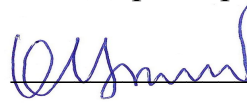


Федеральное государственное образовательное бюджетное
учреждение высшего образования
«Финансовый университет при Правительстве Российской Федерации»

Омский филиал Финуниверситета

Кафедра «Естественно-научные и гуманитарные дисциплины»

УТВЕРЖДАЮ
Директор Омского филиала
Финуниверситета
 О.Р. Онищенко
«23» июня 2026 г.

Шамис В.А.

СОВРЕМЕННЫЕ ТЕХНОЛОГИИ ПРИКЛАДНОГО ПРОГРАММИРОВАНИЯ И ОБРАБОТКИ ДАННЫХ

Рабочая программа дисциплины
для студентов, обучающихся по направлению подготовки
38.03.01 «Экономика», ОП «Бизнес-анализ, налоги и аудит»,
профиль: «Учет, анализ и аудит»
(программа подготовки бакалавра)

*Рекомендовано Ученым советом Омского филиала Финуниверситета
(протокол от «23» июня 2026 г. №34)*

*Одобрено кафедрой «Естественно-научных и гуманитарных дисциплин»
(протокол от «29» апреля 2026 г. №4)*

Омск 2026

Рабочую программу дисциплины разработал:
кандидат психологических наук Шамис В.А.

Современные технологии прикладного программирования и обработки данных. Рабочая программа дисциплины для студентов, обучающихся по направлению 38.03.01 «Экономика» ОП «Бизнес-анализ, налоги и аудит», профиль: «Учет, анализ и аудит» – Омск: Омский филиал Финуниверситета, кафедра «Естественно-научные и гуманитарные дисциплины», 2026. – 38 с.

© Омский филиал Финансового университета
при Правительстве Российской Федерации,
2026

СОДЕРЖАНИЕ

1.	Наименование дисциплины	5
2.	Перечень планируемых результатов обучения по дисциплине с указанием индикаторов их достижения и планируемых результатов обучения	5
3.	Место дисциплины в структуре образовательной программы	7
4.	Объем дисциплины (модуля) в зачетных единицах и в академических часах с выделением объема аудиторной (лекции, семинары) и самостоятельной работы обучающихся	7
5.	Содержание дисциплины, структурированное по темам (разделам) дисциплины с указанием их объемов (в академических часах) и видов учебных занятий	8
	5.1. Содержание дисциплины	8
	5.2. Учебно-тематический план	10
	5.3. Содержание семинаров, практических занятий	11
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 (согласно таблице 2)	13
7.	Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине	22
	7.1. Перечень компетенций, формируемых в процессе освоения дисциплины	22
	7.2. Показатели и критерии оценивания компетенций	23
	7.3. Типовые контрольные задания и иные материалы, необходимые для оценки индикаторов достижения компетенций, умений и знаний	27

7.4. Методические материалы, определяющие процедуру оценивания знаний, умений, владений	34
8. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины	34
9. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины	35
10. Методические указания для обучающихся по освоению дисциплины	36
11. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень необходимого программного обеспечения и информационных справочных систем	38
12. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине	38

1. Наименование дисциплины

Наименование дисциплины – Б.1.2.2.2.1.2. «Современные технологии прикладного программирования и обработки данных».

Целью учебной дисциплины «Современные технологии прикладного программирования и обработки данных» является знакомство с возможностями использования современного ИТ-инструментария для анализа данных, в том числе в финансово-экономических приложениях

Задачи дисциплины:

- Введение в программирование на языке Python;
- Знакомство с базовыми технологиями для анализа и обработки данных;
- Знакомство с технологиями работы со структурированными данными.

2. Перечень планируемых результатов обучения по дисциплине с указанием индикаторов их достижения и планируемых результатов обучения

В совокупности с другими дисциплинами части, формируемой участниками образовательных отношений, образовательного стандарта Финуниверситета дисциплина «Современные технологии прикладного программирования и обработки данных» обеспечивает инструментарий формирования следующих профессиональных компетенций бакалавра:

Таблица 1

Код компетенции	Наименование компетенции	Индикаторы достижения компетенции ¹	Результаты обучения (владения ² , умения и знания), соотнесенные с компетенциями/индикаторами достижения компетенции
ПКП-5	Способность к использованию специальных программных продуктов, применяемых для	1. Использует специальные программные продукты для выполнения бухгалтерско-	Знать: основы функционирования специальных программных продуктов, применяемых для выполнения бухгалтерско-аналитических и контрольных

¹ Заполняется при реализации актуализированных ОС ВО ФУ и ФГОС ВО 3++

² Владения формулируются только при реализации ОС ВО ФУ первого поколения и ФГОС ВО 3+

	выполнения бухгалтерско-аналитических и контрольных функций в экономическом субъекте	аналитических и контрольных функций в экономическом субъекте. 2. Демонстрирует владение специальными программными продуктами, применяемыми для выполнения бухгалтерско-аналитических и контрольных функций в экономическом субъекте.	функций в экономическом субъекте Уметь: использовать основной функционал специальных программных продуктов при выполнении бухгалтерско-аналитических и контрольных функций в экономическом субъекте Знать: особенности функционирования специальных программных продуктов и их применения для выполнения бухгалтерско-аналитических и контрольных функций в экономическом субъекте Уметь: применять специальные программные продукты для выполнения бухгалтерско-аналитических и контрольных функций в экономическом субъекте
УК – 15	Способность релевантно решать задачи, используя информационные ресурсы и информационно-коммуникационные технологии для достижения целей, связанных с профессиональной деятельностью, обучением, участием в жизни общества и других сферах жизни	1. Самостоятельно выбирает и использует цифровые средства общения, осуществляет поиск и/или создание контента в соответствии с целью взаимодействия, в том числе для организации совместной деятельности.	Знать: основные источники, а также, современные средства и инструменты поиска контента Уметь: выбирать формы и методы подачи контента для организации совместной работы
		2. Владеет навыками организации взаимодействия и коммуникации с помощью информационных	Знать: современные средства организации совместной удаленной работы Уметь: выбирать средства организации совместной деятельности

		систем и/или цифровых сервисов и технологий. Осуществляет подбор и применение различных информационно-коммуникационных средств для решения образовательных и профессиональных задач.	
--	--	--	--

3. Место дисциплины в структуре образовательной программы

Дисциплина «Современные технологии прикладного программирования и обработки данных» является элективной дисциплиной цикла профиля, модуль 1 «Технологии прикладного прогнозирования и анализа данных» по направлению подготовки 38.03.01 «Экономика» (бакалавриат).

