

Федеральное государственное образовательное бюджетное учреждение
высшего образования

**«ФИНАНСОВЫЙ УНИВЕРСИТЕТ ПРИ ПРАВИТЕЛЬСТВЕ
РОССИЙСКОЙ ФЕДЕРАЦИИ»**

(Финансовый университет)

Краснодарский филиал Финуниверситета

Кафедра «Математика и информатика»

СОГЛАСОВАНО

ООО «Портал-Юг»
Генеральный директор



Е.В. Мостовой

«20» февраля 2024 г.

УТВЕРЖДАЮ

Краснодарский филиал
Финуниверситета

Директор

Э.В.Соболев

«20» февраля 2024 г.

Лузин Алексей Иванович

СОВРЕМЕННЫЕ ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ

Рабочая программа дисциплины

для студентов, обучающихся по направлениям подготовки
38.03.05 «Бизнес-информатика» профиль «Бизнес аналитика»
(очно-заочная форма обучения)

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

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

Краснодар 2024

УДК 004.434:5
ББК 32.973
Л83

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

Лузин А.С. Рабочая программа дисциплины для обучающихся по направлению 38.03.05 «Бизнес-информатика» – Краснодар: Краснодарский филиал Финуниверситета, кафедра «Математика и информатика», 2024 г.

Дисциплина современные технологии программирования относится к циклу математики и информатики по направлению подготовки 38.03.05 «Бизнес-информатика», профиль «ИТ-менеджмент в бизнесе».

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

Рабочая программа дисциплины

СОВРЕМЕННЫЕ ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ
(учебно-методический семинар)

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

Усл. п.л. 2,0. Изд. № _от.

Тираж 100 экз.

Заказ № .

Отпечатано в Краснодарском филиале Финуниверситета

Содержание

1. Наименование дисциплины	4
2. Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине	4
3. Место дисциплины в структуре образовательной программы	6
4. Объем дисциплины в зачетных единицах и в академических часах с выделением объема аудиторной (лекции, семинары) и самостоятельной работы обучающихся	6
5. Содержание дисциплины, структурированное по темам (разделам) дисциплины с указанием их объемов (в академических часах) и видов учебных занятий	7
5.1 Содержание дисциплины	7
5.2 Учебно-тематический план	8
5.3 Содержание семинаров, практических занятий	9
6. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине	10
6.1 Перечень вопросов, отводимых на самостоятельное освоение дисциплины, формы внеаудиторной самостоятельной работы	10
6.2 Перечень вопросов, заданий, тем для подготовки к текущему контролю ...	12
7. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине	18
7.1. Описание показателей и критериев оценивания компетенций	18
7.2. Вопросы для оценки знаний и умений, характеризующих формирование компетенций	26
7.3 Практико-ориентированные задания	27
7.4 Тесты	31
8. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины	33
9. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины	34
10. Методические указания для обучающихся по освоению дисциплины	34
Источники нумеруются арабскими цифрами без точки и печатаются с абзацного отступа	36
11. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень необходимого программного обеспечения и информационных справочных систем	38
11.1 Комплект лицензионного программного обеспечения	38
11.2 Современные профессиональные базы данных и информационные справочные системы	38
11.3 Сертифицированные программные и аппаратные средства защиты информации: не предусмотрены	38
12. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине	39

1. Наименование дисциплины

Дисциплина Б1.В.02.08 «Современные технологии программирования»

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

Дисциплина Б1.В.02.08 «Современные технологии программирования» обеспечивает формирования следующих компетенций направления 38.03.05 «Бизнес-информатика»:

Таблица 1 – Компетенции, формируемые в результате изучения дисциплины «Современные технологии программирования» направления подготовки 38.03.05 «Бизнес-информатика»

Код компетенции	Наименование компетенции	Индикаторы достижения компетенции	Результаты обучения (умения и знания), соотнесенные с 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.	Знать: основные методы и средства получения, представления, хранения и обработки данных на базе современных технологий прикладного программирования и баз данных. Уметь: применять основные методы и средства получения, представления, хранения и обработки данных на базе современных технологий прикладного программирования и баз данных
		2. Демонстрирует владение профессиональным и пакетами прикладных программ.	Знать: принципы использования профессиональных пакетов прикладных программ на базе современных технологий прикладного программирования и обработки данных. Уметь: использовать профессиональные пакеты прикладных программ на базе современных технологий прикладного программирования и обработки данных.
		3. Выбирает необходимое прикладное программное обеспечение в зависимости от решаемой задачи	Знать: критерии выбора прикладного программного обеспечения на базе современных технологий прикладного программирования и обработки данных в зависимости от решаемой задачи Уметь: выбрать прикладное программное обеспечение на базе современных технологий прикладного программирования и обработки данных в зависимости от решаемой задачи
		4. Использует прикладное программное обеспечение для решения	Знать: принципы и процедуры использования прикладного программного обеспечения для решения конкретных прикладных задач на базе современных технологий прикладного программирования и обработки данных.

		конкретных прикладных задач	Уметь: применить прикладное программное обеспечение для решения конкретных прикладных задач с использованием современных технологий прикладного программирования и обработки данных
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.	Знать: состав и структуру требуемых данных, методы их сбора, обработки и интерпретации Уметь: четко описывать состав и структуру требуемых данных и информации, грамотно реализовывать процессы их сбора, обработки и интерпретации
		2. Обосновывает сущность происходящего, выявляет закономерности, понимает природу вариабельности	Знать: Теоретические основы понимания природы вариабельности Уметь: обосновывать сущность происходящего, выявлять закономерности, понимать природу вариабельности
		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	Знать: теоретические основы методов оценки результатов классификации, прикладное назначение классификационных групп. Уметь: формулировать признак классификации, выделять соответствующие ему группы однородных "объектов", идентифицировать общие свойства элементов этих групп, оценивать полноту результатов классификации, показывать прикладное назначение классификационных групп.
		4. Грамотно, логично, аргументировано формирует собственные суждения и оценки. Отличает факты от мнений, интерпретаций, оценок и т.д. в рассуждениях других участников деятельности	Знать: основы грамотной, логичной, аргументированной формулировки собственных суждений и оценки. Уметь: грамотно, логично, аргументировано формировать собственные суждения и оценки. Отличать факты от мнений, интерпретаций, оценок и т.д. в рассуждениях других участников деятельности

		5. Аргументированно и логично представляет свою точку зрения посредством и на основе системного описан ия	Знать: теоретические основы системного описания Уметь: аргументированно и логично представлять свою точку зрения посредством и на основе системного описания
--	--	---	---

3. Место дисциплины в структуре образовательной программы

Дисциплина Б1.В.02.08 «Современные технологии программирования» является дисциплиной модуля предпрофильного профессионального цикла подготовки бакалавров по направлению 38.03.05 «Бизнес-информатика».

