МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ,
МИНИСТЕРСТВО НАУКИ, ВЫСШЕГО ОБРАЗОВАНИЯ И ИННОВАЦИЙ
КЫРГЫЗСКОЙ РЕСПУБЛИКИ

МОО ВО Кыргызско-Российский Славянский университет
имени первого Президента Российской Федерации Б.Н. Ельцина



УТВЕРЖДАЮ

Директор Социо-Гуманитарного Института
Голобородько И. Л. _____

_____ 2026 г.

Объектно-ориентированное программирование и визуальное моделирование

рабочая программа дисциплины (модуля)

Закреплена за школой **Международная бизнес-школа**

Учебный план Направление 38.03.01 - РФ, 580100 - КР Экономика
Профиль "Математические методы в экономике"

Квалификация **бакалавр**

Форма обучения **очная**

Общая трудоемкость **3 ЗЕТ**

Часов по учебному плану 108

в том числе:

аудиторные занятия 48

самостоятельная работа 59,9

Виды контроля в семестрах:

зачет с оценкой 6

Распределение часов дисциплины по семестрам

Семестр (<Курс>.<Семестр на курсе>)	6 (3.2)		Итого	
Недель	18			
Вид занятий	УП	РП	УП	РП
Лекции	16	16	16	16
Практические	32	32	32	32
Контактная работа в период теоретического обучения	0,1	0,1	0,1	0,1
В том числе инт.	8	8	8	8
В том числе в форме практ.подготовки	32	32	32	32
Итого ауд.	48	48	48	48
Контактная работа	48,1	48,1	48,1	48,1
Сам. работа	59,9	59,9	59,9	59,9
Итого	108	108	108	108

Программу составил(и):
Доцент, к.т.н., Мусакулова Жылдыз Абдыманаповна _____

Согласовано РОП:
Мокроусов Николай Васильевич _____

Рабочая программа дисциплины
Объектно-ориентированное программирование и визуальное моделирование

разработана в соответствии с ФГОС 3++:
Федеральный государственный образовательный стандарт высшего образования - бакалавриат по направлению подготовки 38.03.01 Экономика (приказ Минобрнауки России от 12.08.2020 г. № 954)

составлена на основании учебного плана:

Направление 38.03.01 - РФ, 580100 - КР Экономика
Профиль "Математические методы в экономике"
утвержденного учёным советом вуза от 26.06.2026 протокол № 11.

Рабочая программа одобрена на заседании Международной бизнес-школы

Протокол от 03.09.2026 г. № 1
Срок действия программы: 2026-2030 уч.г.
Руководитель подразделения: Кочербаева Айнура Анатольевна

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ
к рабочей программе дисциплины (модуля)

№ п/п	Основание внесения изменения	Раздел/ пункт	Содержание изменения
1.			
2.			
3.			
4.			
5.			

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ	
1.1	Дисциплина «Объектно-ориентированное программирование и визуальное моделирование» предполагает обучение
1.2	будущих специалистов в области экономики, навыкам программирования на объектно-ориентированном языке
1.3	Python, а также изучение возможностей визуального моделирования для создания программных решений в сфере
1.4	экономики и бизнеса.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП	
Цикл (раздел) ООП:	Б1.В
2.1	**Требования к предварительной подготовке обучающегося:**
2.1.1	Информационные технологии в экономике
2.1.2	Алгоритмизация и программирование экономических задач
2.1.3	Прикладное программное обеспечение
2.2	**Дисциплины и практики, для которых освоение данной дисциплины (модуля) необходимо как предшествующее:**
2.2.1	Системы искусственного интеллекта
2.2.2	Нейронные сети
2.2.3	Преддипломная практика
2.2.4	Анализ данных

3. КОМПЕТЕНЦИИ ОБУЧАЮЩЕГОСЯ, ФОРМИРУЕМЫЕ В РЕЗУЛЬТАТЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)	
ПК-2: Способен обоснованно использовать современные инструменты и методы сбора, подготовки и анализа данных.	
Знать:	
Уровень 1	Основные принципы объектно-ориентированного программирования (инкапсуляция, наследование, полиморфизм).
Уровень 2	Методы разработки объектно-ориентированных моделей для решения экономических задач.
Уровень 3	Принципы работы с библиотеками и фреймворками Python для построения моделей и анализа данных.
Уметь:	
Уровень 1	Проектировать и реализовывать программы, использующие объектно-ориентированный подход.
Уровень 2	Использовать Python для разработки программ.
Уровень 3	Применять объектно-ориентированный подход для анализа и моделирования экономических процессов.
Владеть:	
Уровень 1	Построения и использования объектных структур данных (классы, объекты, наследование).
Уровень 2	Написания структурированного, читаемого и документированного кода.
Уровень 3	Применения ООП для разработки программных систем, автоматизирующих задачи анализа данных и исследования операций.