При изучении дисциплины «Современные технологии прикладного программирования и обработки данных» необходимы знания, умения и компетенции, которые получены при изучении школьных курсов математики и информатики.

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

Изучение дисциплины «Современные технологии прикладного программирования и обработки данных» предусматривает чтение лекций, проведение семинарских занятий и самостоятельную работу студентов.

Общая трудоёмкость дисциплины составляет 3 зачетные единицы.

Вид промежуточной аттестации – зачет.

Вид учебной работы	Часы (очная форма)	
	Всего (в часах и з.е.)	Семестр 6 (в часах)
Общая трудоемкость дисциплины	108 (3 з.е.)	108 (3 з.е.)
Контактная работа - аудиторные занятия	34	34
Лекции	16	16
Семинары	18	18
Самостоятельная работа	74	74
Вид текущего контроля	Контрольная работа	Контрольная работа
Вид промежуточной аттестации	зачет	зачет

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

5.1. Программа дисциплины

Тема 1. Введение в программирование на языке Python

Задачи анализа данных, понятие набора данных (dataset). Подготовительные операции для выполнения анализа данных: загрузка данных, трансформация данных, изучение данных, очистка данных, визуализация данных.

Технологический стек анализа данных, построенный на базе языка программирования Python. Язык программирования Python: основные характеристики, возможности языка для решения задач анализа данных и машинного обучения. Версии языка программирования Python, дистрибутивы и библиотеки Python. Знакомство с дистрибутивом Anaconda и составом инструментов для задач анализа данных и машинного обучения, входящих в дистрибутив. Интерактивная оболочка IPython notebook: принципы работы и применение для решения задач анализа данных и машинного обучения

Тема 2. Основные синтаксические конструкции Python

Знакомство с типами данных и операциями, переменными. Возможности работы со строками в Python. Основные операции над строками, функции и методы для работы со строками. Структура программы. Инструкции выражений, операторы сравнения, логические операторы. Инструкция

ветвления `if...else`. Инструкция цикла `while`. Инструкция цикла `for` и Инструкции `break`, `continue`, `pass`, `else`.

Работа со списками в Python. Создание списка. Операции над списками.

Перебор элементов списка. Многомерные списки. Методы списков. Кортежи.

Работа со словарями в Python. Создание словаря. Операции над словарями.

Перебор элементов словаря. Методы для работы со словарями. Множества.

Тема 3. Базовые технологии для анализа данных

Знакомство с библиотеками `numpy` и `pandas` и решением базовых задач подготовительных операций для выполнения анализа данных с помощью этих библиотек.

Постановки задач машинного обучения. Объекты и признаки. Типы признаков: бинарные, номинальные, порядковые, количественные. Типы задач машинного обучения: классификация, регрессия, прогнозирование, кластеризация. Примеры задач решаемых методами машинного обучения. Проблема недообучения / переобучения.

Тема 4. Технологии работы со структурированными данными

Обзор технологий хранения данных: файловых систем, реляционных СУБД, OLAP, Data Warehouses, не реляционных (“NoSQL”) баз данных. Сравнительный анализ и области применения различных технологий хранения информации. Работа с файлами. Работа с реляционными базами данных на примере SQLite.

Краткий обзор основных видов не реляционных баз данных: хранилищ «ключ-значение», хранилище семейств колонок, документо-ориентированная СУБД, баз данных на основе графов. Сравнительный анализ и области применения не реляционных баз данных.

Хранение и обмен структурированной информацией в виде документов или сообщений. Форматы представления переносимой структурированной информации. Сравнение различных принципов представления структурированной информации: закрытые и открытые форматы, бинарное и текстовое представление данных.

Универсальные форматы хранения структурированной информации (разметки документов): CSV, XML, HTML (XHTML), JSON. Язык разметки XML: основные принципы построения и специфика использования. Построение схемы документа с помощью XML DTD или XML Schema. HTML (XHTML) – отличие от XML, специфика использования. Формат представления структурированной информации JSON: принципы построения, специфика использования.

Тема 5. Технологии обработки данных

Знакомство с различными классами информационно-аналитических систем. Технологии Data Mining. Технологии анализа больших объемов данных (Big Data): причины возникновения, основные особенности функционирования и специфика создания приложений.

Сравнительный анализ различных подходов к анализу экономически значимой информации: от построения систем отчетов до алгоритмов машинного обучения. Особенности построения информационно-аналитических систем с применением алгоритмов машинного обучения. Основные этапы создания информационно-аналитических систем с использованием алгоритмов машинного обучения

5.2. Учебно-тематический план

Таблица 2

№ п/ п	Наименование раздела (темы) дисциплины	Трудоемкость в часах						Формы тетекущего контроля успеваемости
		Всего часов	Аудиторная работа				Самостоя тельная работа Общая	
			Общая	Лекции	Практические и семинарские	В т.ч. занятия в интерактивных формах		
1.	Введение в программирование на языке Python	23	8	2	4	4	15	УО, ППЗ
2.	Основные синтаксические конструкции Python	23	8	4	4	4	15	УО, ППЗ

№ п/ п	Наименование раздела (темы) дисциплины	Трудоемкость в часах						Формы тетекущего контроля успеваемости
		Всего часов	Аудиторная работа				Самостоя тельная работа Общая	
			Общая	Лекции	Практические и семинарские	В т.ч. занятия в интерактивных формах		
3.	Базовые технологии для анализа данных	16	4	2	2	2	14	УО, ППЗ, КР
4.	Технологии работы со структурированным и данными	23	8	4	4	4	15	УО, ППЗ
5.	Технологии обработки данных	23	8	4	4	4	15	УО, ППЗ
	В целом по дисциплине	108	34	16	18	18	74	Контрольная работа
	ИТОГО в %					53%		

*Сокращения в таблице: УО – устный опрос; ППЗ – проверка практических заданий, КР – домашняя контрольная работа

5.3. Содержание практических и семинарских занятий

Практические и семинарские занятия служат для углубленного изучения материала дисциплины и приобретения практических навыков в соответствии с ее целями и задачами. Семинарские занятия предусматривают проведение дискуссий по темам занятия, обсуждения проблем в рамках тем в малой группе (работа в группе), подготовку докладов, сообщений, мультимедийных презентаций, выполнение контрольных работ.

