

Федеральное государственное образовательное бюджетное учреждение
высшего образования

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

**Кафедра информационных технологий
Факультет информационных технологий и анализа больших данных**

Документ подписан усиленной неквалифицированной электронной подписью
Организация: Финансовый университет при Правительстве РФ
Утверждено: Проректор по учебной и методической работе Е.А. Каменева
Сертификат: ru8BoWz0FeuMZB/K+xsy6Ojj4VuScVC6
Дата: 26.09.2025 г.

М.В. Смирнов

Практикум по программированию

Рабочая программа дисциплины

для студентов, обучающихся по направлению подготовки:

38.03.01 - Экономика,

Образовательная программа «Финансовые инструменты и цифровые технологии
анализа»

Профиль:

«Финансовые инструменты и цифровые технологии анализа»

Рекомендовано

*Факультет информационных технологий и анализа больших данных
(протокол № 03 от 16.12.2025 г.)*

Одобрено

*Кафедра информационных технологий
(протокол № 12 от 03.12.2025 г.)*

© Москва 2025

СОДЕРЖАНИЕ

1.	Наименование дисциплины	3
2.	Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине	3
3.	Место дисциплины в структуре образовательной программы	6
4.	Объем дисциплины (модуля) 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	10
5.3.	Содержание семинаров	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	16
7.	Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине	21
8.	Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины	34
9.	Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины	36
10.	Методические указания для обучающихся по освоению дисциплины	37
11.	Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень необходимого программного обеспечения и информационных справочных систем	37
12.	Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине	38

1. Наименование дисциплины

«Практикум по программированию».

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

Код компетенции	Наименование компетенции	Индикаторы достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции
ПКН-3	Способность осуществлять сбор, обработку и статистический анализ данных, применять математические методы для решения стандартных профессиональных финансово-экономических задач, интерпретировать полученные результаты	Проводит сбор, обработку и статистический анализ данных для решения финансово-экономических задач	знать: методы сбора, обработки и статистического анализа данных, применяемые при решении финансово-экономических задач уметь: выбирать и применять методы сбора, обработки и статистического анализа данных при решении финансово-экономических задач
		Формулирует математические постановки финансово-экономических задач, переходит от экономических постановок задач к математическим моделям	знать: математические понятия, используемые при построении математических моделей экономических задач уметь: формулировать математические постановки финансово-экономических задач и переходить от экономических постановок задач к математическим моделям
		Системно подходит к выбору математических методов и информационных технологий для решения конкретных финансово-экономических задач в профессиональной области	знать: особенности математических методов и информационных технологий, используемых для решения финансово-экономических задач в профессиональной области уметь: обоснованно выбирать математические методы и информационные технологии для решения конкретных финансово-экономических задач в профессиональной области
		Анализирует результаты исследования математических моделей финансово-экономических задач и делает на их	знать: экономический смысл результатов, получаемых при математическом моделировании финансово-экономических задач, и методы их интерпретации для

		основании количественные и качественные выводы и рекомендации по принятию финансово-экономических решений	формулировки рекомендаций по принятию финансово-экономических решений уметь: анализировать результаты исследования математических моделей финансово-экономических задач и делать на их основании количественные и качественные выводы и рекомендации по принятию финансово-экономических решений
УК-4	Способность использовать прикладное программное обеспечение при решении профессиональных задач	Использует основные методы и средства получения, представления, хранения и обработки данных	знать: основные методы получения, представления, хранения и обработки данных уметь: применять основные методы получения, представления, хранения и обработки данных
		Демонстрирует владение профессиональными пакетами прикладных программ	знать: профессиональные пакеты прикладных программ уметь: использовать профессиональные пакеты прикладных программ
		Выбирает необходимое прикладное программное обеспечение в зависимости от решаемой задачи	знать: необходимое прикладное программное обеспечение уметь: выбирать необходимое прикладное программное обеспечение в зависимости от решаемых задач
		Использует прикладное программное обеспечение для решения конкретных прикладных задач	знать: назначение прикладного программного обеспечения уметь: использовать прикладное программное обеспечение для решения конкретных прикладных задач
ПКП-4	Способность применять математический аппарат при разработке вычислительных алгоритмов	Владеет математическим аппаратом, необходимым для разработки вычислительных алгоритмов	знать: основные математические, понятия необходимые для разработки вычислительных алгоритмов уметь: применять математический аппарат для разработки вычислительных алгоритмов
		Разрабатывает вычислительные алгоритмы для решения прикладных задач	знать: методы разработки вычислительных алгоритмов для решения прикладных задач уметь: разрабатывать вычислительные алгоритмы для решения прикладных задач
ПКП-5	Способность	Демонстрирует знания в	знать: основные методы

	применять математическое моделирование для решения прикладных задач	области теории и методологии математического моделирования и системного анализа процессов	математического моделирования и системного анализа процессов для решения прикладных задач уметь: решать прикладные задачи с помощью методов математического моделирования
		Строит математические модели, анализирует и интерпретирует результаты моделирования для решения задач прикладного характера, в том числе в части проектирования программного обеспечения	знать: математические модели и методы проектирования программного обеспечения уметь: применять математические модели и интерпретировать результаты моделирования для решения задач прикладного характера, в том числе в части проектирования программного обеспечения

3. Место дисциплины в структуре образовательной программы

Дисциплина «Практикум по программированию» относится к «Модулю дисциплин дополнительной квалификации - программист».

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

Очная форма обучения

Вид учебной работы по дисциплине	Всего (в з/е и часах)	Семестр 3 (в часах)	Семестр 4 (в часах)
Общая трудоёмкость дисциплины	4/144	72	72
Контактная работа- Аудиторные занятия	32	16	16
Лекции	0		
Семинары, практические занятия	32	16	16
Самостоятельная работа	112	56	56
Вид текущего контроля	Проектная работа, Проектная работа	Проектная работа	Проектная работа
Вид промежуточной аттестации	Зачет, Зачет	Зачет	Зачет

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. Основы языка Python

Обработка числовой информации. Встроенные функции и модули для работы с числами. Реализация числовых алгоритмов с использованием инструкции ветвления и циклов.

Обработка текстовой информации. Функции и методы для работы со строками. Регулярные выражения.