В результате освоения дисциплины обучающийся должен

3.1	**Знать:**
3.1.1	Основные принципы объектно-ориентированного программирования (инкапсуляция, наследование, полиморфизм).
3.1.2	Синтаксис и семантику языка Python для разработки программных решений.
3.1.3	Методы разработки объектно-ориентированных моделей для решения экономических задач.
3.1.4	Принципы работы с библиотеками и фреймворками Python для построения моделей и анализа данных.
3.2	**Уметь:**
3.2.1	Проектировать и реализовывать программы, использующие объектно-ориентированный подход.
3.2.2	Использовать Python для разработки программ.
3.2.3	Применять объектно-ориентированный подход для анализа и моделирования экономических процессов.
3.2.4	Применения ООП для разработки программных систем, автоматизирующих задачи анализа данных и исследования операций.
3.3	**Владеть:**
3.3.1	Построения и использования объектных структур данных (классы, объекты, наследование).
3.3.2	Написания структурированного, читаемого и документированного кода.

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)								
Код занятия	**Наименование разделов и тем / вид занятия/**	**Семестр / Курс**	**Часов**	**Компетен-ции**	**Литература**	**Инте ракт.**	**Пр. подг.**	**Примечание**
	Раздел 1. Раздел 1. Введение в ООП.							
1.1	Основы программирования на Python: переменные, типы данных, операторы, ввод-вывод /Лек/	6	2		Л1.1Л2.2Л3.1 Э1			
1.2	Функции и модули в Python: создание, вызов и работа с библиотеками. /Лек/	6	2		Л1.4Л2.4Л3.1 Э1			
1.3	Основы объектно-ориентированного программирования: классы, объекты и методы /Лек/	6	2		Л1.5Л2.4Л3.1 Э1			
1.4	Принципы ООП: инкапсуляция, наследование, полиморфизм. /Лек/	6	2		Л1.4Л2.4Л3.1 Э1			
1.5	Решение задач с использованием базовых конструкций Python (условия, циклы). /Пр/	6	2		Л1.1Л2.2Л3.3 Э1 Э2 Э3		2	
1.6	Написание программ с использованием функций: передача параметров, возвращаемые значения. /Пр/	6	2		Л1.1 Л1.4Л2.2Л3.1 Э1 Э2 Э3		2	
1.7	Создание собственных модулей и подключение библиотек /Пр/	6	2		Л1.1Л2.2Л3.1 Э1 Э2 Э3		2	
1.8	Создание классов и объектов для моделирования простых задач. /Пр/	6	2		Л1.4Л2.4Л3.1 Э1 Э2 Э3	2	2	
1.9	Инкапсуляция: управление доступом к данным через методы /Пр/	6	2		Л1.1Л2.2 Л2.4Л3.1 Э1 Э2 Э3		2	
1.10	Реализация иерархии классов с использованием наследования /Пр/	6	2		Л1.1Л2.2 Л2.4Л3.1 Э2	2	2	
1.11	Полиморфизм: перегрузка методов и операторов. /Пр/	6	2		Л1.1Л2.4Л3.1 Э1 Э2 Э3		2	
1.12	Обработка исключений и работа с ошибками в Python. /Пр/	6	2		Л1.1Л2.2Л3.1 Э1 Э2 Э3		2	
1.13	Создание программ для обработки пользовательского ввода с использованием функций. /Ср/	6	5		Л1.1 Л1.4Л2.2Л3.1 Э1			