Таблица 3

Наименование тем (разделов) дисциплины	Перечень вопросов для обсуждения на семинарских, практических занятиях, рекомендуемые источники из разделов 8,9 (указываются раздел и порядковый номер источника)	Формы проведения занятий
Тема 1. Введение в программирование на языке Python	Установка Python, установка дистрибутива Anaconda. Работа в интерактивном режиме интерпретатора. Интерактивная оболочка IPython notebook: принципы работы и применения. Среда программирования. Ис-	Индивидуальное выполнение заданий, групповой

	пользование документации Рекомендуемые источники из разделов 8,9 - 8.1; 8.2; 8.4; 8.3	разбор результатов выполнения заданий (30% времени на интерактивные технологии)
Тема 2. Основные синтаксические конструкции Python	Изучение базовых конструкций языка программирования Python Рекомендуемые источники из разделов 8,9 - 8.1; 8.2; 8.4; 8.3	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий (33% времени на интерактивные технологии)
Тема 3. Базовые технологии для анализа данных	Знакомство с информационными технологиями анализа данных Рекомендуемые источники из разделов 8,9 - 8.5; 8.8; 8.9; 8.10; 8.11; 8.12	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий (33% времени на интерактивные технологии)
Тема 4. Технологии работы со структурированными данными	Создание на языке Python приложений использующих универсальные форматы хранения структурированной информации Рекомендуемые источники из разделов 8,9 - 8.2; 8.4; 8.3; 8.5; 8.8; 8.9; 8.10; 8.11; 8.12	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий (25% времени на интерактивные технологии)
Тема 5. Технологии обработки данных	Знакомство с аналитическими инструментами на языке Python Рекомендуемые источники из разделов 8,9 - 8.2; 8.4; 8.3; 8.5; 8.8; 8.9; 8.10; 8.11; 8.12	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий (25% 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

Самостоятельная работа включает также подготовку к семинарским занятиям, написание контрольной работы.

Таблица 4

Наименование тем (разделов) дисциплины	Перечень вопросов, отводимых на самостоятельное освоение	Формы внеаудиторной самостоятельной работы
1.	Знакомство с современным стеком ИТ анализа данных	РЛ, РЭИ, РАП
2.	Основные синтаксические конструкции Python	РЛ, РЭИ, РАП
3.	Базовые технологии для анализа данных	РЛ, РЭИ, РАП
4.	Технологии работы со структурированными данными	РЛ, РЭИ, РАП
5.	Технологии продвинутого анализа данных	РЛ, РЭИ, РАП
ИТОГО		

* Сокращения в таблице: РЛ – работа с литературой; РЭИ – работа с электронными источниками; РАП – разработка алгоритмов и программ

6.2. Перечень вопросов, заданий, тем для подготовки к текущему контролю (согласно таблице 2)

Примерные темы контрольной работы:

Тематика внеаудиторной самостоятельной работы: контрольной работы (КР), соответствует содержанию дисциплины, и определяется преподавателем.

Примерная тема контрольного домашнего задания «Освоение синтаксических и семантических конструкций языка программирования Python».

Для выполнения домашнего контрольного задания необходимо повторить материалы лекций и семинаров. Результат контрольной работы должен в согласованные сроки быть передан преподавателю в электронной форме. Перед решением обязательно должно быть размещено условие задачи.

Тексты лекций, задания на самостоятельную работу, примеры решения типовых задач, а также вспомогательные материалы находятся на преподавательском диске, доступном студентам.

Для проведения тестирования используется комплекс тестовых заданий, разработанных для каждой темы. Результаты тестирования, устных опросов, контрольных работ и т.д. фиксируются в ведомостях преподавателя, которые находятся на преподавательском диске. Для каждой темы разработаны дополнительные задачи, решение которых в инициативном порядке могут выполнять студенты. Преподаватель за каждые N самостоятельно разработанных программ может выставить дополнительно M баллов.

Рабочая версия интегрированной среды программирования, используемая при разработке программ, а также соответствующая документация находятся в открытом доступе в сети Интернет и доступны для скачивания, инсталляции и использования в соответствии с принятой процедурой регистрации.

Примерный вариант заданий контрольной работы

1. Процедурное программирование

1.1. Строки

1.1.1. Инвертировать последовательность слов, разделенных запятыми.

Пример: строка 'SIX, SEVEN, EIGHT, NINE, TEN' будет преобразована в: 'TEN, NINE, EIGHT, SEVEN, SIX'.

1.1.2. На основе строки, представляющей из себя предложение, построить вложенный список, содержащий символы всех слов в предложении.

Пример: строка 'Eeny, meeny, miney, мое; Catch a tiger by his toe.' будет преобразована в: [['E', 'e', 'n', 'y'], ['m', 'e', 'e', 'n', 'y'], ['m', 'i', 'n', 'e', 'y'], ['m', 'o', 'e'], ['C', 'a', 't', 'c', 'h'], ['a'], ['t', 'i', 'g', 'e', 'r'], ['b', 'y'], ['h', 'i', 's'], ['t', 'o', 'e']]

1.1.3. 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'.

1.1.4. 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'

1.2. Генераторы

1.2.1. Используя генератор списков (и не используя код вне него) преобразовать строку по следующей логике: для каждого символа исходной строки создать в итоговом списке строку, содержащую копии символа в количестве, равном номеру символа в исходной строке. Пример: 'abcd' -> ['a', 'bb', 'ccc', 'dddd']

1.2.2. 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'}

1.2.3. Используя генератор словарей (и не используя код вне него) преобразовать словарь в котором ключами являются кортежи из целых чисел в словарь в котором ключом является среднее значение из чисел исходного ключа, значение оставить прежним.

Пример: {(2,4):'a', (1,1,1):'b', (2,3):'c'} -> {3.0:'a', 1.0:'b', 2.5:'c'}

1.2.4. Используя генератор списков (и не используя код вне него) преобразовать список кортежей в список кортежей по следующему правилу: если в кортеже четное количество элементов, то из него нужно удалить последний элемент. В остальных случаях кортежи оставить неизменными.

Пример: [(1,3,4), (2,1), (6,), (2,2,2,1)] -> [(1,3,4), (2,), (6,), (2,2,2,)]

1.2.5. Используя генератор списков (и не используя код вне него) преобразовать два списка (в первом содержатся целые числа, во втором строки, содержащие один символ) в словарь, в котором соответствующие друг другу пары значений из исходных списков преобразованы

в целочисленный ключ и строку состоящую из повторенных символов (количество повторений равно значению ключа).

Пример [2, 4, 1, 3], ['a', 'b', 'c', 'd'] -> {2:'aa', 4:'bbbb', 1:'c', 3:'ddd'}

1.2.6. Используя генератор словарей (и не используя код вне него) преобразовать словарь в котором ключами и значениями являются целые числа в список, в котором содержатся суммы исходных пар ключей и значений, причем, в список включаются только суммы, являющиеся четными числами.