Списки. Использование списков для хранения информации. Методы списков. Многомерные списки. Кортежи. Словари. Типовые случаи использования словарей в программах. Методы для работы со словарями. Множества и методы работы с ними.

Тема 2. Функции и модули

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

Глобальные и локальные переменные. Вложенные функции. Функции высшего порядка.

Создание и использование модулей и пакетов. Модули стандартной библиотеки.

Тема 3. Обработка исключений. Работа с файлами средствами языка Python

Понятие исключения. Инструкция `try ... except ... else ... finally`. Классы встроенных исключений. Инструкции `raise` и `assert`. Инструкция `with ... as`.

Работа с файлами в Python. Концепция файла в современных операционных системах и языках программирования. Операции с файлами: открытие, закрытие, чтение, запись. Инструкция `with ... as`.

Использование текстовых и двоичных файлов. Хранение и обработка данных в файлах `pickle` и `shelve`. Работа с файлами различных форматов: `csv`, `xml`, `json`, `xlsx`.

Тема 4. Объектно-ориентированное программирование на Python

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

Методы классов, статические переменные и методы. Управление доступом к атрибутам класса. Динамические операции с атрибутами и интроспекция. Использование специальных методов для расширенного функционала пользовательских классов. Иерархия классов.

Тема 5. Функциональное программирование на Python

Элементы функционального программирования в Python: функции "граждане первого класса", функции высшего порядка, замыкания, функции без побочных эффектов, рекурсия. Неизменяемые структуры данных. Идиомы, распространенные в функциональных языках программирования: итераторы, последовательности, ленивые вычисления.

Декораторы в Python: использование и создание собственных декораторов.

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

Итераторы в Python, итерируемый тип данных. Модуль `itertools`. Функции-генераторы и выражения-генераторы в Python.

Тема 6. Алгоритмы и структуры данных

Динамические массивы. Стеки, очереди, деки. Связные списки. Реализация связных списков на языке Python. Бинарные деревья. Использование бинарных деревьев в прикладных задачах. Двоичное дерево поиска.

Асимптотическая оценка сложности алгоритма. Алгоритмы сортировки и поиска. Бинарный поиск. Простые методы сортировки: обменные сортировки, сортировка выбором, сортировка вставками. Сортировка Шелла. Быстрая сортировка. Сортировка слиянием.

Тема 7. Проектирование приложений

Основные принципы объектно-ориентированного проектирования приложений. Принцип абстракции. Уменьшение зависимости. Основные шаблоны проектирования, интерфейс, делегирование. Порождающие шаблоны: фабричный метод, абстрактная фабрика, строитель. Структурные шаблоны: адаптер, мост, декоратор, фасад.

Поведенческие шаблоны: цепочка обязанностей, команда, посредник, наблюдатель, состояние, стратегия, посетитель, шаблонный метод.

Тема 8. Программирование графических интерфейсов

Событийно-ориентированное программирование. События и обработчики событий. События мыши и клавиатуры. Основные библиотеки графических интерфейсов в Python: tkinter, PyQt, PyGTK. Главный цикл программы. Основные элементы управления: кнопки, ползунки, поля ввода. Работа с графикой.

Тема 9. Системное программирование на языке Python

Чтение и запись файлов. Работа с путями в Windows и Linux. Обход папок. Работа с архивами. Работа с офисными форматами. Работа со структурированными данными. Основы работа с реляционными базами данных в Python.

Тема 10. Сетевое программирование на языке Python

Получение и разбор HTML-страниц. Библиотеки HTML-парсинга. Сокеты. Клиент-серверные приложения. Обращение к внешним API. Многопоточность. Библиотеки многопоточности и многопроцессности. Отправка и получение электронных писем.

Тема 11. Тестирование программ на языке Python

Основные виды тестирования программного обеспечения. Модульное и интеграционное тестирование. Понятие регрессии. Фиксирование и формализация требований. Библиотеки автоматизированного тестирования. Написание автоматических модульных тестов по техническому заданию.

Разработка через тестирование. Проектирование через тестирование. Методология TDD.

Тема 12. Документирование и развертывание программ на языке Python

Основные приемы документирования программного кода. Соглашения о стиле документирования кода. Написание программной документации в формате docstring. Языки форматирования документации: ReST, Markdown. Автоматическая сборка документации с использованием Sphinx. Экспорт документации в популярные форматы.

5.2. Учебно-тематический план

№ п/п	Наименование тем(разделов) дисциплины	Трудоемкость в часах				
		Всего	Контактная работа - Аудиторная работа			Самостоя тельная работа
			Общая, в т.ч.:	Лекции	Семинары, практическ ие занятия	
1	Основы языка Python	10	2	0	2	8
2	Функции и модули	12	2	0	2	10
3	Обработка исключений. Работа с файлами средствами языка Python	10	2	0	2	8
4	Объектно-ориентированное программирование на Python	12	2	0	2	10
5	Функциональное программирование на Python	14	4	0	4	10
6	Алгоритмы и структуры данных	14	4	0	4	10
7	Проектирование приложений	12	2	0	2	10
8	Программирование графических интерфейсов	14	4	0	4	10
9	Системное программирование на языке Python	12	2	0	2	10
10	Сетевое программирование на языке Python	14	4	0	4	10
11	Тестирование программ на языке Python	10	2	0	2	8
12	Документирование и развертывание программ на языке Python	10	2	0	2	8
	Итого	144	32	0	32	112