Программа изучения дисциплины составлена с учетом требований, установленных соответствующим ОС ВО Финуниверситета. Изучение дисциплины базируется на знаниях, приобретенных студентами направления 38.03.05 «Бизнес-информатика» в процессе предшествующего освоения дисциплин «Компьютерный практикум» и «Анализ данных».

В свою очередь, изучение дисциплины «Современные технологии программирования» позволит конкретизировать полученные знания, умения, навыки применительно к обработке и анализу экономических данных.

Знания и навыки, полученные в процессе изучения дисциплины «Современные технологии программирования» могут быть использованы студентами при написании выпускной квалификационной (бакалаврской) работы, в процессе решения круга задач профессиональной деятельности в дальнейшем. Является завершающей дисциплиной модуля.

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

Для направления подготовки 38.03.05 «Бизнес-информатика», профиль «Бизнес-аналитика» по очно-заочной форме обучения общая трудоёмкость дисциплины составляет 3 зач. ед., 108 часов (таблица 3).

Таблица 3 – Трудоёмкость дисциплины «Современные технологии программирования»

Вид учебной работы	Направление подготовки 38.03.05 «Бизнес-информатика»	
	Очно-зочная форма обучения (1 курс)	
	всего	Сем 3
Общая трудоёмкость дисциплины	3/108	3/108
Контактная работа -	20	20

<i>Аудиторные занятия</i>		
<i>Лекции</i>	8	8
Семинары, практические занятия	12	12
<i>Самостоятельная работа</i>	88	88
Вид текущего контроля	контрольная работа	контрольная работа
Вид промежуточной аттестации	Зачет	Зачет

Вид текущего контроля – 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 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

Темы дисциплины и виды занятий для направления подготовки 38.03.05 «Бизнес-информатика», представлены в таблице 4

Таблица 4 - Распределение бюджета времени при изучении дисциплины «Современные технологии программирования» для направления подготовки 38.03.05 «Бизнес-информатика», профиль «Бизнес-аналитика»

№	Наименование темы (раздела) дисциплины	Трудоемкость в часах	Формы текущего контроля
---	--	----------------------	-------------------------------

								успеваемости
		Всего	Аудиторная работа				Самостоятельная работа	
			Общая	Лекции	Практические	В интерактивных формах		
1.	Введение в программирование на языке Python	22	7	4	3	3	15	Самостоятельные работы. Участие в решении задач на практических занятиях. Контрольная работа.
2.	Основные синтаксические конструкции Python	22	7	3	4	4	15	
3.	Базовые технологии для анализа данных	22	7	3	4	4	15	
4.	Технологии работы со структурированным и данными	21	7	3	4	4	14	
5.	Технологии обработки данных	21	6	3	3	3	15	
	Контроль							
	Итого:	108	34	16	18	18	74	

5.3 Содержание семинаров, практических занятий

Цель практических занятий по дисциплине «Современные технологии программирования» – закрепление теоретических знаний, освоение методов финансовых вычислений, формирование навыков проведения расчетов, контроль выполнения заданий для самостоятельной работы. Занятия проводятся в активной и интерактивной формах с привлечением всех студентов к обсуждаемым вопросам, выбору оптимальных способов решения практических задач, что способствует профессиональному развитию личности будущего бакалавра. Содержание практических занятий представлено в таблице 6.

Таблица 6 – Содержание семинаров, практических занятий по дисциплине «Современные технологии программирования\» для направления подготовки 38.03.05 «Бизнес-информатика»

Наименование тем (разделов) дисциплины	Перечень вопросов для обсуждения на семинарских, практических занятиях, рекомендуемые источники из разделов 8,9	Формы проведения занятий
Введение в программирование на языке Python	Изучение технологического стека анализа данных, построенного на базе языка программирования Python. Рекомендуемые источники: 8.1 - 8.4;	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий

Основные синтаксические конструкции Python	Изучение базовых конструкций языка программирования Python. Рекомендуемые источники: 8.1 - 8.4;	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий
Базовые технологии для анализа данных	Знакомство с информационными технологиями анализа данных Python. Рекомендуемые источники: 8.5; 8.8 - 8.12	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий
Технологии работы со структурированными данными	Изучение примеров работы с форматами CSV, XML, XHTML, HTML, JSON при помощи библиотек на языке Python, проектирование собственного приложения, работающего с одним из данных форматов. Рекомендуемые источники: 8.2 - 8.5; 8.8 - 8.12	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий
Технологии обработки данных	Изучение примеров построения аналитических инструментов на языке Python, проектирование инструментария анализа данных собственного приложения. Рекомендуемые источники: 8.2 - 8.5; 8.8 - 8.12	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий

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

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

Основная цель самостоятельной работы студента при изучении дисциплины «Современные технологии программирования» — закрепить теоретические знания, полученные в ходе лекционных занятий, глубоко изучить, используя рекомендованную литературу, а также лекции по курсу, основные теоретические аспекты дисциплины, связанные с методами ценообразования.

Самостоятельная работа студента в процессе изучения дисциплины включает:

- освоение рекомендованной преподавателем по данной дисциплине основной и дополнительной учебной литературы;
- изучение корпоративных образовательных ресурсов (электронные учебники, электронные библиотеки, электронные видеокурсы и др.);
- выполнение домашних заданий в виде решения отдельных задач;
- самостоятельный поиск информации в Интернете;
- выполнение контрольной работы;
- консультации по наиболее сложным вопросам;

- участие в работе видео-клуба по кафедре и ежегодных студенческих научных конференциях;
- подготовку к экзамену.

Таблица 7 – Перечень вопросов, отводимых на самостоятельное освоение обучающимися дисциплины «Современные технологии программирования» направления подготовки 38.03.05 «Бизнес-информатика»

Наименование разделов, тем, входящих в дисциплину	Указание разделов и тем, отводимых на самостоятельное освоение обучающимися	Форма внеаудиторной самостоятельной работы
Введение в программирование на языке Python	Знакомство с интерактивной оболочкой IPython notebook. Изучение принципов работы в оболочке	работа с литературой; работа с электронными источниками; разработка алгоритмов и программ.
Основные синтаксические конструкции Python	Работа с множествами, генераторы списков и словарей.	работа с литературой; работа с электронными источниками; разработка алгоритмов и программ.
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	работа с литературой; работа с электронными источниками; разработка алгоритмов и программ.
Технологии работы со структурированными данными	Изучение библиотек Python для работ с данными в форматах XML, XHTML, HTML, JSON	работа с литературой; работа с электронными источниками; разработка алгоритмов и программ.
Технологии обработки данных	Изучение библиотек Python для анализа данных	работа с литературой; работа с электронными источниками; разработка алгоритмов и программ.

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

Проведение самостоятельной работы предполагает работу с конспектами лекций, решение задач по темам дисциплины, подготовку контрольных работ.

