 2024 г.

 /Орловцева О.М./

Начальник учебно-методического отдела  
«27» июня 2024 г.

 /Толстикова В.С./

Заведующий кафедрой  
«Бизнес-информатика и высшая математика»  
«27» июня 2024 г.

 /Дробышева И.В./

## Содержание приложения к РПД

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 (модуля) в зачетных единицах и в академических часах с выделением объема аудиторной (лекции, семинары) и самостоятельной работы обучающихся	5
5.	Содержание дисциплины, структурированное по темам (разделам) дисциплины с указанием их объемов (в академических часах) и видов учебных занятий	6
5.1	Содержание дисциплины	6
5.2	Учебно-тематический план	8
5.3	Содержание семинаров, практических занятий	10
6.	Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине	13
6.1	Перечень вопросов, отводимых на самостоятельное освоение дисциплины, формы внеаудиторной самостоятельной работы	13
6.2	Перечень вопросов, заданий, тем для подготовки к текущему контролю	14
7.	Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине	17
8.	Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины	24
9	Перечень ресурсов информационно – телекоммуникационной сети «Интернет», необходимых для освоения дисциплины	25
10.	Методические указания для обучающихся по освоению дисциплины	26
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	28
11.2	Современные профессиональные базы данных и информационные справочные системы	29
11.3	Сертифицированные программные и аппаратные средства защиты информации	29
12.	Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине	29

## 1. Наименование дисциплины

«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

Таблица 1

Код компетенции	Наименование компетенции	Индикаторы достижения компетенции	Результаты обучения (владения, умения и знания), соотнесенные с компетенциями/индикаторами 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.	<b>Знать</b> методы и средства получения, представления, хранения и обработки данных. <b>Уметь</b> получать, представлять, хранить и обрабатывать данные.
		2. Демонстрирует владение профессиональными пакетами прикладных программ.	<b>Знать:</b> основные профессиональные пакеты прикладных программ. <b>Уметь:</b> использовать профессиональные пакеты прикладных программ.
		3. Выбирает необходимое прикладное программное обеспечение в зависимости от решаемой задачи.	<b>Знать:</b> рынок прикладного ПО. <b>Уметь:</b> формировать критерии выборы и осуществлять выбор ПО.
		4. Использует прикладное программное обеспечение для решения конкретных прикладных задач.	<b>Знать:</b> прикладное ПО. <b>Уметь:</b> решать конкретные прикладные задачи.
УК-10	Способность осуществлять поиск, критически анализировать, обобщать и систематизировать информацию, использовать системный подход для решения поставленных задач	1. Четко описывает состав и структуру требуемых данных и информации, грамотно реализует процессы их сбора, обработки и интерпретации.	<b>Знать</b> состав и структуру данных, методы сбора, обработки и интерпретации данных и информации <b>Уметь</b> применять методы сбора, обработки и интерпретации данных и информации при решении конкретных задач
		2. Обосновывает сущность происходящего, выявляет закономерности, понимает природу вариативности.	<b>Знать</b> принципы выявления закономерностей, теорию изменчивости данных <b>Уметь</b> выявлять закономерности реальных данных, оценивать их вариативность
		3. Формулирует признак классификации, выделяет соответствующие ему группы однородных объектов, идентифицирует общие свойства элементов этих групп, оценивает полноту результатов классификации, показывает прикладное	<b>Знать</b> подходы к классификации данных <b>Уметь</b> выполнять классификацию при решении конкретных задач и оценивать полноту выполнения классификации

		назначение классификационных групп.	
		4. Грамотно, логично, аргументировано формирует собственные суждения и оценки. Отличает факты от мнений и интерпретаций, оценок и т.д. в рассуждениях других участников деятельности.	<b>Знать</b> методы аргументации, принципы оценивания, интерпретации информации <b>Уметь</b> применять методы аргументации, принципы оценивания, интерпретации информации
		5. Аргументировано и логично представляет свою точку зрения посредством и на основе системного описания.	<b>Знать</b> принципы и методы аргументированного и логичного представления своей точки зрения посредством и на основе системного описания <b>Уметь</b> аргументировано и логично представлять свою точку зрения посредством и на основе системного описания

### 3. Место дисциплины в структуре образовательной программы

Дисциплина «Алгоритмы и структуры данных в языке Python» имеет порядковый номер Б.1.1.2.7. в структуре основной образовательной программы по направлению подготовки 38.03.05 – Бизнес-информатика, ОП «Цифровая трансформация управления бизнесом» и относится к циклу «Математика и информатика».