5.3. Содержание семинаров

Наименование тем (разделов) дисциплины	Перечень вопросов для обсуждения на семинарах, практических занятиях	Формы проведения занятий
Основы языка Python	Решение задач с использованием инструкции ветвления. Решение задач с использованием циклов. Создание и обработка списков. Многомерные списки. Обработка текстовой информации. Использование словарей и множеств. Совместное использование различных типов данных.	Интерактивная форма, работа на компьютере.
Функции и модули	Создание и использование функций. Анонимные функции. Функции высшего порядка.	Интерактивная форма, работа на компьютере.
Обработка исключений. Работа с файлами средствами языка Python	Ошибки и исключения в программировании, классы исключений. Обработка встроенных исключений. Создание и обработка пользовательских исключений. Работа с текстовыми файлами. Работа с двоичными файлами. Сохранение объектов в файл.	Интерактивная форма, работа на компьютере.
Объектно-ориентированное программирование на Python	Создание классов и объектов. Наследование и полиморфизм. Специальные методы классов.	Интерактивная форма, работа на компьютере.
Функциональное программирование на Python	Функции map, filter, reduce, any, all. Декораторы. Итераторы. Выражения-генераторы. Итераторы, итерируемый тип данных.	Интерактивная форма, работа на компьютере.
Алгоритмы и структуры данных	Создание связанных списков. Использование стеков и очередей при решении задач. Алгоритмы сортировки и поиска.	Интерактивная форма, работа на компьютере.
Проектирование приложений	Создание примеров программных реализаций для распространенных шаблонов проектирования.	Интерактивная форма, работа на компьютере.
Программирование графических интерфейсов	Создание графического приложения с использованием встроенных возможностей языка и библиотек.	Интерактивная форма, работа на компьютере.
Системное программирование на языке Python	Автоматизация рутинных административных задач. Создание Web-парсера.	Интерактивная форма, работа на компьютере.
Сетевое программирование на языке Python	Создание многопоточного многопользовательского многофункционального сервера на сокетах.	Интерактивная форма, работа на компьютере.
Тестирование программ на языке Python	Написание модульных тестов по описанию. Разработка программы по тестам.	Интерактивная форма, работа на компьютере.
Документирование и развертывание	Документирование программы по описанию и исходному коду. Экспорт документации в формате	Интерактивная форма, работа на компьютере.

программ на языке Python	сайта.	
-----------------------------	--------	--

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

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

Наименование тем (разделов) дисциплины	Перечень вопросов, отводимых на самостоятельное освоение	Формы внеаудиторной самостоятельной работы
Основы языка Python	Функции модулей math, random, copy. Регулярные выражения. Кортежи. Множества.	Работа с учебной литературой и документацией. Решение задач с использованием Jupyter Notebook. Выполнение домашних заданий.
Функции и модули	Создание и использование модулей и пакетов.	Работа с учебной литературой и документацией. Решение задач с использованием Jupyter Notebook. Выполнение домашних заданий.
Обработка исключений. Работа с файлами средствами языка Python	Работа с файлами в форматах htm, xml, csv, pdf, xlsx. Создание и обработка пользовательских исключений.	Работа с учебной литературой и документацией. Решение задач с использованием Jupyter Notebook. Выполнение домашних заданий.
Объектно-ориентированное программирование на Python	Специальные методы классов.	Работа с учебной литературой и документацией. Решение задач с использованием Jupyter Notebook. Выполнение домашних заданий.
Функциональное программирование на Python	Функции any, all, zip.	Работа с учебной литературой и документацией. Решение задач с

		использованием Jupyter Notebook. Выполнение домашних заданий.
Алгоритмы и структуры данных	Бинарные деревья. Использование бинарных деревьев в прикладных задачах. Двоичное дерево поиска.	Работа с учебной литературой и документацией. Решение задач с использованием Jupyter Notebook. Выполнение домашних заданий.
Проектирование приложений	Внедрение зависимости, инверсия управления.	Работа с учебной литературой и документацией. Решение задач с использованием Jupyter Notebook. Выполнение домашних заданий.
Программирование графических интерфейсов	Создание игр. Библиотека PyGame.	Работа с учебной литературой и документацией. Решение задач с использованием Jupyter Notebook. Выполнение домашних заданий.
Системное программирование на языке Python	Создание административных скриптов.	Работа с учебной литературой и документацией. Решение задач с использованием Jupyter Notebook. Выполнение домашних заданий.
Сетевое программирование на языке Python	Создание веб-сервиса.	Работа с учебной литературой и документацией. Решение задач с использованием Jupyter Notebook. Выполнение домашних заданий.
Тестирование программ на языке Python	Экстремальное программирование.	Работа с учебной литературой и документацией. Решение задач с

		использованием Jupyter Notebook. Выполнение домашних заданий.
Документирование и развертывание программ на языке Python	Непрерывная интеграция. Экспорт документации.	Работа с учебной литературой и документацией. Решение задач с использованием Jupyter Notebook. Выполнение домашних заданий.

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

Примерные задания проектной работы

1. Реализовать программу, с которой можно играть в логическую игру «Быки и коровы» (описание правил игры: <http://робомозг.рф/Articles/BullsAndCowsRules>). Программа загадывает число, пользователь вводит очередной вариант отгадываемого числа, программа возвращает количество быков и коров и в случае выигрыша игрока сообщает о победе и завершается. Сама программа не ходит, т. е. не пытается отгадать число загаданное игроком. Взаимодействие с программой производится через консоль, при запросе данных от пользователя программа сообщает, что ожидает от пользователя и проверяет корректность ввода.

2. Реализовать программу, при помощи которой два игрока могут играть в «Крестики-нолики» на поле 3 на 3. Взаимодействие с программой производится через консоль. Игровое поле изображается в виде трех текстовых строк и перерисовывается при каждом изменении состояния поля. При запросе данных от пользователя программа сообщает, что ожидает от пользователя (в частности, координаты новой отметки на поле) и проверяет корректность ввода. Программа должна уметь автоматически определять, что партия окончена, и сообщать о победе одного из игроков или о ничьей. Сама программа не ходит, т. е. не пытается ставить крестики и нолики с целью заполнить линию.

3. Реализовать программу, при помощи которой два игрока могут играть в игру «Супер ним». Правила игры следующие. На шахматной доске в некоторых клетках случайно разбросаны фишки или пуговицы. Игроки ходят по очереди. За один ход можно снять все фишки с какой-либо горизонтали или вертикали, на которой они есть. Выигрывает тот, кто заберет последние фишки. (описание правил игры: <https://habr.com/ru/articles/764674/?ysclid=mia9aaq6gy516732502>). Взаимодействие с программой производится через консоль. Игровое поле изображается в виде текстовых строк и перерисовывается при каждом изменении состояния поля. При запросе данных от пользователя программа сообщает, что ожидает от пользователя (в частности, координаты новой отметки на поле) и проверяет корректность ввода. Программа должна уметь автоматически определять, что партия окончена, и сообщать о победе одного из игроков. Сама программа не ходит, т. е. не пытается выбирать строки или столбцы с целью победить в игре.