Текущий контроль осуществляется в ходе учебного процесса и консультирования студентов, по результатам выполнения самостоятельных работ. Основными формами текущего контроля знаний являются:

- обсуждение вынесенных в планах семинарских (практических) занятий вопросов тем и контрольных вопросов;
- решение задач, тестов и их обсуждение в точки зрения умения формулировать выводы, вносить рекомендации и принимать адекватные управленческие решения;
- выполнение контрольной работы;

О подходе к оценке знаний студентов преподаватель информирует студентов на первом семинарском (практическом) занятии. На последнем

семинарском (практическом) занятии студентам сообщается оценка, которую они получают по итогам работы в семестре. Студенты могут улучшить свою оценку по итогам работы в семестре за счет отработки пропущенных занятий. Отработка пропусков, имевших место по причине работы студентов во время занятий, не допускается.

Промежуточный контроль по учебной дисциплине «Современные технологии программирования» направления подготовки 38.03.05 «Бизнес-информатика», профиля «ИТ-менеджмент в бизнесе» проводится в форме экзамена в устной или письменной форме в виде ответов на вопросы.

Критерии балльно-рейтинговой оценки текущего контроля успеваемости содержатся в соответствующих методических рекомендациях кафедры «Математика и информатика».

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 14 1 5 котором ключем 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 `summlate` 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]. 16 1 7 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.

Примеры теоретических заданий домашней контрольной

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. Именованние 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, 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.
21. Объявление и вызов функции в Python. Параметры функции со значением по умолчанию и комментирование функции. Получение информации о функции. Способы передачи параметров при вызове функции.
22. 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.
23. Анонимные функции в Python их возможности и ограничения. Типичные сценарии использования анонимных функций.
24. Синтаксис и семантика обработки исключительных ситуаций в Python.
25. Создание пользовательских исключений и инструкция assert.
26. Базовые операции для работы с файлами в Python.
27. Использование инструкции with ... as на примере работы с файлами.
28. Использование модулей pickle и shelve для сохранения объектов в файл и их восстановления.
29. Модули в Python и их отличие от скриптов Python. Варианты синтаксиса импорта модуля и объектов модуля. Применение импортированных объектов. Порядок поиска модулей и специфика их загрузки. Загрузка модулей из глобального репозитория.
30. Импорт кода из пакетов. Организация пакетов в Python.
31. Концепция класса и объекта. Принципы и механизмы ООП.

32. Объявление класса, конструктор, создание объектов и одиночное наследование в Python. 33. Управление доступом к атрибутам класса в Python.
34. Полиморфизм и утиная типизация, проверка принадлежности объекта к классу в языке Python.
35. Методы классов и статические переменные, методы в Python.
36. Интроспекция и динамические операции с объектами в Python.
37. Специальные методы для использования пользовательских классов со стандартными операторами и функциями.
38. Основные возможности, поддерживаемые функциональными языками программирования. Поддержка функционального программирования в Python.
39. Концепция «функции – граждане первого класса» в языке программирования, поддержка этой концепции в Python. Специфика лямбда-функций в Python.
40. Глобальные и локальные переменные в функциях на примере Python. Побочные эффекты вызова функций и их последствия.
41. Вложенные функции и замыкания, специфика реализации в Python.
42. Функции высшего порядка и декораторы в Python.
43. Концепция map/filter/reduce. Работа map() в различных вариациях в Python. 2.2.3
44. Концепция map/filter/reduce. Работа filter() в Python. Использование модуля operator при работе с filter.
45. Концепция map/filter/reduce. Работа reduce() в Python. Использование reduce() с начальным значением, имеющим тип, отличный от возвращаемого редуцирующей функцией.
46. Итераторы в Python: встроенные итераторы, создание собственных итераторов, типичные способы обхода итераторов и принцип их работы.
47. Встроенные функции для работы с итераторами и возможности модуля itertools.
48. Функции генераторы и выражения генераторы: создание и применение в Python.
49. Специфика массивов, как структур данных. Динамические массивы – специфика работы, сложность операций. Специфика работы с array в Python.
50. Абстрактная структура данных стек: базовые и расширенные операции, их сложность. Реализация стека в Python.
51. Абстрактная структура данных очередь: базовые и расширенные операции, их сложность. Реализация очереди в Python.
52. Специфика реализации и скорости основных операций в очереди на базе массива и связанного списка.
53. Связанные списки: однонаправленные и двунаправленные. Сравнение скорости выполнения основных операций в связанных списках и в динамическом массиве.
54. Алгоритм сортировки выбором, сложность сортировки и возможности по ее улучшению.

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

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

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

58. Алгоритм сортировки слиянием, сложность сортировки.

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

Оценочные средства предназначены для контроля и оценки образовательных достижений обучающихся, освоивших программу учебной дисциплины «Современные технологии программирования».

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

7.1. Описание показателей и критериев оценивания компетенций

Планируемые результаты освоения компетенции (индикатора достижения компетенции)	Уровень освоения				Оценочное средство
	«неудовлетворительно»	«удовлетворительно»	«хорошо»	«отлично»	
УК-4 Способность использовать прикладное программное обеспечение при решении профессиональных задач					
1. Использует основные методы и средства получения, представления, хранения и обработки данных.					
Знать: основные методы и средства получения, представления, хранения и обработки данных на базе современных технологий прикладного программирования и баз данных.	Фрагментарное представление об основных методах и средствах получения, представления, хранения и обработки данных на базе современных технологий прикладного программирования и баз	Неполные представления об основных методах и средствах получения, представления, хранения и обработки данных на базе современных технологий прикладного программирования и баз данных	Сформированные, но содержащие отдельные пробелы представления об основных методах и средствах получения, представления, хранения и обработки данных на базе современных технологий	Сформированные систематические представления об основных методах и средствах получения, представления, хранения и обработки данных на базе современных технологий прикладного программирования	Вопросы для оценки знаний и умений, задания в виде расчетных задач, тестовые задания

Планируемые результаты освоения компетенции (индикатора достижения компетенции)	Уровень освоения				Оценочное средство
	«неудовлетворительно»	«удовлетворительно»	«хорошо»	«отлично»	
	данных		прикладного программирования и баз данных	ания и баз данных	
Уметь: применять основные методы и средства получения, представления, хранения и обработки данных на базе современных технологий прикладного программирования и баз данных	Фрагментарное умение применять основные методы и средства получения, представления, хранения и обработки данных на базе современных технологий прикладного программирования и баз данных	Несистематическое применение умений применять основные методы и средства получения, представления, хранения и обработки данных на базе современных технологий прикладного программирования и баз данных	В целом успешное, но содержащее отдельные пробелы умение применять основные методы и средства получения, представления, хранения и обработки данных на базе современных технологий прикладного программирования и баз данных	Сформированное умение применять основные методы и средства получения, представления, хранения и обработки данных на базе современных технологий прикладного программирования и баз данных	Вопросы для оценки знаний и умений, задания в виде расчетных задач, тестовые задания
2. Демонстрирует владение профессиональными пакетами прикладных программ.					
Знать: принципы использования профессиональных пакетов прикладных программ на базе современных технологий прикладного программирования и обработки данных.	Фрагментарное представление о принципах использования профессиональных пакетов прикладных программ на базе современных технологий прикладного программирования и	Неполные представления о принципах использования профессиональных пакетов прикладных программ на базе современных технологий прикладного программирования и обработки	Сформированные, но содержащие отдельные пробелы представления о принципах использования профессиональных пакетов прикладных программ на базе современных технологий	Сформированные систематические представления о принципах использования профессиональных пакетов прикладных программ на базе современных технологий прикладного программирования и	Вопросы для оценки знаний и умений, задания в виде расчетных задач, тестовые задания