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

Таблица 2

Вид учебной работы по дисциплине	Всего (в з/ед. и часах)	Семестр 1 (в часах)	Семестр 2 (в часах)
<b>Общая трудоемкость дисциплины</b>	<b>6 зач.ед. / 216 час.</b>	<b>108</b>	<b>108</b>
<b>Контактная работа - Аудиторные занятия</b>	<b>100</b>	<b>50</b>	<b>50</b>
<i>Лекции</i>	<i>32</i>	<i>16</i>	<i>16</i>
<i>Семинары, практические занятия</i>	<i>68</i>	<i>34</i>	<i>34</i>
<b>Самостоятельная работа</b>	<b>116</b>	<b>58</b>	<b>58</b>
Вид промежуточной аттестации	зачет, экзамен	зачет	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. Именованное 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 запаковка 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**



Таблица 3

№ п/п	Наименование тем (разделов) дисциплины	Трудоемкость в часах					Формы текущего контроля успеваемости
		Всего	Контактная работа - Аудиторная работа			Самостоятельная работа	
			Общая, в т.ч.:	Лекции	Семинары, практические занятия		
1.	Введение в программирование на Python	14	6	2	4	8	Самостоятельное решение задач (программирование), выступление на семинаре
2.	Управляющие конструкции, списки и кортежи	16	6	2	4	10	Самостоятельное решение задач (программирование), выступление на семинаре
3.	Словари, множества и выражениягенераторы	18	8	2	6	10	Самостоятельное решение задач (программирование), выступление на семинаре
4.	Функции	20	10	4	6	10	Самостоятельное решение задач (программирование), выступление на семинаре
5.	Работа с файлами и обработка исключительных ситуаций	18	8	2	6	10	Самостоятельное решение задач (программирование), выступление на семинаре
6.	Модули и пакеты	20	10	4	6	10	Самостоятельное решение задач (программирование), выступление на семинаре
7.	Введение в объектно- ориентированное программирование	18	8	2	6	10	Самостоятельное решение задач (программирование), выступление на семинаре
8.	Объектно- ориентированное программирование в Python	22	12	4	8	10	Самостоятельное решение задач (программирование), выступление на семинаре

9.	Введение в функциональное программирование	16	6	2	4	10	Самостоятельное решение задач (программирование), выступление на семинаре
10.	Функциональное программирование в Python	18	8	2	6	10	Самостоятельное решение задач (программирование), выступление на семинаре
11.	Структуры данных: массивы, стеки, очереди, списки	20	10	4	6	10	Самостоятельное решение задач (программирование), выступление на семинаре
12.	Алгоритмы поиска и сортировки	16	8	2	6	8	Самостоятельное решение задач (программирование), выступление на семинаре
	В целом по дисциплине	216	100	32	68	116	Согласно учебному плану: контрольные работы
	Итого в %		46	32	68	54	