4. Написать калькулятор для строковых выражений вида '<число> <операция> <число>', где <число> – неотрицательное целое число меньше 100, записанное словами, например "тридцать четыре", <арифметическая операция> – одна из операций "плюс", "минус", "умножить". Результат выполнения операции вернуть в виде текстового представления числа. Пример `calc("двадцать пять плюс тринадцать")` => "тридцать восемь". Оформить калькулятор в виде функции, которая принимает на вход строку и возвращает строку.

На основе модулей `csv`, `pickle` и прямой работы с файлами:

5. Создать функции `load_table`, `save_table` по загрузке/сохранению табличных данных во внутреннее представление модуля / из внутреннего представления модуля:

- файла формата `csv` (отдельный модуль с `load_table`, `save_table` в рамках общего пакета);

- файла формата `pickle` (отдельный модуль с `load_table`, `save_table` в рамках общего пакета), модуль использует удобную для автора работы структуру таблицы;

- текстового файла (только функция `save_table`, сохраняющая в текстовом файле представление таблицы, аналогичное выводу на печать с помощью функции `print_table`).

Примечание: внутреннее представление может базироваться на словаре, где по разным ключам хранятся ключевые «атрибуты» таблицы, а значения таблицы хранятся в виде вложенных списков. Студент может выбрать другое внутреннее представление таблицы (согласовав его с преподавателем), в том числе, студенты знакомые с ООП на Python, могут реализовать собственный класс для таблицы. При определении API модулей максимально полно использовать возможности сигнатур функций на Python (значения по умолчанию, упаковка/распаковка, в т. ч. именованных параметров, возвращение множественных значений), интенсивно выполнять проверки и возбуждать исключительные ситуации.

6. Реализовать программу, которая позволяет играть в шахматы на компьютере. Взаимодействие с программой производится через консоль (базовый вариант). Игровое поле изображается в виде 8 текстовых строк, плюс строки с буквенным обозначением столбцов (см. пример на Рис. 1) и перерисовывается при каждом изменении состояния поля. При запросе данных от пользователя программа

сообщает, что ожидает от пользователя (например, позицию фигуры для следующего хода белыми; целевую позицию выбранной фигуры) и проверяет корректность ввода (допускаются только ходы соответствующие правилам шахмат; поддержка рокировки, сложных правил для пешек и проверки мата вынесена в отдельные пункты). Программа должна считать количество сделанных ходов. Сама программа не ходит: т. е. не пытается выполнить ходы за одну из сторон, а предоставляет поочередно вводить ходы за белых и черных.

7. Реализовать на языке Python задачу математического программирования по определению оптимального маршрута движения коммивояжера, цель которого состоит в том, чтобы посетить все объекты, записанные в задании, за кратчайший срок и с наименьшими затратами. В теории графов Задача коммивояжера – это поиск пути, связывающего два или более узла, с использованием критерия оптимальности. Существует несколько способов решения задачи коммивояжера:

- Метод перебора – вычисляются все возможные пути, а затем сравниваются. Количество путей в графе, состоящем из n городов, равно $n!$ Решение при таком подходе требует больших вычислительных затрат;
- Метод ветвей и границ – проблема разбивается на подзадачи. Решение отдельных подзадач обеспечит оптимальное решение;
- Динамическое программирование – метод поиска оптимальных решений путем анализа всех возможных маршрутов;
- Метод ближайшего соседа – эвристически жадный подход, при котором мы выбирается соседний узел. Вычислительно менее затратен, чем динамический подход, но не дает гарантии оптимального решения.

8. Реализация собственного пакета модулей для манипулирования плоскими фигурами. Базовые требования к функциональности программы.

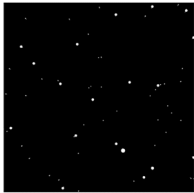
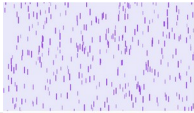
Реализовать API, который позволяет генерировать, преобразовывать и визуализировать последовательность плоских полигонов, представленных в виде кортежа кортежей (например: $((0,0), (0,1), (1,1), (1,0))$ – представление для квадрата). Последовательности представлений полигонов представляют из себя итераторы (далее: последовательности полигонов). Решать задачи с использованием функционального стиля программирования, в том числе активно использовать функции из модуля *itertools* и *functools*.

Критерии балльной оценки различных форм текущего контроля успеваемости содержатся в соответствующих методических рекомендациях Кафедры информационных технологий.

Дополнительная информация

1. Простая анимация. Цель работы: освоить приемы работы с графической библиотекой, а именно: построение графических примитивов, создание анимации, управление главным циклом работы программы.

Все задания подразумевают творческий подход к их выполнению. Задание описывает только общую канву программы. В процессе выполнения студент может столкнуться с рядом дизайнерских решений – выбор скорости анимации, размера элементов, цветового решения и так далее. Все это ложится на студента как на разработчика программы. По умолчанию предусматривается использование библиотеки tkinter. По желанию студента и по согласованию с преподавателем можно выбрать другой подобный инструмент решения задачи.

Звездное поле В произвольном месте на экране появляются звезды, которые движутся по направлению от центра экрана. Должно складываться ощущение движения наблюдателя сквозь звездное поле. Сделайте звезды разного размера и цветов. 1. Реализуйте движение в 3D, где размер звезды зависит от расстояния до нее. 2. Скорость движения регулируется с клавиатуры. 3. Мышкой можно регулировать направление движения.	
Дождь На экране постоянно генерируются контрастные “капли”, равномерно движущиеся вниз. 1. Поэкспериментируйте с плотностью генерации капель. 2. Капли должны создавать ощущение трехмерности – дальние движутся медленнее и меньше в размерах.	
Солнечная система В центре экрана находится неподвижное солнце. Вокруг него вращаются несколько планет по круговым орбитам с разной скоростью и на разном расстоянии. 1. Попробуйте добавить прозрачность. 2. У планет могут быть собственные спутники. 3. Добавьте пояс астероидов.	

2. Сложная анимация

Цель работы – применить полученные навыки работы с графикой и анимацией для выполнения более сложных проектов.