Пример: {2:4, 3:2, 12:6, 5:4, 1:3} -> [6, 18, 4]

1.2.7. Используя генератор списков (и не используя код вне него) преобразовать список содержащий положительные целые числа в список, элементами которого являются списки с длиной равной соответствующему числу в первом списке. Содержимым вложенных списков являются последовательно идущие целые числа начиная с 1. Пример: [3, 1, 4] -> [[1, 2, 3], [1], [1, 2, 3, 4]]

1.2.8. Используя генератор списков (и не используя код вне него) преобразовать строку по следующей логике: для каждого символа исходной строки создать в итоговом списке строку, содержащую копии символа в количестве, равном номеру символа рассчитанному с конца исходной строки.

Пример: 'abcd' -> ['aaaa', 'bbb', 'cc', 'd']

2. Объектно-ориентированное программирование

2.1. Иерархия.

2.1.1. Создать иерархию классов для фруктов, продающихся в магазине.

Иерархия должна содержать не менее 5 классов, и не менее

3х уровней. Объекты должны содержать не менее 3х атрибутов и 2х методов (часть из которых должны быть перегружены). В конструкторах должны корректно использоваться конструкторы базовых классов.

Необходимо заполнить список представителями всех классов (всего не менее 10 объектов) и продемонстрировать работу полиморфизма.

2.1.2. Создать иерархию классов для фруктов, продающихся в магазине.

Иерархия должна содержать не менее 3 классов. Объекты должны содержать не

менее 4х атрибутов. Часть атрибутов должна быть защищена от изменения, а часть и от изменения, и от чтения. Необходимо заполнить список представителями всех классов (всего не менее 10 объектов) и продемонстрировать созданную защиту.

2.1.3. 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. Функции и функциональное программирование.

3.1. Функции

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

Пример: параметры: 1, 3, 2, 2 -> [1, 4, 6, 8] . Необязательный булевский параметр `mul` должен позволять заменять суммирование умножением.

Пример: параметры: 1, 3, 2, 2 -> [1, 3, 6, 12]

3.1.2. Реализовать функцию `gerl` , которая принимает на вход строку и набор заранее неизвестных параметров. Результатом функции является строка, в которой слова совпадающие с именами параметров заменены на значения параметров.

Пример: строка: 'replace my val abc', параметры `my='s1'`, `abc='fff'` -> Результат: 'replace s1 val fff'

3.1.3. Реализовать функцию `psort`, которая принимает на вход набор заранее неизвестных поименованных параметров. Функция возвращает список значений параметров отсортированный по именам параметров.

Пример: `psort(c=21, a=22, ac=17, b=16) -> [22, 17, 16, 21]`

3.1.4. Реализовать функцию `psort`, которая принимает на вход набор заранее неизвестных поименованных параметров. Функция возвращает список имен параметров, отсортированный по значениям параметров.

Пример: `psort(c=21, a=22, ac=17, b=16) -> [b, ac, c, a]`

3.1.5. Реализовать функцию `nam_par`, которая принимает на вход заранее неизвестное количество параметров и необязательный параметр `name` в который можно передать строку. Функция возвращает словарь в котором переданные параметры являются значениями, ключами для них являются соответствующие (сопоставленные по порядку следования) символы из строки `name`. Если строка `name` не задана, то значения присваиваются по порядку английского алфавита. Пример 1: `nam_par(7, 3, 1, 8, 10, 13, name='xyzafg') -> {'x':7, 'y':3, 'z':1, 'a':8, 'f':10, 'g':13}`

Пример 2: `nam_par(21, 'val', -3.5) -> {'a':21, 'b':'val', 'c':-3.5}`

3.1.6. Реализовать функцию `par_val`, которая принимает на вход заранее неизвестное количество именованных параметров (значения параметров - строки) и возвращает список имен параметров, которым соответствуют строки, содержащие более двух слов.

Пример: `par_val(pp='abba war', fan='oneword', zr='a x') -> [pp, zr]`

1.1. Рекурсии

1.1.1. Рекурсивно реализовать функцию `fib(n)` вычисляющую значение n -го числа Фибоначчи. Числа Фибоначчи — элементы числовой последовательности в которой каждое последующее число равно сумме двух предыдущих чисел (Числа Фибоначчи: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, ...).

1.1.2. Создать рекурсивную реализацию функции поиска элемента в отсортированном списке методом деления пополам (бинарного поиска) `sort_search(sorted_list, value, from, to)`. Аргументы функции: `sorted_list` —

отсортированный список, в котором проводится поиск; `value` – искомое значение; `from` – индекс элемента в списке, начиная с которого осуществляется поиск; `to` – индекс элемента в списке, до которого (не включительно) осуществляется поиск. Функция возвращает индекс первого вхождения `value` или `None`, в случае отсутствия элемента.

1.1.3. Создать рекурсивную реализацию функции поиска максимального числа в списке, содержащем целые числа. В качестве основного шага рекурсии использовать переход от поиска в данном списке к двум операциям поиска в списках вдвое меньшей длины.

Функция должна иметь вид `max_search(int_list, from, to)`. Аргументы функции: `int_list` – список, содержащий целые числа; `from` – индекс элемента в списке, начиная с которого осуществляется поиск; `to` – индекс элемента в списке, до которого (не включительно) осуществляется поиск. Функция возвращает значение максимального элемента.

1.2. Декораторы

1.2.1. Реализовать декоратор, который выводит на печать возвращаемые значения функции.

1.2.2. Реализовать декоратор с именем `not_none`, который генерирует исключительную ситуацию если декорируемая функция вернула значения `None`.

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

1.3. `map/filter/reduce`

1.3.1. При помощи механизма `map/filter/reduce` возвести в квадрат числа от 1 до 100, и рассчитать их сумму, не включая в сумму числа, кратные 10.

1.3.2. Дан список целых чисел. При помощи механизма `map/filter/reduce` рассчитать остаток от деления на 17 для каждого из чисел списка и получить произведение тех остатков, величина которых больше 7.

1.3.3. Дано предложение без знаков препинания. Превратить предложение в список слов. При помощи механизма `map/filter/reduce` отбросить у каждого

слова последнюю букву и склеить в одну строку те обрезанные слова, длина которых больше 5.

2. Структуры данных

2.1. Стеки, очереди

2.1.1. При помощи стека (можно использовать любую реализацию стека) проверить что в строке содержащей открывающие и закрывающие скобки трех типов: (), [], {}. Соблюдается корректный баланс и вложенность скобок.