### 5.3 Содержание семинаров, практических занятий

Таблица 4

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

	<i>Источники: 8.1; 8.2; 8.3</i>	технологии)
2. Управляющие конструкции, списки и кортежи	<p>Базовые числовые типы, строки, списки, словари, переменные, базовые операторы.</p> <p><i>Источники: 8.1; 8.2; 8.3</i></p>	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий (не менее 30% времени на интерактивные технологии)
3. Словари, множества и выражения-генераторы	<p>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.</p> <p><i>Источники: 8.1; 8.2; 8.3</i></p>	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий (не менее 30% времени на интерактивные технологии)
4. Функции	<p>Создание функций, область видимости переменной, передача аргументов в функцию. Лямбда-функции.</p> <p><i>Источники: 8.1; 8.2; 8.3</i></p>	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий (не менее 30% времени на интерактивные технологии)
5. Работа с файлами и обработка исключительных ситуаций	<p>Исключения. Инструкция try...except...else...finally. Классы встроенных исключений. Создание пользовательских исключений. Инструкция assert.</p> <p>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.</p> <p><i>Источники: 8.1; 8.2; 8.3</i></p>	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 import и from. Создание собственных модулей и пакетов.</p> <p><i>Источники: 8.1; 8.2; 8.3</i></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 super().</p> <p><i>Источники: 8.1; 8.2; 8.3</i></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><i>Источники: 8.1; 8.2; 8.3</i></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.</p> <p>Декораторы в Python: использование и создание собственных декораторов.</p> <p><i>Источники: 8.1; 8.2; 8.3</i></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.</p> <p>Итераторы в Python, итерируемый тип данных. Модуль itertools.</p> <p><i>Источники: 8.1; 8.2; 8.3</i></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><i>Источники: 8.1; 8.2; 8.3</i></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). Реализация на Python эффективных методов сортировки: быстрая сортировка; сортировка Шелла; сортировка слиянием.</p> <p><i>Источники: 8.1; 8.2; 8.3</i></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**

При изучении дисциплины используются следующие виды самостоятельной работы:

- Выполнение внеаудиторной контрольной работы.
- Подготовка к зачету.
- Работа с литературой: проработка учебного материала (по конспектам лекций, учебной и научной литературе).
- Самостоятельное изучение тем теоретического курса, не вошедших в лекционный материал.
- Работа с тестами и вопросами для самопроверки: выполнение тренировочных тестов.
- Подготовка к интерактивному занятию.
- Подготовка к тестированию.

Таблица 5

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 упаковка 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 объектноориентированное 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 (согласно таблице 2)

Текущий контроль осуществляется в ходе учебного процесса и контроля самостоятельной работы студентов, а также по результатам выполнения аудиторных самостоятельных работ, контрольной работы. Основными формами текущего контроля знаний являются:

- обсуждение вопросов по теме практических занятий, в том числе выступления у доски;

- выполнение домашних и аудиторных практических работ и обсуждение результатов;
- выполнение контрольных работ.

Промежуточная аттестация проводится в форме зачета и экзамена.

### **Примерные вопросы к контрольной работе (1 семестр)**

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.

### **Примерные задания контрольной работы (1 семестр)**

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 'Eeny, meeny, miney, мое'

### **Примерные вопросы к контрольной работе (2 семестр)**

1. Списки. Создание, основные операции.
2. Основные методы списка.
3. Кортежи. Создание, основные методы и операции.
4. Словари. Создание, основные операции.
5. Методы для работы со словарями.
6. Множества. Создание, основные методы и операции.
7. Переменные. Правила именования переменных.
8. Динамическая типизация.
9. Операторы сравнения и логические операторы.
10. Инструкция if...else.
11. Инструкция цикла while.
12. Инструкция цикла for.
13. Создание и вызов функции.
14. Передача аргументов функцию.
15. Функции-генераторы.
16. Лямбда-функции.
17. Модули. Инструкции import и from.
18. Базовые принципы объектно-ориентированного программирования.
19. Класс, метод класса, атрибут класса. Определение класса и создание экземпляра класса.
20. Конструктор и деструктор.



21. Наследование.
22. Абстрактные методы класса.
23. Статические методы класса.
24. Свойства класса.
25. Исключения. Обработка исключений.
26. Пользовательские исключения.

### **Примерные задания контрольной работы (2 семестр)**

1. 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.
2. Реализовать декоратор с именем `not_none`, который генерирует исключительную ситуацию если декорируемая функция вернула значения `None`. (20 баллов)
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. 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.*

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

Перечень компетенций, формируемых в процессе освоения дисциплины, содержится в разделе 2. Перечень планируемых результатов освоения образовательной программы с указанием индикаторов их достижения, соотнесенных с планируемыми результатами обучения по дисциплине.