Планируемые результаты освоения компетенции (индикатора достижения компетенции)	Уровень освоения				Оценочное средство
	«неудовлетворительно»	«удовлетворительно»	«хорошо»	«отлично»	
	обработки данных.	данных.	прикладного программирования и обработки данных.	обработки данных.	
Уметь: использовать профессиональные пакеты прикладных программ на базе современных технологий прикладного программирования и обработки данных.	Фрагментарное умение использовать профессиональные пакеты прикладных программ на базе современных технологий прикладного программирования и обработки данных	Несистематическое применение умений использовать профессиональные пакеты прикладных программ на базе современных технологий прикладного программирования и обработки данных	В целом успешное, но содержащее отдельные пробелы умение использовать профессиональные пакеты прикладных программ на базе современных технологий прикладного программирования и обработки данных	Сформированное умение использовать профессиональные пакеты прикладных программ на базе современных технологий прикладного программирования и обработки данных	Вопросы для оценки знаний и умений, задания в виде расчетных задач, тестовые задания
3. Выбирает необходимое прикладное программное обеспечение в зависимости от решаемой задачи					
Знать: критерии выбора прикладного программного обеспечения на базе современных технологий прикладного программирования и обработки данных в зависимости от решаемой задачи	Фрагментарное представление о критериях выбора прикладного программного обеспечения на базе современных технологий прикладного программирования и обработки данных в зависимости от решаемой задачи	Неполные представления о критериях выбора прикладного программного обеспечения на базе современных технологий прикладного программирования и обработки данных в зависимости от решаемой задачи	Сформированные, но содержащие отдельные пробелы представления о критериях выбора прикладного программного обеспечения на базе современных технологий прикладного программирования и обработки данных в зависимости от решаемой задачи	Сформированные систематические представления о критериях выбора прикладного программного обеспечения на базе современных технологий прикладного программирования и обработки данных в зависимости от решаемой задачи	Вопросы для оценки знаний и умений, задания в виде расчетных задач, тестовые задания

Планируемые результаты освоения компетенции (индикатора достижения компетенции)	Уровень освоения				Оценочное средство
	«неудовлетворительно»	«удовлетворительно»	«хорошо»	«отлично»	
			зависимости от решаемой задачи	от решаемой задачи	
Уметь: выбрать прикладное программное обеспечение на базе современных технологий прикладного программирования и обработки данных в зависимости от решаемой задачи	Фрагментарное умение выбрать прикладное программное обеспечение на базе современных технологий прикладного программирования и обработки данных в зависимости от решаемой задачи	Несистематическое применение умений выбрать прикладное программное обеспечение на базе современных технологий прикладного программирования и обработки данных в зависимости от решаемой задачи	В целом успешное, но содержащее отдельные пробелы умение выбрать прикладное программное обеспечение на базе современных технологий прикладного программирования и обработки данных в зависимости от решаемой задачи	Сформированное умение выбрать прикладное программное обеспечение на базе современных технологий прикладного программирования и обработки данных в зависимости от решаемой задачи	Вопросы для оценки знаний и умений, задания в виде расчетных задач, тестовые задания
4. Использует прикладное программное обеспечение для решения конкретных прикладных задач					
Знать: принципы и процедуры использования прикладного программного обеспечения для решения конкретных прикладных задач на базе современных технологий прикладного программирования и обработки	Фрагментарное представление о принципах и процедурах использования прикладного программного обеспечения для решения конкретных прикладных задач на базе современных технологий прикладного	Неполные представления о принципах и процедурах использования прикладного программного обеспечения для решения конкретных прикладных задач на базе современных технологий прикладного программирования	Сформированные, но содержащие отдельные пробелы представления о принципах и процедурах использования прикладного программного обеспечения для решения конкретных прикладных задач на базе	Сформированные систематические представления о принципах и процедурах использования прикладного программного обеспечения для решения конкретных прикладных задач на базе современных технологий прикладного программирования	Вопросы для оценки знаний и умений, задания в виде расчетных задач, тестовые задания

Планируемые результаты освоения компетенции (индикатора достижения компетенции)	Уровень освоения				Оценочное средство
	«неудовлетворительно»	«удовлетворительно»	«хорошо»	«отлично»	
данных.	программирования и обработки данных	вания и обработки данных	современных технологий прикладного программирования и обработки данных	ания и обработки данных	
Уметь: применить прикладное программное обеспечение для решения конкретных прикладных задач с использованием современных технологий прикладного программирования и обработки данных	Фрагментарное умение применить прикладное программное обеспечение для решения конкретных прикладных задач с использованием современных технологий прикладного программирования и обработки данных	Несистематическое применение умений применить прикладное программное обеспечение для решения конкретных прикладных задач с использованием современных технологий прикладного программирования и обработки данных	В целом успешное, но содержащее отдельные пробелы умение применить прикладное программное обеспечение для решения конкретных прикладных задач с использованием современных технологий прикладного программирования и обработки данных	Сформированное умение применить прикладное программное обеспечение для решения конкретных прикладных задач с использованием современных технологий прикладного программирования и обработки данных	Вопросы для оценки знаний и умений, задания в виде расчетных задач, тестовые задания
УК-10 Способность осуществлять поиск, критически анализировать, обобщать и систематизировать информацию, использовать системный подход для решения поставленных задач					
1. Четко описывает состав и структуру требуемых данных и информации, грамотно реализует процессы их сбора, обработки и интерпретации.					
Знать: состав и структуру требуемых данных, методы их сбора, обработки и интерпретации	Фрагментарное представление о составе и структуре требуемых данных, методах их сбора, обработки и	Неполные представления о составе и структуре требуемых данных, методах их сбора, обработки и интерпретации	Сформированные, но содержащие отдельные пробелы представления о составе и структуре требуемых данных,	Сформированные систематические представления о составе и структуре требуемых данных, методах их	Вопросы для оценки знаний и умений, задания в виде расчетных задач, тестовые задания