1. Фейерверки. На экране случайно генерируются фейерверки. Сам фейерверк и его частицы имеют ограниченное время жизни. Добавьте генерацию фейерверков разных цветов и прозрачность. Поэкспериментируйте с гравитацией частиц.
2. Притяжение. Реализуйте систему частиц, каждая из которых притягивается к определенному месту на экране. Одновременно, каждая частица отталкивается от текущих координат мышки. Должно получиться, что курсор "разбрасывает" частицы от себя. Нарисуйте местами притяжения частиц какой-нибудь рисунок или надпись. Реализуйте импорт рисунка из внешнего файла.
3. Анимация круга. С помощью полярных координат реализующее превращение круга в треугольник как на приведенной анимации
4. Папоротник Барнсли. При помощи интерактивного алгоритма постройте из точек фрактальную фигуру. Манипулируя параметрами уравнения, получите другие виды фрактальных растений.

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

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

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

ПKN-3 Способность осуществлять сбор, обработку и статистический анализ данных, применять математические методы для решения стандартных профессиональных финансово-экономических задач, интерпретировать полученные результаты

1) Проводит сбор, обработку и статистический анализ данных для решения финансово-экономических задач

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: методы сбора, обработки и статистического анализа данных, применяемые при решении финансово-экономических задач

Уметь: выбирать и применять методы сбора, обработки и статистического анализа данных при решении финансово-экономических задач

Типовые контрольные задания

Выполните обработку структурированного файла о результатах торгов акциями конкретного предприятия на московской бирже за несколько месяцев. Рассчитайте показатели описательной статистики. Сформулируйте рекомендации о целесообразности включения пакета акций предприятия в инвестиционный портфель.

2) Формулирует математические постановки финансово-экономических задач, переходит от экономических постановок задач к математическим моделям

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: математические понятия, используемые при построении математических моделей экономических задач

Уметь: формулировать математические постановки финансово-экономических задач и переходить от экономических постановок задач к математическим моделям

Типовые контрольные задания

Найти кратчайший путь в графе с условием посещения всех узлов только один раз и возвращения в исходный узел. Необходимо вывести на экран кратчайший путь.

Необходимо посетить несколько населённых пунктов. Известны денежные затраты на дорогу между каждыми двумя пунктами. Составить маршрут так, чтобы общие затраты были наименьшими.

3) Системно подходит к выбору математических методов и информационных технологий для решения конкретных финансово-экономических задач в профессиональной области

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: особенности математических методов и информационных технологий, используемых для решения финансово-экономических задач в профессиональной области

Уметь: обоснованно выбирать математические методы и информационные технологии для решения конкретных финансово-экономических задач в профессиональной области

Типовые контрольные задания

Пусть необходимо выполнить n операций, используя m механизмов (по одному для каждой операции). Известно время выполнения каждой операции каждым механизмом. Распределить операции между механизмами так, чтобы общее время выполнения операций было наименьшим.

4) Анализирует результаты исследования математических моделей финансово-экономических задач и делает на их основании количественные и качественные выводы и рекомендации по принятию финансово-экономических решений

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: экономический смысл результатов, получаемых при математическом моделировании финансово-экономических задач, и методы их интерпретации для формулировки рекомендаций по принятию финансово-экономических решений

Уметь: анализировать результаты исследования математических моделей финансово-экономических задач и делать на их основании количественные и качественные выводы и рекомендации по принятию финансово-экономических решений

Типовые контрольные задания

Рассмотрим суждение: "Если цены высокие, то и заработная плата должна быть высокой. Цены высокие или применяется регулирование цен. Если применяется регулирование цен, то нет инфляции. Инфляция есть. Следовательно, заработная плата должна быть высокой". Проанализировать это суждение в терминах алгебры высказываний и проверить его корректность.

УК-4 Способность использовать прикладное программное обеспечение при решении профессиональных задач

1) Использует основные методы и средства получения, представления, хранения и обработки данных

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: основные методы получения, представления, хранения и обработки данных

Уметь: применять основные методы получения, представления, хранения и обработки данных

Типовые контрольные задания

На специализированном сайте получить данные в формате CSV об изменении цены акций конкретной организации за определенный период времени. Импортировать эти данные в среду программирования в виде объекта типа DataFrame. Привести столбцы таблицы с ценой акций к числовому типу.

2) Демонстрирует владение профессиональными пакетами прикладных программ

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: профессиональные пакеты прикладных программ

Уметь: использовать профессиональные пакеты прикладных программ

Типовые контрольные задания

Разработать Python-скрипт, который автоматизирует обработку данных о продажах. Входные данные: Файл sales_data.xlsx (Дата, Товар, Регион, Цена, Количество). С помощью библиотеки pandas загрузить данные, удалить дубликаты и рассчитать столбец «Выручка» (Цена × Количество). Сгруппировать данные по регионам, вычислив общую сумму продаж. Используя библиотеку openpyxl или xlswriter, сохранить итоговую таблицу в файл report.xlsx. Программно применить денежный формат к столбцам с суммами. Построить столбчатую диаграмму продаж по регионам внутри Excel-файла.

3) Выбирает необходимое прикладное программное обеспечение в зависимости от решаемой задачи

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: необходимое прикладное программное обеспечение

Уметь: выбирать необходимое прикладное программное обеспечение в зависимости от решаемых задач

Типовые контрольные задания

Разработать программную систему, состоящую из двух модулей: 1) Frontend-модуль (Python): Отвечает за интерфейс с пользователем, чтение исходных данных из файла, валидацию данных, запуск вычислительного модуля и визуализацию/сохранение результата; 2) Backend-модуль (Компилируемый язык): Отвечает за реализацию ресурсоемкого алгоритма обработки данных.

4) Использует прикладное программное обеспечение для решения конкретных прикладных задач

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: назначение прикладного программного обеспечения

Уметь: использовать прикладное программное обеспечение для решения конкретных прикладных задач

Типовые контрольные задания

Разработать программу (скрипт или десктопное приложение), которое автоматизирует рутинный бизнес-процесс. Программа должна получать «сырые» данные, обрабатывать их и выгружать результат в формате, удобном для конечного пользователя (Excel-таблица с графиками или PDF-отчет). Входные данные: CSV-файл выгрузки из кассового аппарата (Дата, Товар, Категория, Цена, Кол-во). Рассчитать общую выручку по каждой категории товаров. Найти «Топ-3» самых продаваемых товаров месяца.