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

Таблица 7

Наименование компетенции	Наименование индикаторов достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции	Типовые контрольные задания
Способность использовать прикладное программное обеспечение при решении профессиональных задач (УК-4)	1. Использует основные методы и средства получения, представления, хранения и обработки данных.	<b>Знать:</b> методы и средства получения, представления, хранения и обработки данных.  <b>Уметь:</b> получать, представлять, хранить и обрабатывать данные.	Напишите скрипт на Python обрабатывающий текстовый файл так, что в нем все заглавные буквы преобразуются в строчные.  Напишите скрипт на Python обрабатывающий текстовый файл и получение на его основе словаря, ключами являются слова текста, а значениями количество встречаемости слов. Сохраните в файле CSV.
	2. Демонстрирует владение профессиональными пакетами прикладных программ.	<b>Знать:</b> основные профессиональные пакеты прикладных программ.  <b>Уметь:</b> использовать профессиональные пакеты прикладных программ.	Установите пакет Anaconda запустите программу Jupyter Notebook составьте тестовый код на Python.  Используя программу Jupyter Notebook составьте код на Python, определяющий существование треугольника по трем заданным длинам сторон. Если существует, то определить его тип. Используйте библиотеку math.

<p>Способность осуществлять поиск, критически анализировать, обобщать и систематизировать информацию, использовать системный подход для решения поставленных задач (УК-10)</p>	<p>3. Выбирает необходимое прикладное программное обеспечение в зависимости от решаемой задачи.</p>	<p><b>Знать:</b> рынок прикладного ПО.</p> <p><b>Уметь:</b> формировать критерии выборы и осуществлять выбор ПО.</p>	<p>Выберите библиотеку Python для работы с массивами. Выполните программный код бинарного поиска данных.</p> <p>Выберите библиотеку Python для хэширования данных. Выполните программный код быстрого поиска текстовых строк.</p>
	<p>4. Использует прикладное программное обеспечение для решения конкретных прикладных задач.</p>	<p><b>Знать:</b> прикладное ПО.</p> <p><b>Уметь:</b> решать конкретные прикладные задачи.</p>	<p>Установите пакет Anaconda запустите программу Jupyter Notebook составьте тестовый код на Python.</p> <p>Реализуйте программу Python которой в качестве аргумента командной строки передается имя CSV-файла, в первом столбце находятся числа, которые необходимо отсортировать. Программа создает новый файл, в котором первый столбец отсортирован.</p>
	<p>1. Четко описывает состав и структуру требуемых данных и информации, грамотно реализует процессы их сбора, обработки и интерпретации.</p>	<p><b>Знать:</b> процессы сбора, обработки и интерпретации информации.</p> <p><b>Уметь:</b> описывать состав и структуру требуемых данных и информации.</p>	<p>Выполните обработку текстового файла, в котором зафиксированы факты продаж, необходимого для подсчета суммарных продаж по различным географическим подразделениям фирмы. На основе полученных данных составьте CSV файл, хранящий полученные результаты.</p> <p>Опишите структуру CSV файла фиксирующего факты продаж, необходимого для подсчета суммарных продаж по различным географическим подразделениям фирмы.</p>

2. Обосновывает сущность происходящего, выявляет закономерности, понимает природу вариабельности.	<p><b>Знать:</b> принципы описания закономерностей и вариабельности</p> <p><b>Уметь:</b> обосновывать сущность происходящего</p>	<p>Опишите закономерности/вариабельность в синтаксических конструкциях Python для работы со списками и словарями.</p> <p>Определите количество элементов списка, значения которых совпадают с ключами заданного словаря и количество элементов списка, значения которых равны значениям в словаре.</p>
3. Формулирует признак классификации, выделяет соответствующие ему группы однородных «объектов», идентифицирует общие свойства элементов этих групп, оценивает полноту результатов классификации, показывает прикладное назначение классификационных групп.	<p><b>Знать:</b> подходы к выполнению классификации</p> <p><b>Уметь:</b> выполнять классификацию и оценивать полноту выполнения классификации.</p>	<p>Опишите подходы используемые в Python для группировки функций, ориентированных на обработку однотипных объектов и решение близких задач.</p> <p>Выполните классификацию с использованием бинарного дерева. Оцените количество операций и глубину дерева.</p>
4. Грамотно, логично, аргументировано формирует собственные суждения и оценки. Отличает факты от мнений, интерпретаций, оценок и т.д. в рассуждениях других участников деятельности.	<p><b>Знать:</b> методы аргументации и оценкам.</p> <p><b>Уметь:</b> формировать суждения и оценки.</p>	<p>Сформируйте собственные суждения о применимости Python для решения задач машинного обучения.</p> <p>Выполните задачу построения бинарного дерева двумя способами: на основе списков и на основе узлов и ссылок. Оцените эффективность работы каждого способа.</p>
5. Аргументированно и логично представляет свою точку зрения посредством и на основе системного описания.	<p><b>Знать:</b> подходы к описанию собственного мнения.</p> <p><b>Уметь:</b> представлять свою точку зрения.</p>	<p>Сформулируйте задачи по обработке массивов, которые вы сможете решить на основе языка Python</p> <p>Выскажите свою точку зрения о применимости Python для создания крупной корпоративной транзакционной системы на языке программирования Python.</p>