Планируемые результаты освоения компетенции (индикатора достижения компетенции)	Уровень освоения				Оценочное средство
	«неудовлетворительно»	«удовлетворительно»	«хорошо»	«отлично»	
	интерпретации	ии	методах их сбора, обработки и интерпретации	сбора, обработки и интерпретации	
Уметь: четко описывать состав и структуру требуемых данных и информации, грамотно реализовывать процессы их сбора, обработки и интерпретации	Фрагментарное умение четко описывать состав и структуру требуемых данных и информации, грамотно реализовывать процессы их сбора, обработки и интерпретации	Несистематическое применение умений четко описывать состав и структуру требуемых данных и информации, грамотно реализовывать процессы их сбора, обработки и интерпретации	В целом успешное, но содержащее отдельные пробелы умение четко описывать состав и структуру требуемых данных и информации, грамотно реализовывать процессы их сбора, обработки и интерпретации	Сформированное умение четко описывать состав и структуру требуемых данных и информации, грамотно реализовывать процессы их сбора, обработки и интерпретации	Вопросы для оценки знаний и умений, задания в виде расчетных задач, тестовые задания
2. Обосновывает сущность происходящего, выявляет закономерности, понимает природу variability.					
Знать: Теоретические основы понимания природы variability	Фрагментарное представление о теоретических основах понимания природы variability	Неполные представления о теоретических основах понимания природы variability	Сформированные, но содержащие отдельные пробелы представления о теоретических основах понимания природы variability	Сформированные систематические представления о теоретических основах понимания природы variability	Вопросы для оценки знаний и умений, задания в виде расчетных задач, тестовые задания
Уметь: обосновывать сущность происходящего, выявлять закономерности	Фрагментарное умение обосновывать сущность происходящего, выявлять закономерности	Несистематическое применение умений обосновывать сущность происходящего, выявлять закономерности	В целом успешное, но содержащее отдельные пробелы умение	Сформированное умение обосновывать сущность происходящего, выявлять закономерности	Вопросы для оценки знаний и умений, задания в виде

Планируемые результаты освоения компетенции (индикатора достижения компетенции)	Уровень освоения				Оценочное средство
	«неудовлетворительно»	«удовлетворительно»	«хорошо»	«отлично»	
сти, понимать природу variability	закономерно сти, понимать природу variability	происходяще го, выявлять закономерности, понимать природу variability	обосновывать сущность происходяще го, выявлять закономерности, понимать природу variability	закономерности, понимать природу variability	расчетных задач, тестовые задания
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.					
Знать: теоретические основы методов оценки результатов классификации, прикладное назначение классификационных групп	Фрагментарное представление о теоретических основах методов оценки результатов классификации, прикладное назначение классификационных групп	Неполные представления о теоретических основах методов оценки результатов классификации, прикладное назначение классификационных групп	Сформированные, но содержащие отдельные пробелы представления о теоретических основах методов оценки результатов классификации, прикладное назначение классификационных групп	Сформированные систематические представления о теоретических основах методов оценки результатов классификации, прикладное назначение классификационных групп	Вопросы для оценки знаний и умений, задания в виде расчетных задач, тестовые задания
Уметь: формулировать признак классификации, выделять соответствующие ему группы однородных "объектов", идентифицировать общие	Фрагментарное умение формулировать признак классификации, выделять соответствующие ему группы однородных "объектов", идентифицировать общие	Несистематическое применение умений формулировать признак классификации, выделять соответствующие ему группы однородных	В целом успешное, но содержащее отдельные пробелы умение формулировать признак классификации, выделять соответствующие ему	Сформированное умение формулировать признак классификации, выделять соответствующие ему группы однородных "объектов", идентифицировать общие	Вопросы для оценки знаний и умений, задания в виде расчетных задач, тестовые задания

Планируемые результаты освоения компетенции (индикатора достижения компетенции)	Уровень освоения				Оценочное средство
	«неудовлетворительно»	«удовлетворительно»	«хорошо»	«отлично»	
свойства элементов этих групп, оценивать полноту результатов классификации, показывать прикладное назначение классификационных групп	описать общие свойства элементов этих групп, оценивать полноту результатов классификации, показывать прикладное назначение классификационных групп	"объектов", идентифицировать общие свойства элементов этих групп, оценивать полноту результатов классификации, показывать прикладное назначение классификационных групп	группы однородных "объектов", идентифицировать общие свойства элементов этих групп, оценивать полноту результатов классификации, показывать прикладное назначение классификационных групп	описать общие свойства элементов этих групп, оценивать полноту результатов классификации, показывать прикладное назначение классификационных групп	
4. Грамотно, логично, аргументировано формирует собственные суждения и оценки. Отличает факты от мнений, интерпретаций, оценок и т.д. в рассуждениях других участников деятельности					
Знать: основы грамотной, логичной, аргументированной формулировки и собственных суждений и оценки.	Фрагментарное представление об основах грамотной, логичной, аргументированной формулировки и собственных суждений и оценки.	Неполное представление об основах грамотной, логичной, аргументированной формулировки и собственных суждений и оценки.	Сформированные, но содержащие отдельные пробелы представления об основах грамотной, логичной, аргументированной формулировки и собственных суждений и оценки.	Сформированные систематические представления об основах грамотной, логичной, аргументированной формулировки и собственных суждений и оценки.	Вопросы для оценки знаний и умений, задания в виде расчетных задач, тестовые задания
Уметь: грамотно, логично, аргументировано формировать собственные	Фрагментарное умение грамотно, логично, аргументировано формировать	Несистематическое применение умений грамотно, логично, аргументиро	В целом успешное, но содержащее отдельные пробелы умение грамотно,	Сформированное умение грамотно, логично, аргументировано формировать	Вопросы для оценки знаний и умений, задания в виде расчетных

Планируемые результаты освоения компетенции (индикатора достижения компетенции)	Уровень освоения				Оценочное средство
	«неудовлетворительно»	«удовлетворительно»	«хорошо»	«отлично»	
суждения и оценки. Отличать факты от мнений, интерпретаций, оценок и т.д. в рассуждениях других участников деятельности	собственные суждения и оценки. Отличать факты от мнений, интерпретаций, оценок и т.д. в рассуждениях других участников деятельности	вано формировать собственные суждения и оценки. Отличать факты от мнений, интерпретаций, оценок и т.д. в рассуждениях других участников деятельности	логично, аргументировано формировать собственные суждения и оценки. Отличать факты от мнений, интерпретаций, оценок и т.д. в рассуждениях других участников деятельности	собственные суждения и оценки. Отличать факты от мнений, интерпретаций, оценок и т.д. в рассуждениях других участников деятельности	задач, тестовые задания

7.2. Вопросы для оценки знаний и умений, характеризующих формирование компетенций