ПКП-4 Способность применять математический аппарат при разработке вычислительных алгоритмов

1) Владеет математическим аппаратом, необходимым для разработки вычислительных алгоритмов

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: основные математические, понятия необходимые для разработки вычислительных алгоритмов

Уметь: применять математический аппарат для разработки вычислительных алгоритмов

Типовые контрольные задания

Определить, является ли система функций полной. Если не является, дополнить её до полной наименьшим возможным количеством функций.

2) Разрабатывает вычислительные алгоритмы для решения прикладных задач

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: методы разработки вычислительных алгоритмов для решения прикладных задач

Уметь: разрабатывать вычислительные алгоритмы для решения прикладных задач

Типовые контрольные задания

Построить машину Тьюринга, которая правильно вычисляет для всех целых неотрицательных значений аргумента x функцию $f(x) = x \bmod 2$.

ПКП-5 Способность применять математическое моделирование для решения прикладных задач

1) Демонстрирует знания в области теории и методологии математического моделирования и системного анализа процессов

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: основные методы математического моделирования и системного анализа процессов для решения прикладных задач

Уметь: решать прикладные задачи с помощью методов математического моделирования

Типовые контрольные задания

Компания оценивает запуск нового производства сроком на 5 лет. Из-за нестабильности рынка ключевые финансовые показатели не являются фиксированными. Начальные инвестиции варьируются в заданном диапазоне, а ежегодный денежный поток зависит от случайного объема продаж и инфляции. Разработать модель оценки стоимости проекта. Задать вероятностные распределения для стартовых затрат и ежегодных доходов. Реализовать алгоритм дисконтирования будущих денежных потоков по заданной ставке.

2) Строит математические модели, анализирует и интерпретирует результаты моделирования для решения задач прикладного характера, в том числе в части проектирования программного обеспечения

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: математические модели и методы проектирования программного обеспечения

Уметь: применять математические модели и интерпретировать результаты моделирования для решения задач прикладного характера, в том числе в части проектирования программного обеспечения

Типовые контрольные задания

Разработайте систему управления умным домом, которая должна быть расширяемой: пользователи могут докупать новые устройства разных производителей (лампочки, термостаты, датчики движения, умные замки), и система должна уметь с ними работать без переписывания основного кода ядра.

Примеры практико-ориентированных заданий

3 семестр

1. Чему будет равно значение переменной A после выполнения оператора: $A = [i + 10 \text{ for } i \text{ in range}(5, 0, -1)]$
2. Имеется строка $S = "1234567890"$. Чему равно значение S1, если $S1 = S[1:2] + S[4:-1]$
3. Используя генератор словарей, для заданного натурального числа n создайте словарь D, в котором будет ровно n элементов. Элементами словаря являются $\{ 's1': 10, 's2': 20, 's3': 30, \dots \}$, т. е. значение равно номеру элемента (нумерация с 1), умноженному на 10, а ключ состоит из символа 's', к которому дописан номер.
4. Имеется список L вида $[[1, 3, 65], [11, 5, 6], [15, 33, 11]]$. Отсортируйте список по возрастанию последнего значения элемента.
5. После выполнения приведенного кода будет напечатано ...

```
def data(d=[]):
```

```
d.append(1)
```

```
return d
```

```
a = data()
b = data()
c = data([3])
print(b, c)
```

a) [1] [3]

b) [2] [4]

c) [1,1] [3,1]

d) [1,2] [3,4]

e) сообщение об ошибке

6. После выполнения приведенного кода будет напечатано ...

```
class Class1:
    def __init__(self, n):
        self._x = n
    def gx(self):
        return self._x+2
x = property(gx)
c = Class1(2)
print(c._x)
```

a) 2

b) 4

c) None

d) Возникнет ошибка

4 семестр

1. Приведите пример приватного IP-адреса

a. 196.168.0.1

b. 127.0.0.10

c. 87.250.250.242

d. 173.194.73.113

2. Приведите пример публичного IP-адреса

a. 8.8.8.8

b. 127.0.0.1

c. 196.168.0.1

d. 10.38.51.16

3. Соглашение о порядке и способе связи между компьютерами это:

a. сетевой протокол

b. сетевой интерфейс

c. коммутация

d. адресация

4. Как называется самый распространенный стандарт на архитектуру локальных сетей на основе кабельного соединения

a. Ethernet

b. Wi-Fi

c. TCP

d. GlobalNet X

5. Приведите пример стандарта из серии IEEE 802

a. Wi-Fi

b. витая пара

c. POSIX

d. USB

6. Назовите команду, отображающую основную информацию о сетевых подключениях в ОС Linux

a. ipconfig

b. ifconfig

c. netstat

d. traceroute

7. Назовите команду, отображающую основную информацию о сетевых подключениях в ОС Windows

a. ifconfig

b. ipconfig

c. netstat

d . tracert

8. Назовите пример сетевой топологии

a. звезда

b. снежинка

c. лабиринт

d. круговая

Примерные вопросы для подготовки к зачету, зачету

3 семестр

1. Функции модуля math

2. Инструкция if...else

3. Инструкция цикла while

4. Инструкция цикла for

5. Функция range

6. Строки. Операции над строками (+, *, in, доступ по индексу, получение среза).
Функция len

7. Методы строк: strip, find, count, replace
8. Списки в Python. Создание: создание пустого списка, методы append, split, функция list. Генераторы списков
9. Создание копии списка (срезы, функции list и deepcopy)
10. Основные операции над списками (+, *, in, доступ по индексу). Перебор элементов списка
11. Добавление и удаление элементов списка (методы append, insert, pop, remove). Методы reverse, join
12. Сортировка списков. Параметры метода sort: key, reverse
13. Кортежи. Создание. Создание пустого кортежа и кортежа из одного элемента. Операции (+, *, in, доступ по индексу, получение среза)
14. Функции tuple, len
15. Распаковка последовательности
16. Словари. Создание словаря. Функции dict, zip
17. Операции над словарями (in, del)
18. Перебор элементов словаря. Методы get, clear, copy, keys, values
19. Множества. Создание. Функция set
20. Операции над множествами: in, |, &, -, ^, <=, >=, <, >, ==. Методы add, discard
21. Создание и вызов функции
22. Передача аргументов в функцию. Необязательные параметры функций. Функции в качестве аргументов
23. Глобальные и локальные переменные
24. Анонимные функции
25. Создание и использование модулей. Инструкции import и from
26. Пакеты. Использование пакетов
27. Инструкция try ... except ... else ... finally
28. Инструкции raise и assert