2.1.2. Реализовать функцию `st_reverse(a_string)`, которая при помощи стека инвертирует строку (меняет порядок букв на обратный). Пример: `st_reverse('abcd') -> 'dcba'`

2.1.3. Реализовать функцию `list3_to2(list1, list2, list3)`, которая при помощи очереди превращает 3 списка одинаковой длины в 2 списка, длина которых не различается больше чем на 1, в полученных очередях должны чередоваться элементы из разных исходных списков и не должен нарушаться их порядок (т.е. если 'А' шло раньше 'В' в исходном списке, то 'В' не может идти раньше 'А' в итоговом списке).

3. Сортировки.

3.1. Простые сортировки

3.1.1. Реализовать алгоритм обменной сортировки.

3.1.2. Реализовать алгоритм сортировки выбором.

3.1.3. Реализовать алгоритм сортировки вставками.

3.2. Эффективные сортировки

3.2.1. Реализовать алгоритм сортировки Шелла.

3.2.2. Реализовать алгоритм быстрой сортировки.

3.2.3. Реализовать алгоритм сортировки слиянием.

Примеры теоретических заданий контрольной работы:

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

2. Концепция «функции – граждане первого класса» в языке программирования, поддержка этой концепции в Python. Специфика лямбда-функций в Python.
3. Глобальные и локальные переменные в функциях на примере Python. Побочные эффекты вызова функций и их последствия.
4. Вложенные функции и замыкания, специфика реализации в Python.
5. Функции высшего порядка и декораторы в Python.
6. Концепция map/filter/reduce. Работа map() в различных вариациях в Python.
7. Концепция map/filter/reduce. Работа filter() в Python. Использование модуля operator при работе с filter.
8. Концепция map/filter/reduce. Работа reduce() в Python.
9. Использование reduce с начальным значением, имеющим тип, отличный от возвращаемого редуцирующей функцией.
10. Итераторы в Python: встроенные итераторы, создание собственных итераторов, типичные способы обхода итераторов и принцип их работы.
11. Встроенные функции для работы с итераторами и возможности модуля itertools.
12. Функции генераторы и выражения генераторы: создание и применение в Python.
13. Специфика массивов, как структур данных. Динамические массивы – специфика работы, сложность операций.
14. Специфика работы с array в Python.
15. Абстрактная структура данных стек: базовые и расширенные операции, их сложность. Реализация стека в Python.
16. Абстрактная структура данных очередь: базовые и расширенные операции, их сложность. Реализация очереди в Python.
17. Специфика реализации и скорости основных операций в очереди на базе массива и связанного списка.
18. Связанные списки: однонаправленные и двунаправленные.
19. Сравнение скорости выполнения основных операций в связанных списках и в динамическом массиве.
20. Алгоритм сортировки выбором, сложность сортировки и возможности по ее улучшению.
21. Алгоритм сортировки вставками, его сложность. Алгоритм быстрого поиска в отсортированном массиве.
22. Сложность поиска в отсортированном и не отсортированном массиве.
23. Алгоритм сортировки Шелла, сложность сортировки и возможности по ее улучшению.

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

25. Алгоритм сортировки слиянием, сложность сортировки.

Оценка знаний по 100-балльной шкале проводится в соответствии с нормативными документами Финуниверситета.

Ориентировочное распределение максимальных баллов по видам работы:

Таблица 7

№ п/п	Вид отчетности	Баллы всего
1.	Текущий контроль в т.ч:	40
	доклад, презентация, участие в групповой дискуссии, решение задач	30
	подготовка контрольной работы	10
2.	Зачет	60
3.	Итого	100

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

7.1. Перечень компетенций, формируемых в процессе освоения дисциплины

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

7.2. Описание показателей и критериев оценивания компетенций, описание шкал оценивания.

Зачет по дисциплине выставляется студенту при условии сформированности по каждой компетенции как минимум порогового уровня.

Таблица 8.

Результаты обучения по дисциплине (модулю)	Шкала и критерии оценки результатов			
	Высокий уровень (отлично)	Продвинутый уровень (хорошо)	Пороговый уровень (удовлетворительно)	Пороговый уровень не достигнут (неудовлетворительно)
ПКП-5 Способность к использованию специальных программных продуктов, применяемых для выполнения бухгалтерско-аналитических и контрольных функций в экономическом субъекте 1. <i>Использует специальные программные продукты для выполнения бухгалтерско-аналитических и контрольных функций в экономическом субъекте.</i> 2. <i>Демонстрирует владение специальными программными продуктами, применяемыми для выполнения бухгалтерско-аналитических и контрольных функций в экономическом субъекте.</i>				
знать: особенности функционирования специальных программных продуктов и их применения для выполнения бухгалтерско-аналитических и контрольных функций в экономическом субъекте	Знает методы и инструменты специальных программных продуктов и особенности их применения для выполнения бухгалтерско-аналитических и контрольных функций в экономическом субъекте	Знает методы и инструменты специальных программных продуктов и основные аспекты их применения для выполнения бухгалтерско-аналитических и контрольных функций в экономическом субъекте	Частично знает методы и инструменты специальных программных продуктов и их применения для выполнения бухгалтерско-аналитических и контрольных функций в экономическом субъекте	Не знает методы и инструменты специальных программных продуктов и особенности их применения для выполнения бухгалтерско-аналитических и контрольных функций в экономическом субъекте;

уметь: применять специальные программные продукты для выполнения бухгалтерско-аналитических и контрольных функций в экономическом субъекте	Умеет работать в среде специализированных компьютерных программ при решении нестандартных профессиональных задач	Умеет работать в среде специализированных компьютерных программ при решении стандартных профессиональных задач	Плохо умеет работать в среде специализированных компьютерных программ при решении стандартных профессиональных задач	Не умеет работать в среде специализированных компьютерных программ при решении стандартных профессиональных задач
УК – 15 Способность релевантно решаемым задачам использовать информационные ресурсы и информационно-коммуникационные технологии для достижения целей, связанных с профессиональной деятельностью, обучением, участием в жизни общества и других сферах жизни <i>1. Самостоятельно выбирает и использует цифровые средства общения, осуществляет поиск и/или создание контента в соответствии с целью взаимодействия, в том числе для организации совместной деятельности.</i>				
Знать: основные источники, а также, современные средства и инструменты поиска контента Уметь: выбирать формы и методы подачи контента для организации совместной работы	Знает основные источники, а также, современные средства и инструменты поиска контента Умеет выбирать формы и методы подачи контента для организации совместной работы; решать нетипичные задачи	Знает основные источники, а также, современные средства и инструменты поиска контента Умеет выбирать формы и методы подачи контента для организации совместной работы	Частично знает основные источники, а также, современные средства и инструменты поиска контента Плохо умеет выбирать формы и методы подачи контента для организации совместной работы	Не знает основные источники, а также, современные средства и инструменты поиска контента Не умеет выбирать формы и методы подачи контента для организации совместной работы
<i>2. Владеет навыками организации взаимодействия и коммуникации с помощью информационных систем и/или цифровых сервисов и технологий. Осуществляет подбор и применение различных информационно-коммуникационных средств для решения образовательных и профессиональных задач.</i>				
Знать: современные средства организации совместной удаленной работы Уметь: выбирать средства	Знает современные средства организации совместной удаленной работы Умеет выбирать средства	Знает современные средства организации совместной удаленной работы Умеет выбирать средства организации совместной	Частично знает современные средства организации совместной удаленной работы Плохо умеет выбирать средства организации	Не знает современные средства организации совместной удаленной работы Не умеет выбирать средства организации