Шифр компетенции	Вопросы	Правильный ответ
УК-4	1. Какой показатель надо вычислить, чтобы определить наиболее частое значение в наборе данных?	мода
	2. Напишите название модуля Python, который содержит статистические функции	statistics
	3. Напишите команду Python для вычисления медианы по списку x	statistics.median(x)
	4. Какой модуль Python позволяет загружать таблицы в структуру данных data frame?	pandas
	5. Какой показатель оценивает средний квадрат отклонения от среднего?	дисперсия
	6. Каким символом отделяются друг от друга значения в файлах .csv?	запятая
	7. Какой тип переменной следует использовать в Python для действительных чисел?	float
	8. Какой закон распределения описывается кривой Гаусса?	нормальный закон
	9. От чего зависит достоверность оценки среднего значения показателя?	от объема выборки и дисперсии
	10. Как меняется стандартная ошибка	уменьшается

	среднего при увеличении объема выборки?	
УК-10	11. Как называют значения, которые сильно выбиваются из набора данных, не вписываются в модель по какой-либо причине?	выбросы
	12. Какие форматы файлов с данными может загружать программа, написанная на Python?	любые, если алгоритм может их обработать
	13. Как называется библиотека Python, которая используется для визуализации данных, построения графиков, диаграмм	matplotlib
	14. Какой критерий используется для проверки закона распределения?	критерий Пирсона или критерий Колмогорова
	15. Как называется показатель, оценивающий взаимосвязь двух величин? Как его вычислить в Python?	коэффициент корреляции; <code>numpy.corrcoef(x,y)</code>
	16. Какой модуль Python применяется для проверки стат. гипотез?	<code>scipy.stats</code>
	17. Если надо проверить, что среднее больше определенного значения, то какой критерий надо применить?	критерий Стьюдента (t-статистика)
	18. Какой метод применяется для определения коэффициентов линейной регрессии?	метод наименьших квадратов
	19. Какие значения чаще всего принимают величины, подчиняющиеся нормальному закону распределения?	значения возле среднего
	20. В чем смысл правила 3 сигма?	отклонение от среднего более чем на 3 стандартных отклонения маловероятно

7.3 Практико-ориентированные задания

Шифр компетенции	Практико-ориентированные задания	Правильный ответ
УК-4	1. Напишите программу, которая считывает данные в 4-м столбце таблицы в файле "GAZP.csv" в список	<pre>f=open('GAZP.csv','r') x=[] for i in range(10): z=f.readline().split(',')[3] try: z=float(z) x.append(z) except: pass</pre>
	2. Напишите программу, загружающую таблицу из файла как data frame	<pre>import pandas as pd table1= pd.read_csv('GAZP.csv') df=pd.DataFrame(table1)</pre>
	3. Напишите программу, оценивающую центр распределения данных,	<pre>import statistics mean_y=statistics.mean(y) median_y=statistics.median(y)</pre>

	содержащихся в списке y.	<pre>try: mode_y=statistics.mode(y) except: mode_y='не существует'</pre>
	4. Напишите программу, составляющую словарь, где ключи – названия показателей (среднее, ст отклонение), а значения – их величины.	<pre>import statistics mean_y=statistics.mean(y) stdev_y=statistics.stdev(y) dict_y=dict_y={'среднее':mean_y,'ст отклонение':stdev_y}</pre>
	5. Как можно импортировать функцию mean() из библиотеки statistics, чтобы не указывать название библиотеки при её использовании?	<pre>from statistics import mean print(mean([1,2,3]))</pre>
	6. Вычислите коэффициент вариации в Python	<pre>import statistics mean_y=statistics.mean(y) stdev_y=statistics.stdev(y) if mean_y != 0: c_var=stdev_y/mean_y else: c_var='не существует, т.к. среднее равно 0'</pre>
	7. Посчитайте размах данных по списку y	<pre>min_y=min(y) max_y=max(y) range_y=max_y-min_y</pre>
	8. Постройте график по 3 точкам, используя библиотеку matplotlib	<pre>import matplotlib.pyplot as plt x = [1,2,3] y = [2,4,1] plt.plot(x, y) plt.xlabel('x - axis') plt.ylabel('y - axis') plt.title('graph') plt.show()</pre>
	9. Напишите программу для построения графика по загруженной из файла таблице	<pre>import matplotlib.pyplot as plt import pandas as pd df = pd.read_csv('fdata.csv', sep=',', parse_dates=True, index_col=0) df.plot() plt.show()</pre>
УК-10	10. Посчитайте коэффициент корреляции Пирсона между двумя показателями	<pre>import numpy as np x = np.array([0, 1, 3]) y = np.array([2, 4, 5]) print(np.corrcoef(x, y))</pre>
	11. Посчитайте дневные доходности акций «Газпром», их среднее и стандартное отклонение	<pre>from statistics import mean, stdev f=open('GAZP.csv','r') x=[] for i in range(10): z=f.readline().split(',')[3]</pre>

		<pre> try: z=float(z) x.append(z) except: pass for i in range(len(x)-1): d[i]=(x[i]-x[i-1])/x[i-1] print(mean(d),stdev(d)) </pre>
	<p>12. Найдите среди всех заработных плат (Wage) самую высокую и запишите это значение в переменную rich. Посчитайте, сколько зарабатывают сотрудники за год (Wage) вместе, и сохраните значение в переменной total. Выведите результаты кода через запятую: rich, low, total.</p>	<pre> import pandas as pd table1 = pd.read_csv('personel.csv') rich = table1.Wage.max() low = table1.Value.min() total = table1.Wage.sum() print(rich, low, total) </pre>
	<p>13. Напишите программу, которая загрузит из файла таблицу с данными по сотрудникам и посчитает средний возраст, округленный до целых (столбец Age)</p>	<pre> import pandas as pd table1 = pd.read_csv('personel.csv') age = round(table1.Age.mean()) </pre>
	<p>14. Напишите программу на Python для построения двух или более линий с легендами, различной ширины и цвета.</p>	<pre> import matplotlib.pyplot as plt # line 1 x1 = [10,20,30] y1 = [20,40,10] # line 2 x2 = [10,20,30] y2 = [40,10,30] plt.xlabel('x - axis') plt.ylabel('y - axis') plt.title("Two or more lines") plt.plot(x1,y1, color='blue', linewidth = 3, label='line1-width-3') plt.plot(x2,y2, color='red', linewidth = 5, label='line2-width-5') plt.legend() plt.show() </pre>
	<p>15. Напишите программу на Python для отображения сетки и нарисуйте линейные графики</p>	<pre> import datetime as DT from matplotlib import pyplot as plt from matplotlib.dates import date2num data = [(DT.datetime.strptime('2016-10-03', </pre>