## **Примерные вопросы для подготовки к зачету**

### **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. Операции со словарями, учитывающие возможное отсутствие ключа. Операции многоэлементного изменения словарей. Операции поэлементного извлечения из словаря и их использование.
10. Множества в Python. Основные способы создания, получения и изменения значений. Обход множеств. Выполнение основных операций с парой множеств в Python.
11. Кортежи в Python. Отличия кортежей от списков. Распаковка и частичная распаковка кортежей.
12. Выражения генераторы и генераторы списков в Python. Использование условий в генераторах.
13. Функции стандартной библиотеки для работы с контейнерами.

### **Тема 4. Функции**

14. Объявление и вызов функции в Python. Параметры функции со значением по умолчанию и комментирование функции.
15. Функция. Получение информации о функции. Способы передачи параметров при вызове функции.
16. Передача переменного количества параметров (именованных и не именованных) в функции Python.
17. Вызов функции с позиционными параметрами, находящимися в списке, и именованными параметрами, находящимися в словаре.

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

18. Синтаксис и семантика обработки исключительных ситуаций в Python.
19. Создание пользовательских исключений и инструкция assert.
20. Базовые операции для работы с файлами в Python. Использование инструкции with ... as на примере работы с файлами.
21. CSV файлы. Методы и функции работы с CSV файлами.
22. Бинарные файлы. Методы и функции работы с бинарными файлами.

## **Тема 6. Модули и пакеты**

22. Модули в Python и их отличие от скриптов Python. Варианты синтаксиса импорта модуля и объектов модуля.
23. Применение импортированных объектов. Порядок поиска модулей и специфика их загрузки. Загрузка модулей из глобального репозитория.

## **Примерные вопросы для подготовки к экзамену 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 лямбдафункций 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 работа 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 экзаменационного билета

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

Кафедра	«Бизнес-информатика и высшая математика»		
Дисциплина	Алгоритмы и структуры данных в языке Python		
Факультет/ филиал	Калужский филиал	Форма обучения	очная
Семестр	2	Направление	38.03.05 Бизнес-информатика
Профиль	Цифровая трансформация управления бизнесом» (ИТ-менеджмент в бизнесе)		

## ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 00

**Задание 1. (15 баллов)** Функции высшего порядка. Встроенные функции высшего порядка. Реализация `sorted(seq, key=None, reverse=False)` в Python и примеры использования.



**Задание 2. (20 баллов)** Дана цифра  $x$  (от 1 до 8). Используя простой генератор списков выведите список чисел от  $xxx$  до  $(x+1)(x+1)(x+1)$  включительно. Например: ввели цифру 7: печатаем список от 777 до 888.

**Задание 3. (25 баллов)** С помощью функций высшего порядка `map/filter/reduce` вычислите сколько раз символ «#» встречается в списке строк `w = [' МЕРЧ####', '# м*об#**', 'СТив', '# кРи#пер*']`. Удалите все элементы, не являющиеся буквами. Отсортируйте список слов по алфавиту. Запишите в файл каждое слово в списке с новой строки, измените регистры букв в каждом слове, согласно правилам русского языка (имена героев с большой буквы).