организации совместной деятельности	организации совместной деятельности; решать нетипичные задачи	деятельности	совместной деятельности	совместной деятельности
-------------------------------------	--	--------------	-------------------------	-------------------------

Текущий контроль осуществляется в ходе учебного процесса и консультирования студентов, а также на основе результатов выполнения самостоятельных работ. На контроль выносятся:

- вопросы, вынесенные на самостоятельное изучение и осмысление,
- вопросы и тесты на освоение минимально необходимого материала,
- выполнение и защита лабораторных и контрольных работ.
- **промежуточная аттестация проводится в форме зачета.**

Оценка знаний по 100-балльной шкале проводится в соответствии с нормативными документами Финуниверситета.

Таблица 9.

Требования к результатам освоения дисциплины	Оценка или зачет	Баллы (рейтинговая оценка)
Высокий уровень – глубокое усвоение программного материала, логически стройное его изложение, дискуссионность данной проблематики, умение связать теорию с возможностями ее применения на практике, свободное решение задач и обоснование принятого решения, владение методологией и методиками исследования, методами моделирования; Продвинутый уровень – твердые знания программного материала, допустимы несущественные неточности в ответе на вопрос, правильное применение теоретических положений при решении вопросов и задач, умение выбирать конкретные информационные технологии и методы решения сложных задач, используя методы сбора, расчета, анализа, классификации, интерпретации данных; Пороговый уровень - знание только основного материала, допустимы неточности в ответе на вопросы, нарушение логической последовательности в изложении программного материала, умение решать простые задачи на основе базовых знаний и заданных алгоритмов действий, испытывать затруднения при решении практических задач;	зачет	50 баллов и выше
Недостаточный уровень - незнание значительной части программного материала, неумение даже с помощью преподавателя сформулировать правильные ответы на задаваемые вопросы, невыполнение практических заданий.	незачет	49 баллов и ниже

7.3. Типовые контрольные задания и иные материалы, необходимые для оценки индикаторов достижения компетенций, умений и знаний

Промежуточная аттестация проводится в форме зачета.

Компетенция	Индикаторы достижения компетенции.	Вопросы к зачету, Тестовые задания, Практико-ориентированные задания
ПКП-5 Способность к использованию специальных программных продуктов, применяемых для выполнения бухгалтерско-аналитических и контрольных функций в экономическом субъекте	1. Использует специальные программные продукты для выполнения бухгалтерско-аналитических и контрольных функций в экономическом субъекте.	Знать: Вопросы к зачету: 1. Парадигмы программирования их суть и сильные стороны. Типичные представители различных парадигм, применение различных парадигм в Python. 2. Специфика типизации в языках программирования (различные аспекты типизации). Реализация типизации в Python. 3. Именованное переменных и других объектов в Python (правила и соглашения). Числовые типы: литералы, объявление и операции. 4. Присвоение по ссылке и по значению. Специфика создания объектов и присвоения в Python, особенности работы с объектами целочисленного типа. 5. Разница между копированием и присвоением. Проблема утечки динамической памяти, сборка мусора. Копирование, присвоение и стратегия управления динамической памятью в Python. 6. Булевский тип, сравнения и условные операторы в Python. 7. Циклы в Python, работа и устройство цикла for, типичное применение range и enumerate в цикле for. 8. Строки в Python. Принципы работы и основные операторы и функции. 9. Списки в Python. Различные способы создания и копирования списков в Python. Обход списка и поиск элементов в списке. 10. Списки в Python. Обращение к элементам списка и создание срезов. Стандартные агрегирующие функции, работающие со списками. 11. Списки в Python. Ключевые операции, проводящие к изменению списка и порождающие измененные списки. 12. Словари в Python. Основные способы создания, получения и изменения значений. Обход словарей.

		13. Преобразование между словарями и списками в Python. Операции с представлениями словарей. 14. Операции со словарями, учитывающие возможное отсутствие ключа. Операции многоэлементного изменения словарей. Операции поэлементного извлечения из словаря и их использование. 15. Множества в Python. Основные способы создания, получения и изменения значений. Обход множеств. 16. Выполнение основных операций с парой множеств в Python. 17. Кортежи в Python. Отличия кортежей от списков. Распаковка и частичная распаковка кортежей. 18. Выражения генераторы и генераторы списков в Python. Использование условий в генераторах. 19. Генераторы множеств и словарей в Python. Использование условий в генераторах. 20. Функции стандартной библиотеки для работы с контейнерами. Уметь: <u>Тестовые задания</u> 1. Сколько библиотек можно импортировать в один проект? 1. Не более 10 2. Не более 23 3. Не более 5 4. Не более 3 5. Неограниченное количество 2. Что покажет этот код? <pre>for j in 'Hi! I'm mister Robert': if j == '\': print("Найдено") break else: print ("Готово")</pre> 1. "Готово" 2. "Найдено" 3. Ошибку в коде
--	--	---

	2. Демонстрирует владение специальными программными продуктами, применяемыми для выполнения бухгалтерско-аналитических и контрольных функций в	4. "Найдено" и "Готово" 3. Как получить данные от пользователя? 1. Использовать метод read() 2. Использовать метод cin() 3. Использовать метод readLine() 4. Использовать метод get() 5. Использовать метод input() <u>Практико-ориентированные задания</u> Задание 1 Есть список a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]. Выведите все элементы, которые меньше 5. Задание 2 Даны списки: a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]; b = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13]. Нужно вернуть список, который состоит из элементов, общих для этих двух списков. Задание 3 Отсортируйте словарь по значению в порядке возрастания и убывания. Знать: <u>Вопросы к зачету:</u> 1. Объявление и вызов функции в Python. Параметры функции со значением по умолчанию и комментирование функции. 2. Получение информации о функции. Способы передачи параметров при вызове функции. 3. Передача переменного количества параметров (именованных и не именованных) в функции Python. Вызов функции с позиционными параметрами, находящимися в списке, и именованными параметрами, находящимися в словаре. 4. Анонимные функции в Python их возможности и ограничения.
--	---	--