	<p>значения закрытия Alphabet Inc. в период с 3 октября 2016 года по 7 октября 2016 года. Настроил линии сетки с рендерингом с большей сеткой (основной сеткой) и меньшей сетка (второстепенная сетка). Включите сетку, но отключите галочки.</p>	<pre> "% Y-%m-%d"), 772.559998), (DT.datetime.strptime('2016-10-04', "% Y- % m-%d"), 776.429993), (DT.datetime.strptime('2016-10-05', "% Y- % m-%d"), 776.469971), (DT.datetime.strptime('2016-10-06', "% Y- % m-%d"), 776.859985), (DT.datetime.strptime('2016-10-07', "% Y- % m-%d"), 775.080017)] x = [date2num(date) for (date, value) in data] y = [value for (date, value) in data] fig = plt.figure() graph = fig.add_subplot(111) # Plot the data as a red line with round markers graph.plot(x,y,'r-o') # Set the xtick locations graph.set_xticks(x) # Set the xtick labels graph.set_xticklabels([date.strftime("% Y-%m-%d") for (date, value) in data]) # naming the x axis plt.xlabel('Date') # naming the y axis plt.ylabel('Closing Value') # giving a title plt.title('Closing stock value of Alphabet Inc.') # Turn on the minor TICKS, which are required for the minor GRID plt.minorticks_on() # Customize the major grid plt.grid(which='major', linestyle='-', lin- ewidth='0.5', color='red') # Customize the minor grid plt.grid(which='minor', linestyle=':', lin- ewidth='0.5', color='black') # Turn off the display of all ticks. plt.tick_params(which='both', # Options for both major and minor ticks top='off', # turn off top ticks left='off', # turn off left ticks right='off', # turn off right ticks bottom='off') # turn off bottom ticks plt.show() </pre>
	<p>16. Напишите программу на Python для отображения гистограммы популярности языков программирования.</p>	<pre> import matplotlib.pyplot as plt x = ['Java', 'Python', 'PHP', 'JavaScript', 'C#', 'C++'] popularity = [22.2, 17.6, 8.8, 8, 7.7, 6.7] x_pos = [i for i, _ in enumerate(x)] plt.bar(x_pos, popularity, color='blue') plt.xlabel("Languages") </pre>

		<pre>plt.ylabel("Popularity") plt.title("Popularity of Programming Language\n" + "Worldwide, Oct 2017 compared to a year ago") plt.xticks(x_pos, x) # Turn on the grid plt.minorticks_on() plt.grid(which='major', linestyle='-', linewidth='0.5', color='red') # Customize the minor grid plt.grid(which='minor', linestyle=':', linewidth='0.5', color='black') plt.show()</pre>
	17. Напишите программу с использованием модуля Pandas для удаления последних n строк таблицы, записанной в структуру данных DataFrame.	<pre>import pandas as pd d = {'col1': [1, 2, 3, 4, 7, 11], 'col2': [4, 5, 6, 9, 5, 0], 'col3': [7, 5, 8, 12, 1, 11]} df = pd.DataFrame(data=d) print("Original DataFrame") print(df) print("\nAfter removing last 3 rows of the said DataFrame:") df1 = df.iloc[:3] print(df1)</pre>
	18. Решите задачу на построение линейной регрессии по двум спискам числовых значений	<pre>import numpy as np from sklearn.linear_model import LinearRegression x = np.array([5, 15, 25, 35, 45, 55]).reshape((-1, 1)) y = np.array([5, 20, 14, 32, 22, 38]) model = LinearRegression() model.fit(x, y) model = LinearRegression().fit(x, y) y_pred = model.predict(x) print('predicted response:', y_pred, sep='\n') y_pred = model.intercept_ + model.coef_ * x print('predicted response:', y_pred, sep='\n')</pre>

7.4 Тесты

Шифр компетенции	Тестовые задания	Правильный ответ
УК-4	1. Число, делящее множество чисел пополам, так, что 50% значений больше него, 50% меньше - 1) медиана 2) мода 3) дисперсия 4) среднее	1
	2. Для оценки разброса данных вычисляют: 1) дисперсию 2) моду 3) асимметрию 4) эксцесс	1

	3. Чтобы получить доступ к 1-му элементу списка x надо написать: 1) x[1] 2) x[0] 3) x(1) 4) x(0)	2
	4. Функция mode содержится в библиотеке 1) statistics 2) numpy 3) pandas 4) scipy	1
	5. Если медиана и мода больше среднего, можно ожидать, что: 1) чаще будут встречаться значения выше среднего 2) чаще будут встречаться значения ниже среднего 3) значения выше и ниже среднего будут встречаться с равной вероятностью	1
	6. Программа, написанная на Python, превосходит расчеты в табличном процессоре 1) скоростью вычислений 2) простотой создания 3) объемом на жестком диске 4) объемом в оперативной памяти	1
УК-10	7. Функции и методы для построения графиков содержатся в библиотеке: 1) statistics 2) matplotlib 3) numpy 4) pandas	2
	8. Функция stdev содержится в библиотеке 1) statistics 2) numpy 3) pandas 4) scipy	1
	9. Команды, которые можно использовать для загрузки столбца таблицы, содержащего текстовые и числовые значения вперемешку: 1) try except 2) if else 3) float, str	1
	10. Закон распределения, который лучше всего описывает распределение доходностей финансового актива: 1) нормальный 2) показательный 3) равномерный 4) гама	1
	11. Для моделирования временного интервала между событиями целесообразнее всего применять: 1) нормальный закон распределения 2) показательный закон распределения 3) равномерный закон распределения 4) гама распределение	2

	12. При проверке стат. гипотезы о числовом значении показателя нулевая гипотеза обычно формулируется о 1) равенстве 2) неравенстве	1
--	--	---

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

Основная литература

1. Жмак, Е. И. Информационные технологии и основы программирования : учебное пособие : в 2 частях / Е. И. Жмак, А. Л. Коваль. — Новосибирск : НГТУ, 2022 — Часть 1 : Информационные технологии и основы программирования : учебное пособие : в 2 ч. Ч1. — 2022. — 64 с. — ISBN 978-5-7782-4839-7. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/404771> — Режим доступа: для авториз. пользователей.

2. Згуральская, Е. Н. Технологии программирования : учебное пособие / Е. Н. Згуральская. — Ульяновск : УлГТУ, 2020. — 71 с. — ISBN 978-5-9795-1995-1. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/165011> — Режим доступа: для авториз. пользователей.

Дополнительная литература

3. Никитина, Т. П. Программирование. Основы Python / Т. П. Никитина, Л. В. Королев. — Санкт-Петербург : Лань, 2023. — 156 с. — ISBN 978-5-507-45283-5. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/302714> — Режим доступа: для авториз. пользователей.

4. Букунов, С. В. Разработка приложений с графическим пользовательским интерфейсом на языке Python / С. В. Букунов, О. В. Букунова. — Санкт-Петербург : Лань, 2023. — 88 с. — ISBN 978-5-507-45191-3. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/292856> — Режим доступа: для авториз. пользователей.