- 29. Инструкция with ... as
- 30. Классы встроенных исключений
- 31. Работа с текстовыми файлами. Открытие файла (функция open) и закрытие файла (метод close). Чтение текстового файла (методы read, readline, readlines). Перебор строк файла в цикле for. Запись в текстовый файл (метод write, функция print с параметром file)
- 32. Сохранение объектов в файл (функции dump и load модуля pickle)
- 33. Понятие класса и объекта. Определение класса и создание экземпляра класса
- 34. Методы класса. Параметр self. Статические методы. Закрытые методы
- 35. Атрибуты класса и экземпляра класса. Доступ к атрибуту. Закрытые атрибуты
- 36. Свойства. Создание и использование свойства
- 37. Наследование. Базовый и производный классы. Переопределение методов
- 38. Специальные методы. Перегрузка операторов
- 39. Функции map, filter, reduce, any, all
- 40. Декораторы функций
- 41. Итераторы
- 42. Функции-генераторы
- 43. Массивы. Использование массивов
- 44. Стеки. Использование стеков. Реализация стека
- 45. Очереди. Использование очереди. Реализация очереди
- 46. Связные списки. Использование связанных списков. Реализация связанных списков
- 47. Бинарные деревья. Использование бинарных деревьев. Реализация бинарных деревьев
- 48. Бинарный поиск
- 49. Обменные сортировки
- 50. Сортировка Шелла
- 51. Быстрая сортировка

4 семестр

1. Шаблон проектирования: интерфейс
2. Шаблон проектирования: делегирование
3. Шаблон проектирования: фабричный метод
4. Шаблон проектирования: абстрактная фабрика
5. Шаблон проектирования: строитель
6. Шаблон проектирования: адаптер
7. Шаблон проектирования: мост
8. Шаблон проектирования: декоратор
9. Шаблон проектирования: фасад
10. Шаблон проектирования: цепочка обязанностей
11. Шаблон проектирования: команда
12. Шаблон проектирования: посредник
13. Шаблон проектирования: наблюдатель
14. Шаблон проектирования: состояние
15. Шаблон проектирования: стратегия
16. Шаблон проектирования: посетитель
17. Шаблон проектирования: шаблонный метод
18. Событийно-ориентированное программирование
19. События и обработчики событий. Виды событий
20. События мыши и клавиатуры
21. Модули tkinter, PyQt, PyGTK – особенности и отличия
22. Главный цикл программы
23. Основные элементы управления: кнопки, ползунки, поля ввода
24. Работа с графикой

25. Работа с путями в Windows и Linux
26. Основные форматы хранения данных: CSV, XML, JSON
27. Получение и разбор HTML-страниц
28. Библиотеки HTML-парсинга
29. Сокеты
30. Клиент-серверные приложения
31. Обращение к внешним API
32. Многопоточность
33. Загруженность ребра сетевой структуры
34. Библиотеки многопоточности и многопроцессности
35. Отправка и получение электронных писем
36. Основные виды тестирования программного обеспечения
37. Модульное и интеграционное тестирование
38. Понятие регрессии и регрессионного тестирования
39. Библиотеки автоматизированного тестирования
40. Разработка через тестирование

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

Нормативные правовые акты:

Федеральный закон от 27.07.2006 № 152-ФЗ «О персональных данных»

Федеральный закон от 27.07.2006 № 149-ФЗ «Об информации, информационных технологиях и о защите информации»

Федеральный закон от 06.04.2011 № 63-ФЗ «Об электронной подписи»

Основная литература:

1. Гуриков, Сергей Ростиславович Основы алгоритмизации и программирования на Python : Учебное пособие / Московский технический университет связи и информатики 1 Москва : ООО "Научно-издательский центр ИНФРА-М", 2025 343 с. (Высшее образование) ВО - Бакалавриат <https://znanium.ru/catalog/document?id=453296> ISBN 978-5-16-020255-6 ISBN 978-5-16-102278-8 (электр. издание) . [БИК ID: RU\infra-m\znanium\bibl\2166199]

2. Практикум по программированию на языке Python : учебное пособие для студентов, обучающихся по направлениям "Прикладная математика и информатика", "Прикладная информатика", "Информационная безопасность", "Бизнес-информатика" (программа подготовки бакалавра) / Р.И. Горохова, Е.П. Догадина, В.И. Долгов, В.И. Макрушин ; Финуниверситет, Департамент анализа данных и машинного обучения факультета информационных технологий и анализа больших данных Москва : Финуниверситет, 2023 1 файл (6,93 Мб) Свободный доступ из сети Интернет (чтение, печать, копирование) (дата обращения 09.10.2023) + Текст : электронный. [БИК ID: RU\FA\books\137296]

3. Федоров, Дмитрий Юрьевич Программирование на python : учебное пособие для вузов / Д. Ю. Федоров. 6-е изд., пер. и доп Электрон. дан. Москва : Юрайт, 2025 187 с (Высшее образование) URL: <https://urait.ru/bcode/556864> (дата обращения: 08.04.2025). Режим доступа: Электронно-библиотечная система Юрайт, для авториз. пользователей <https://urait.ru/bcode/556864> ISBN 978-5-534-19666-5 : 829.00. [БИК ID: RU2fURAIT2f556864]

Дополнительная литература:

1. Карякин, Михаил Игоревич Технологии программирования и компьютерный практикум на языке Python : Учебное пособие / Южный федеральный университет

Ростов-на-Дону : Издательство Южного федерального университета (ЮФУ), 2022 242 с. ВО - Бакалавриат <https://znanium.com/catalog/document?id=429844> ISBN 978-5-9275-4108-9. [БИК ID: RU\infra-m\znanium\bibl\2057604]

2. Чернышев, Станислав Андреевич Основы программирования на Python : учебник для вузов / С. А. Чернышев. 2-е изд., пер. и доп Электрон. дан. Москва : Юрайт, 2025 349 с (Высшее образование) URL: <https://urait.ru/bcode/567821> (дата обращения: 01.09.2025). Режим доступа: Электронно-библиотечная система Юрайт, для авториз. пользователей <https://urait.ru/bcode/567821> ISBN 978-5-534-17139-6 : 1729.00. [БИК ID: RU2fURAIT2f567821]