Подготовил:

Утверждаю:  
Зав. кафедрой

Дата

## **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: 16.05.2023). — Текст : электронный.

2. Шелудько, В. М. Язык программирования высокого уровня Python. Функции, структуры данных, дополнительные модули : учебное пособие / В. М. Шелудько ; Южный федеральный университет. - Ростов-на Дону ; Таганрог : Издательство Южного федерального университета, 2017. - 107 с. — ЭБС ZNANIUM.com. - URL: <https://znanium.com/catalog/product/1021664> (дата обращения: 16.05.2023). — Текст : электронный.

### ***Дополнительная литература:***

3. Северенс, Ч. Введение в программирование на Python / Ч. Северенс. — 2-е изд., испр. — Москва : Национальный Открытый Университет «ИНТУИТ», 2016. — 231 с. — ЭБС Университетская библиотека online. — URL: <https://biblioclub.ru/index.php?page=book&id=429184> (дата обращения: 16.05.2023). — Текст : электронный.

## 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/>.
9. The Python Tutorial // <https://docs.python.org/3/tutorial/index.html>.
10. The Python Standard Library // <https://docs.python.org/3/library/index.html>.
11. SciPy // <http://docs.scipy.org/doc/scipy/reference/>.
12. NumPy User Guide // <http://docs.scipy.org/doc/numpy/user/index.html>.
13. Электронная библиотека Финансового университета (ЭБ) <http://elib.fa.ru/>.
14. Электронно-библиотечная система BOOK.RU <http://www.book.ru>.
15. Электронно-библиотечная система «Университетская библиотека ОНЛАЙН» <http://biblioclub.ru/>.
16. Электронно-библиотечная система Znanium <http://www.znaniy.com>.
17. Электронно-библиотечная система издательства «ЮРАЙТ» <https://urait.ru/>.
18. Электронно-библиотечная система издательства Проспект <http://ebs.prospekt.org/books>.
19. Электронно-библиотечная система издательства «Лань» <https://e.lanbook.com/>.
20. Электронная библиотека Издательского дома «Гребенников» <https://grebennikon.ru/>.

21. Деловая онлайн-библиотека Alpina Digital  
<http://lib.alpinadigital.ru/>.
22. Научная электронная библиотека eLibrary.ru <http://elibrary.ru>.
23. Национальная электронная библиотека <http://нэб.рф/>.
24. СПАРК <https://spark-interfax.ru/>.
25. Система Thomson Reuters Eikon.
26. Бесплатный курс «Основы Python-разработки».  
<https://practicum.yandex.ru/python-free/>
27. Программирование на Python. <https://stepik.org/course/67/promo>

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

Обучающимся в рамках самостоятельной работы следует использовать Методические рекомендации по планированию и организации внеаудиторной самостоятельной работы студентов по образовательным программам бакалавриата и магистратуры в Финансовом университете, утвержденные Приказом ректора №1040/о от 11.05.2021 г.

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.

Проведение практических занятий осуществляется в компьютерных классах и включает в себя реализацию всех этапов проектирования и реализации алгоритмов.

Самостоятельная работа студентов проходит внеаудиторно. Организации самостоятельной работы служит учебно-тематический план изучения дисциплины. В данном плане указана тематика лекций, семинаров, вопросы и задания для самостоятельного изучения. Во время лекций необходимо конспектировать содержание лекции. После лекции необходимо отредактировать записи, оформить конспект, дополняя его содержание дополнительной информацией. При оформлении конспекта целесообразно выделять названия тем и формулировки вопросов, основные определения, примеры.

При затруднении в решении практических вопросов (задач), можно обратиться за консультацией (помощью) к преподавателю. Семинары проходят, как правило, в интерактивной форме и преподаватель учитывает активность обучающихся, направленную на решение предложенных вопросов (вариантов задач), а также вариантов ответов на решаемые вопросы (проблемы).

Не следует бояться дать неверный ответ или допустить иную ошибку: исправление и анализ ошибок в режиме общения с преподавателем и

сокурсниками в ходе семинара способствует более глубокому освоению учебного материала и предупреждает возникновение ошибок в дальнейшем. Домашние задания (подготовку к занятиям) следует осуществлять регулярно. Если то или иное задание, при подготовке к семинару вызвало затруднение, необходимо обратиться к преподавателю за консультацией. Регулярность в выполнении домашних заданий (подготовке к занятиям) - важный фактор качественного освоения дисциплины.