5. Маккинни, У. Python и анализ данных / У. Маккинни ; перевод с английского А. А. Слинкина. — 2-ое изд., испр. и доп. — Москва : ДМК Пресс, 2020. — 540 с. — ISBN 978-5-97060-590-5. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/131721> — Режим доступа: для авториз. пользователей.

6. Янцев, В. В. Web-программирование на Python : учебное пособие для вузов / В. В. Янцев. — 3-е изд., перераб. — Санкт-Петербург : Лань, 2024. —

180 с. — ISBN 978-5-507-48364-8. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/392993> — Режим доступа: для авториз. пользователей.

7. SciPy // <http://docs.scipy.org/doc/scipy/reference/>

8. NumPy User Guide // <http://docs.scipy.org/doc/numpy/user/index.htm>

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

Интернет-ресурсы

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. Информационно-образовательный портал Финансового университета при Правительстве Российской Федерации <http://portal.ufrf.ru/>

8. Каталог курсов Интернет Университета Информационных Технологий <http://www.intuit.ru/>

9. Электронная библиотека Финансового университета (ЭБ) <http://elib.fa.ru/> (<http://library.fa.ru/files/elibfa.pdf>)

10. Электронно-библиотечная система Znanium <http://www.znanium.com>

11. Сайт департамента анализа данных, принятия решений и финансовых технологий. http://fa.ru/dep/data_analysis/

12. Библиотечно – информационный комплекс Финуниверситета при Правительстве РФ. <http://library.fa.ru>.

10. Методические указания для обучающихся по освоению дисциплины

Общие положения по выполнению контрольной работы

Контрольная работа по дисциплине «Современные технологии программирования» предназначена для проверки степени усвоения студентами пройденного материала по данной дисциплине.

Контрольная работа выполняется после прослушивания студентами лекционного курса по дисциплине и их самостоятельной работы с рекомендованной преподавателем учебной литературой. Контрольная работа

должна показать, что ее автор освоил фундаментальные знания в области ценообразования.

Контрольная работа носит теоретико-практический характер и дает дополнительный опыт самостоятельной работы над выбранной темой, заключающийся в подборе необходимой литературы, письменном изложении материала на основе систематизации, обобщении, критическом анализе изученного материала и умении решения практических задач.

В методических указаниях по дисциплине «Современные технологии программирования» представлены 10 вариантов контрольных работ. Номер варианта контрольной работы устанавливается в соответствии с последней цифрой номера зачетной книжки.

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

Требования к оформлению контрольной работы

Текст контрольной работы выполняется с использованием компьютера и распечатывается на одной стороне листа белой бумаги формата А4. Цвет шрифта должен быть черным, шрифт – Times New Roman, размер 14, межстрочный интервал - 1,5. Полуужирный шрифт для выделения названий структурных элементов работы, отдельных слов не используется. Не разрешается использовать компьютерные возможности акцентирования внимания на отдельных терминах, положениях, формулах путем использования шрифтов разной гарнитуры.

Номера страниц проставляют в середине нижнего поля листа, соблюдая сквозную нумерацию. Точка в номере страницы не ставится. Титульный лист включают в общую нумерацию страниц, но номер страницы не проставляется. Нумерация начинается со второй страницы — «Содержание».

Формулы в контрольной работе выделяют из текста в отдельную строку. Выше и ниже каждой формулы должна быть оставлена одна свободная строка. Пояснение значений символов и числовых коэффициентов следует приводить непосредственно под формулой в той же последовательности, в которой они даны в формуле. Формулы нумеруются арабскими цифрами сквозной нумерацией по всей работе, при этом номер формулы указывается в круглых скобках в крайнем правом положении на строке. Формулы должны быть написаны с помощью редактора формул.

Таблицы располагаются непосредственно после текста, в котором они упоминаются впервые, или на следующей странице, нумеруются арабскими цифрами сквозной нумерацией по всей работе. Заголовок таблицы располагается по ширине страницы. Слово «Таблица», ее порядковый номер и название через тире помещают над таблицей слева без абзацного отступа. Точка в конце заголовка не ставится. После таблицы до следующего основного текста работы пропускают одну строку полуторного интервала.

В заключительной части контрольной работы необходимо привести список использованных источников, содержащий не менее 10-12 учебников, монографий и статей периодической печати.

Источники нумеруются арабскими цифрами без точки и печатаются с абзацного отступа.

Объем контрольной работы не должен превышать 12 страниц формата А4 (шрифт 14, полуторный интервал). Образец оформления титульного листа приведен далее.

Варианты контрольных работ приведены в соответствующих учебных изданиях кафедры «Математика и информатика».

Пример оформления титульного листа контрольной работы

Федеральное государственное образовательное бюджетное учреждение
высшего образования

**«ФИНАНСОВЫЙ УНИВЕРСИТЕТ
ПРИ ПРАВИТЕЛЬСТВЕ РОССИЙСКОЙ ФЕДЕРАЦИИ»
(Финансовый университет)**

Краснодарский филиал Финуниверситета

Кафедра «Математика и информатика»

КОНТРОЛЬНАЯ РАБОТА
по дисциплине **«СОВРЕМЕННЫЕ ТЕХНОЛОГИИ ПРИКЛАДНОГО
ПРОГРАММИРОВАНИЯ И ОБРАБОТКИ ДАННЫХ»**

Направление _____

Профиль _____

Вариант _____

Студент _____
(И.О.Ф.)

курс _____

Преподаватель _____
(уч. степень.,
должность И.О.Ф.)

Краснодар 202_

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

Используется свободное программное обеспечение

11.2 Современные профессиональные базы данных и информационные справочные системы

1 Справочная правовая система «Консультант Плюс». URL: <http://www.consultant.ru>

2 Информационно-правовая система «Гарант-аэро». URL: <http://www.garant.ru>

11.3 Сертифицированные программные и аппаратные средства защиты информации: не предусмотрены

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

Электронная информационно-образовательная среда Финансового университета обеспечивает:

– доступ к учебным планам, рабочим программам дисциплин, практик, к указаниям электронных библиотечных систем и электронным образовательным ресурсам, указанным в рабочих программах. По дисциплине «Современные технологии прикладного программирования и обработки данных» в ИОП представлены следующие виды информационных ресурсов:

- аннотации дисциплины;
- видеолекции;
- мультимедийные презентации по всем темам курса;
- методический материал;
- рабочие программы дисциплины.

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

– фиксацию хода образовательного процесса, результатов промежуточной аттестации;

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

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

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

Образовательный процесс обеспечивается специальными помещениями, которые представляют собой аудитории для проведения занятий лекционного типа, занятий семинарского типа, выполнения курсовых работ, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, помещения для самостоятельной работы студентов и помещения для хранения и профилактического обслуживания учебного оборудования. Специальные помещения укомплектованы специализированной мебелью и техническими средствами обучения, обеспечивающими представление учебной информации большой аудитории, демонстрационным оборудованием.

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