3. Коротеев, М.В. Сетевое программирование на Python в Linux : учебное пособие по дисциплине "Машинное обучение" для студентов, обучающихся по направлению: "Прикладная информатика", всех профилей (программы подготовки бакалавров), институт онлайн-образования / М.В. Коротеев, Д.А. Петросов ; Финуниверситет, Департамент анализа данных и машинного обучения факультета информационных технологий и анализа больших данных Москва : Финуниверситет, 2023 1 файл (2,01 Мб) : ил. Доступ из локальной сети Финуниверситета (чтение) (дата обращения 16.10.2023) + Текст : электронный. [БИК ID: RU\FA\books\137316]

4. Букунов, С. В. Разработка приложений с графическим пользовательским интерфейсом на языке Python [Электронный ресурс] : учебное пособие для вузов / Букунов С. В.,Букунова О. В. ; Букунова О. В. Санкт-Петербург : Лань, 2023 88 с. Книга из коллекции Лань - Информатика <https://e.lanbook.com/book/292856> ISBN 978-5-507-45191-3. [БИК ID: RU-LAN-BOOK-292856]

5. Янцев, В. В. Web-программирование на Python [Электронный ресурс] : учебное пособие для вузов / Янцев В. В. 3-е изд., перераб. Санкт-Петербург : Лань, 2024 180 с. Книга из коллекции Лань - Информатика <https://e.lanbook.com/book/392993> ISBN 978-5-507-48364-8. [БИК ID: RU-LAN-BOOK-392993]

6. Жулабова, Ф. Т. Системное программирование. Лабораторные работы [Электронный ресурс] : учебное пособие / Жулабова Ф. Т. Санкт-Петербург : Лань, 2020 208 с. Книга из коллекции Лань - Информатика <https://e.lanbook.com/book/140772> ISBN 978-5-8114-4666-7. [БИК ID: RU-LAN-BOOK-140772]

7. Головнин, О. К. Введение в системное программирование и основы жизненного цикла системных программ [Электронный ресурс] / Головнин О. К., Столбова А. А. Самара : Самарский университет, 2021 172 с. Рекомендовано редакционно-издательским советом федерального государственного автономного образовательного учреждения высшего образования «Самарский национальный исследовательский университет имени академика С.П. Королева» в качестве учебного пособия для обучающихся по основной образовательной программе высшего образования по направлению подготовки 09.03.01 Информатика и вычислительная техника Книга из коллекции Самарский университет - Информатика <https://e.lanbook.com/book/257132> ISBN 978-5-7883-1695-6. [БИК ID: RU-LAN-BOOK-257132]

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

1. Электронная библиотека Финансового университета (ЭБ) <http://elib.fa.ru/> (<http://library.fa.ru/files/elibfa.pdf>)
2. Электронно-библиотечная система BOOK.RU <http://www.book.ru>
3. Электронно-библиотечная система "Университетская библиотека ОНЛАЙН" <http://biblioclub.ru/>
4. Научная электронная библиотека eLibrary.ru <http://elibrary.ru>
5. Информационно-образовательный портал Финуниверситета: <https://org.fa.ru>
6. •Электронно-библиотечная система Znanium <http://www.znanium.ru/>
7. •Электронно-библиотечная система издательства «ЮРАЙТ» <https://urait.ru/>
8. Python Documentation <http://python.org/doc/>
9. •Справочная правовая система «ГАРАНТ» <https://www.garant.ru/>
10. Онлайн-курс «Введение в Data Science и машинное обучение» <https://stepik.org/4852>.
11. Бесплатный онлайн курс «Инженер облачных сервисов» <https://practicum.yandex.ru/ycloud/>
12. •Электронные продукты издательства Elsevier <http://www.sciencedirect.com>

10. Методические указания для обучающихся по освоению дисциплины

На семинарских занятиях студенты с помощью преподавателя осваивают учебные материалы и выполняют индивидуальные задания. Преподаватель распределяет задания между студентами. Преподаватель может пояснить задание в части требований к выполнению или стоящей за заданием математической концепцией. Все задания подразумевают творческий подход к их выполнению. После выполнения задания преподаватель может попросить студента провести публичную демонстрацию выполненного задания, включая устные комментарии написанного хода и алгоритмических решений, примененных студентом при решении задачи.

Самостоятельная работа выполняется студентами в классах или медиатеке университета. При этом используется программное обеспечение Jupyter Notebook в составе пакета программ Anaconda, печатные учебные пособия и электронные образовательные ресурсы.

При подготовке к проектным работам студентам следует использовать нормативные документы Финансового университета, Методические рекомендации по планированию и организации внеаудиторной самостоятельной работы студентов по образовательным программам бакалавриата и магистратуры в Финансовом университете, утвержденные приказом Финуниверситета от 11.05.2021 г. № 1040/о (см. сайт Финансового Университета: на главной странице раздел "Наш университет" → "Единая правовая база Финуниверситета" → "Организация учебного процесса" → "Нормативные документы по самостоятельной работе").

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. Python 3.8
2. Jupyter Notebook (Windows 10)
3. Пакет офисных программ
4. Anaconda

5. Kaspersky

Современные профессиональные базы данных и информационные справочные системы:

1. Информационно-правовая система «Гарант»
2. Информационно-правовая система «Консультант Плюс»
3. Электронная энциклопедия: <http://ru.wikipedia.org/wiki/Wiki>
4. Размещение ЭУК: <https://www.campus.fa.ru/>

Сертифицированные программные и аппаратные средства защиты информации:

1. не предусмотрены

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

1. **Учебная аудитория** для проведения учебных занятий, предусмотренных программой, в том числе групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, оснащенная оборудованием и техническими средствами обучения: мебель аудиторная (столы, стулья, доска аудиторная), набор демонстрационного оборудования (проектор, экран)
2. **Помещение для самостоятельной работы** обучающихся оснащенное компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечением доступа к электронной информационно-образовательной среде Университета.
3. **Компьютерный класс** для проведения учебных занятий, предусмотренных программой, в том числе групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, оснащенный оборудованием и техническими средствами обучения: мебель аудиторная (столы, стулья, доска аудиторная), персональные компьютеры, набор демонстрационного оборудования (проектор, экран)
4. Библиотека, читальный зал