	экономическом субъекте.	5. Типичные сценарии использования анонимных функций. 6. Синтаксис и семантика обработки исключительных ситуаций в Python. 7. Создание пользовательских исключений и инструкция assert. 8. Базовые операции для работы с файлами в Python. 9. Использование инструкции with ... as на примере работы с файлами. 10. Использование модулей pickle и shelve для сохранения объектов в файл и их восстановления. 11. Модули в Python и их отличие от скриптов Python. Варианты синтаксиса импорта модуля и объектов модуля. Применение импортированных объектов. 12. Порядок поиска модулей и специфика их загрузки. Загрузка модулей из глобального репозитория. 13. Импорт кода из пакетов. Организация пакетов в Python. 14. Концепция класса и объекта. Принципы и механизмы ООП. 15. Объявление класса, конструктор, создание объектов и одиночное наследование в Python. 16. Управление доступом к атрибутам класса в Python. 17. Полиморфизм и утиная типизация и проверка принадлежности объекта к классу в языке Python. 18. Методы классов и статические переменные и методы в Python. 19. Интроспекция и динамические операции с объектами в Python. 20. Специальные методы для использования пользовательских классов со стандартными операторами и функциями. Уметь: <u>Тестовые задания</u> 1. Что будет показано в результате? <pre>name = "John" print('Hi, %s' % name)</pre> 1. "Hi, John" 2. Ошибка 3. "Hi, " 4. "Hi, name"
--	-------------------------	--

		2. Что покажет этот код? <pre>for i in range(5): if i % 2 == 0: continue print(i)</pre> 1. Ошибку, так как i не присвоена 2. Числа: 1, 3 и 5 3. Числа: 0, 2 и 4 4. Числа: 1 и 3 5. Ошибку из-за неверного вывода 3. <u>Какие ошибки допущены в коде ниже?</u> <pre>def factorial(n): if n == 0: return 1 else: return n * factorial(n - 1) print(factorial(5))</pre> 1. В коде нет никаких ошибок 2. Функция всегда будет возвращать 1 3. Необходимо указать тип возвращаемого значения 4. Функция не может вызывать сама себя <u>Практико-ориентированные задания</u> Задание 1 Напишите программу для слияния нескольких словарей в один. Задание 2 Найдите три ключа с самыми высокими значениями в словаре my_dict = {'a':500, 'b':5874, 'c': 560, 'd':400, 'e':5874, 'f': 20}.
--	--	---

УК – 15 Способность релевантно решать задачи, использовать информационные ресурсы и информационно-коммуникационные технологии для достижения целей, связанных с профессиональной деятельностью, обучением, участием в жизни общества и других сферах жизни	<i>1. Самостоятельно выбирает и использует цифровые средства общения, осуществляет поиск и/или создание контента в соответствии с целью взаимодействия, в том числе для организации совместной деятельности.</i>	<u>Вопросы к зачету</u> 1. Модули. Инструкции import и from. 2. Базовые принципы объектно-ориентированного программирования. 3. Класс, метод класса, атрибут класса. Определение класса и создание экземпляра класса. <u>Практико-ориентированные задания</u> Задание 1 Определите дату вступления № 44-ФЗ в силу. Задание 2 Объедините контент из двух разных текстовых редакторов. Задайте единые стандарты форматирования по ГОСТ 7.32 – 2017..
	<i>2. Владеет навыками организации взаимодействия и коммуникации с помощью информационных систем и/или цифровых сервисов и технологий. Осуществляет подбор и применение различных информационно-коммуникационных средств для решения образовательных и профессиональных задач.</i>	<u>Практико-ориентированные задания</u> Задание 1 Организуите видеоконференцию Задание 2 Предоставьте доступ к редактированию текстового файла двум коллегам на Яндекс-диске.

Перечень вопросов к зачету:

1. Встроенные числовые типы языка Python.
2. Списки. Создание, основные операции.
3. Основные методы списка.
4. Кортежи. Создание, основные методы и операции.
5. Словари. Создание, основные операции.
6. Методы для работы со словарями.
7. Множества. Создание, основные методы и операции.
8. Переменные. Правила именования переменных.
9. Динамическая типизация.
10. Операторы сравнения и логические операторы.
11. Инструкция if...else.
12. Инструкция цикла while.
13. Инструкция цикла for.
14. Создание и вызов функции.
15. Передача аргументов функцию.
16. Функции-генераторы.
17. Лямбда-функции.
18. Модули. Инструкции import и from.
19. Базовые принципы объектно-ориентированного программирования.
20. Класс, метод класса, атрибут класса. Определение класса и создание экземпляра класса.
21. Конструктор и деструктор.
22. Наследование.
23. Абстрактные методы класса.
24. Статические методы класса.
25. Свойства класса.
26. Исключения. Обработка исключений.
27. Пользовательские исключения.
28. Событие. Обработчик события. Цикл обработки событий.

29. Элемент Кнопка. Создание и настройка.
30. Элемент Кнопка. Создание обработчика события.
31. Элементы Надпись и Текстовое поле. Создание и настройка. Метод get().
32. Элемент Флажок. Создание, настройка, получение статуса флажка.
33. Элемент переключатель. Создание, настройка, доступ к значению.
34. Классы date, time и datetime.
35. Возможности библиотеки SymPy.
36. Возможности библиотеки NumPy.

7.4. Методические материалы, определяющие процедуру оценивания знаний, умений, владений

Соответствующие приказы, распоряжения ректората, директора филиала о контроле знаний студентов.

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

Основная литература:

1. Колдаев, В. Д. Основы алгоритмизации и программирования: учебное пособие / В.Д. Колдаев; под ред. проф. Л.Г. Гагариной. — Москва: ФОРУМ: ИНФРА-М, 2022. — 414 с. — (Среднее профессиональное образование). - ISBN 978-5-8199-0733-7. - Текст: электронный. - URL: <https://znanium.com/catalog/product/1735805>— Режим доступа: по подписке.
2. Колдаев, В. Д. Структуры и алгоритмы обработки данных: учебное пособие / В.Д. Колдаев. — Москва : РИОР: ИНФРА-М, 2021. — 296 с. — (Высшее образование: Бакалавриат). — www.dx.doi.org/10.12737/2833. - ISBN 978-5-369-01264-2. - Текст: электронный. - URL: <https://znanium.com/catalog/product/1230215>— Режим доступа: по подписке.