### **Методические рекомендации по обучению лиц с ограниченными возможностями здоровья**

Профессорско-преподавательский состав знакомится с психологофизиологическими особенностями обучающихся инвалидов и лиц с ограниченными возможностями здоровья (ОВЗ). При необходимости осуществляется дополнительная поддержка преподавания тьюторами, психологами, социальными работниками, прошедшими подготовку ассистентами.

В соответствии с методическими рекомендациями Минобрнауки РФ (утв. 8 апреля 2014 г. № АК-44/05вн) в курсе предполагается использовать социальноактивные и рефлексивные методы обучения, технологии социокультурной реабилитации с целью оказания помощи в установлении полноценных межличностных отношений с другими студентами, создании комфортного психологического климата в студенческой группе. Подбор и разработка учебных материалов производятся с учётом предоставления материала в различных формах: аудиальной, визуальной, с использованием специальных технических средств и информационных систем. Освоение дисциплины лицами с ОВЗ осуществляется с использованием средств обучения общего и специального назначения (персонального и коллективного использования). Материально-техническое обеспечение предусматривает приспособление аудиторий к нуждам лиц с ОВЗ. Форма проведения аттестации для студентов-инвалидов устанавливается с учётом индивидуальных психофизических особенностей.

Для студентов с ОВЗ предусматривается доступная форма предоставления заданий оценочных средств, а именно:

- в печатной или электронной форме (для лиц с нарушениями опорнодвигательного аппарата);
- в печатной форме или электронной форме с увеличенным шрифтом и контрастностью (для лиц с нарушениями слуха, речи, зрения);
- методом чтения ассистентом задания вслух (для лиц с нарушениями зрения). Студентам с инвалидностью увеличивается время на подготовку ответов на контрольные вопросы.

Для таких студентов предусматривается доступная форма предоставления ответов на задания, а именно:

- письменно на бумаге или набором ответов на компьютере (для лиц с нарушениями слуха, речи);
- выбором ответа из возможных вариантов с использованием услуг ассистента (для лиц с нарушениями опорно-двигательного аппарата);

- устно (для лиц с нарушениями зрения, опорно-двигательного аппарата).

При необходимости для обучающихся с инвалидностью процедура оценивания результатов обучения может проводиться в несколько этапов.

### **Методические рекомендации по выполнению расчетно-аналитической работы**

Методические рекомендации по выполнению **расчетно-аналитической работы** предусмотрены в «Методических рекомендациях по подготовке написанию и оформлению расчетно-аналитической работы», разрабатываемой преподавателем кафедры на учебный год, в котором реализуется учебная дисциплины

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

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

1. Антивирусная защита ESET NOD32
2. Windows, Microsoft Office
3. Astra Linux
4. Дистрибутив Python Anaconda (свободно распространяемое ПО).
5. Файловый менеджер Far.

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

1. Информационно-правовая система «Консультант Плюс»;
2. Информационно-правовая система «Гарант»;

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

Сертифицированные программные и аппаратные средства защиты информации не предусмотрены

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

- аудитории для проведения лекционных и семинарских занятий, оборудованные видеопроекционным оборудованием для презентаций, средствами звуковоспроизведения, экраном, доской меловой/интерактивной;

- библиотеку, имеющую рабочие места для студентов, оснащенные компьютерами с доступом к базам данных и сети Интернет

- компьютерные классы с набором лицензионного базового программного обеспечения для проведения практических занятий и выходом в глобальную сеть Internet;

### **Образовательные технологии**

В процессе изучения дисциплины «Алгоритмы и структуры данных в языке Python» предполагается:

- сопровождение курса лекций наглядной презентацией, включающей практические примеры, схемы, графики, табличный материал;

- рассмотрение на семинарских занятиях интерактивных ситуационных задач по проблематике дисциплины;

- деловые игры;

- разбор конкретных ситуаций, коллективное обсуждение проблем российской и зарубежной практики по изучаемым темам;

- виртуальное общение в течение срока изучения курса в целях обеспечения лекций и практических занятий необходимым материалом и также контроля самостоятельной работы студентов.