Дополнительная литература

3. Прохоренок Н.А. Python. Самое необходимое / Н.А. Прохоренок. — Санкт-Петербург: БХВ-Петербург, 2011. — 416 с.

4. Прохоренок Н.А. Python и PyQt. Разработка приложений / Н.А. Прохоренок. – Санкт-Петербург: БХВ-Петербург, 2012.
5. Маккинли У. Python и анализ данных / У. Маккинли; пер. с англ. А.А. Слинкин. – Москва: ДМК Пресс, 2015.
6. Прохоренок Н.А. Python 3. Самое необходимое / Н.А. Прохоренок, В.А. Дронов. – Санкт-Петербург: БХВ-Петербург, 2016. – 464 с.
7. Доусон М. Програмируем на Python / М. Доусон; пер. с англ. – Санкт-Петербург: Питер, 2015. – 416 с.
8. Лутц М. Изучаем Python, / М. Лутц; пер. с англ. – 4-е издание. – Санкт-Петербург: Символ-Плюс, 2011. – 1280 с.
9. Лутц М. Программирование на Python, том I / М. Лутц; пер. с англ.– 4-е издание. – Санкт-Петербург: Символ-Плюс, 2011. – 992 с.
10. Лутц М. Программирование на Python, том II / М. Лутц; пер. с англ.– 4-е издание. – Санкт-Петербург: Символ-Плюс, 2011. – 992 с.
11. SciPy // <http://docs.scipy.org/doc/scipy/reference/>
12. NumPy User Guide // <http://docs.scipy.org/doc/numpy/user/index.html>
13. The Python Standard Library //

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

- 9.1. PyLru 1.0.9 [Электронный ресурс]: сайт. – Режим доступа: <https://pypi.python.org/pypi/pylru>
- 9.2. Python Data Analysis Library [Электронный ресурс]: сайт. – Режим доступа: <http://pandas.pydata.org/>
- 9.3. Python Documentation [Электронный ресурс]: сайт. – Режим доступа: <http://python.org/doc/>
- 9.4. Python Standard Library [Электронный ресурс]: сайт. – Режим доступа: <https://docs.python.org/2/library/>
- 9.5. Scikit-learn Machine Learning in Python [Электронный ресурс]: сайт. – Режим доступа: <http://scikit-learn.org>
- 9.6. Официальный сайт продукта <https://www.python.org/>

- 9.7. Информационно-образовательный портал Финансового университета при Правительстве Российской Федерации <http://portal.ufrf.ru/>
- 9.8. Каталог курсов Интернет Университета Информационных Технологий <http://www.intuit.ru/>
10. Электронная библиотека Финансового университета (ЭБ) <http://elib.fa.ru/>
(<http://library.fa.ru/files/elibfa.pdf>)
11. Электронно-библиотечная система Znanium <http://www.znaniy.com>

10. Методические указания для обучающихся по освоению дисциплины

Изучение дисциплины требует систематического и последовательного накопления знаний, следовательно, пропуски отдельных тем не позволяют глубоко освоить предмет.

Студентам необходимо ознакомиться:

– с содержанием рабочей программы дисциплины (далее - РПД), с целями и задачами дисциплины, ее связями с другими дисциплинами образовательной программы, методическими разработками по данной дисциплине, имеющимися на образовательном портале и сайте кафедры, с графиком консультаций преподавателей данной кафедры.

При подготовке к лекционным занятиям студентам необходимо:

– перед каждой лекцией просматривать рабочую программу дисциплины, что позволит сэкономить время на записывание темы лекции, ее основных вопросов, рекомендуемой литературы;

– на отдельные лекции приносить соответствующий материал на бумажных носителях, представленный лектором на портале или присланный на «электронный почтовый ящик группы» (таблицы, графики, схемы). Данный материал будет охарактеризован, прокомментирован, дополнен непосредственно на лекции;

– перед очередной лекцией необходимо просмотреть по конспекту материал предыдущей лекции. При затруднениях в восприятии материала следует обратиться к основным литературным источникам. Если разобраться в материале опять не удалось, то обратитесь к лектору (по графику его консультаций) или к

преподавателю на практических занятиях. Не оставляйте «белых пятен» в освоении материала.

При подготовке к практическим занятиям студентам следует:

- приносить с собой рекомендованную преподавателем литературу к конкретному занятию;
- до очередного практического занятия по рекомендованным литературным источникам проработать теоретический материал, соответствующей темы занятия;
- в начале занятий задать преподавателю вопросы по материалу, вызвавшему затруднения в его понимании и освоении при решении задач, заданных для самостоятельного решения;
- на занятии доводить каждую задачу до окончательного решения, демонстрировать понимание проведенных расчетов (анализов, ситуаций), в случае затруднений обращаться к преподавателю.

Студентам, пропустившим занятия (независимо от причин) и не имеющим выполненного практического задания, рекомендуется не позже чем в 2-недельный срок явиться на консультацию к преподавателю и отчитаться по теме, рассматриваемой на занятии. Студенты, не отчитавшиеся по каждой не проработанной ими на занятиях теме к началу зачетной сессии, упускают возможность получить положенные баллы за работу в соответствующем семестре.

Самостоятельная работа студентов включает в себя выполнение различного рода заданий, которые ориентированы на более глубокое усвоение материала изучаемой дисциплины. По каждой теме учебной дисциплины студентам предлагается перечень заданий для самостоятельной работы.

К выполнению заданий для самостоятельной работы предъявляются следующие требования: задания должны исполняться самостоятельно и представляться в установленный срок, а также соответствовать установленным требованиям по оформлению.

Студентам следует при подготовке к экзамену параллельно прорабатывать соответствующие теоретические и практические разделы дисциплины, фиксируя неясные моменты для их обсуждения на плановой консультации.

Любая форма самостоятельной работы студента (подготовка к практическому занятию, выполнение самостоятельной, контрольной работ начинается с изучения соответствующей литературы как в библиотеке, так и дома).

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

Комплект лицензионного программного обеспечения:

1. Astra Linux Special Edition

Современные профессиональные базы данных и информационные справочные системы:

1. Справочная правовая система «КонсультантПлюс» (<http://www.consultant.ru>).
2. Информационно-образовательный портал Финансового университета. - <https://org.fa.ru>, <http://campus.fa.ru>

Сертифицированные программные и аппаратные средства защиты информации – не предусмотрено.

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

Освоение дисциплины предполагает использование учебной аудитории для проведения лекционных и семинарских занятий с необходимыми техническими средствами (компьютер, проектор, интерактивная доска, презентации в системе Power Point, банк данных видео по курсу), специально оборудованные аудитории и компьютерные классы с локальной сетью и выходом в Интернет, а также с предустановленными системами.