1.14	Проектирование и реализация классов для работы с текстовыми и числовыми данными. /Ср/	6	5		Л1.1Л2.4Л3.1 Э1 Э2 Э3			
1.15	Решение задач с использованием наследования. /Ср/	6	4		Л1.1 Л1.4Л2.2 Л2.4Л3.1 Э1 Э2 Э3			
1.16	Написание программ с обработкой исключений для решения реальных задач /Ср/	6	4		Л1.1 Л1.4Л2.2Л3.1 Э1 Э2 Э3			
	Раздел 2. Раздел 2. Визуальное моделирование. Работа с Numpy.							
2.1	Введение в библиотеку NumPy: массивы, их структура и основные операции /Лек/	6	2		Л1.1 Л1.5Л2.4Л3.3 Э1			
2.2	Применение NumPy для математических расчетов и линейной алгебры. /Лек/	6	2		Л1.5Л2.2Л3.3 Э1			
2.3	Создание и изменение массивов NumPy: одномерные и многомерные массивы /Пр/	6	2		Л1.1 Л1.5Л2.2Л3.3 Э1 Э2 Э3		2	
2.4	Индексация, срезы и итерация по массивам NumPy. /Пр/	6	2		Л1.5Л2.2Л3.3 Э1 Э2 Э3		2	
2.5	Применение встроенных функций NumPy для вычислений (сумма, среднее, стандартное отклонение). /Пр/	6	2		Л1.5Л2.2Л3.3 Э1 Э2 Э3		2	
2.6	Решение задач линейной алгебры (матрицы, детерминанты, системы уравнений) с использованием NumPy /Пр/	6	2		Л1.5Л2.2Л3.3 Э1 Э2 Э3		2	
2.7	Создание и работа с большими массивами NumPy для обработки данных /Ср/	6	4		Л1.2Л2.1Л3.3 Э1 Э2 Э3			
2.8	Реализация пользовательских функций для выполнения расчетов на основе массивов NumPy. /Ср/	6	4		Л1.2 Л1.5Л2.1Л3.3 Э1 Э2 Э3			
2.9	Изучение методов оптимизации операций с большими объемами данных в NumPy. /Ср/	6	4		Л1.2 Л1.5Л2.1 Л2.4Л3.1 Л3.3 Э1 Э2 Э3			
2.10	Решение прикладных задач экономического анализа с использованием NumPy /Ср/	6	5,9		Л1.5Л2.1Л3.3 Э1 Э2 Э3			

2.11	Решение комплексных задач на основе проектирования и анализа данных, включая разработку программных модулей и алгоритмов. /КрТО/	6	0,1		Л1.2 Л1.5Л2.1 Л2.2Л3.3 Э1 Э2 Э3			
	Раздел 3. Раздел 3. Проектирование. Использование Pandas.							
3.1	Введение в библиотеку Pandas: структуры данных Series и DataFrame /Лек/	6	2		Л1.3Л2.3Л3.3 Э1			
3.2	Методы анализа и преобразования данных в Pandas. /Лек/	6	2		Л1.5Л2.3Л3.1 Л3.3 Э1			
3.3	Создание, импорт и редактирование данных в DataFrame /Пр/	6	2		Л1.2 Л1.3 Л1.5Л2.3Л3.1 Л3.3 Э1 Э2	2	2	
3.4	Обработка данных: фильтрация, сортировка, заполнение пропусков /Пр/	6	2		Л1.2 Л1.5Л2.1 Л2.3Л3.1 Л3.3 Э1 Э2 Э3	2	2	
3.5	Анализ данных: группировка, агрегирование, вычисление статистических показателей /Пр/	6	2		Л1.5Л2.1 Л2.3Л3.1 Л3.3 Э1 Э2 Э3		2	
3.6	Построение графиков и визуализация данных с помощью Pandas и Matplotlib /Пр/	6	2		Л1.1 Л1.5Л2.1 Л2.3Л3.1 Л3.3 Э1 Э2 Э3		2	
3.7	Импорт данных из различных форматов (CSV, Excel) и их предварительная обработка. /Ср/	6	6		Л1.2 Л1.5Л2.1Л3.1 Л3.3 Э1 Э2 Э3			
3.8	Реализация собственных методов для анализа и обработки данных в Pandas. /Ср/	6	6		Л1.3Л2.1Л3.2 Л3.3 Э1 Э2 Э3			
3.9	Построение временных рядов и анализ их изменений с использованием Pandas. /Ср/	6	6		Л1.5Л2.1 Л2.3Л3.2 Э1 Э2 Э3			
3.10	Разработка мини-проекта по анализу экономических данных с использованием библиотеки Pandas. /Ср/	6	6		Л1.2 Л1.5Л2.1Л3.1 Л3.2 Л3.3 Э1 Э2 Э3			

5. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

5.1. Контрольные вопросы и задания

1. Контрольные вопросы

Раздел 1. Введение в объектно-ориентированное программирование

Что представляет собой язык программирования Python? Назовите его основные особенности.

Какие основные типы данных используются в Python?

Какие арифметические, логические и операторы сравнения поддерживаются в Python?

Как организуется ввод и вывод данных в Python?

Что такое условный оператор? Как организуются ветвления в программе?

Для чего используются циклы for и while?

Что такое функция? Как осуществляется ее объявление и вызов?

Какие способы передачи аргументов в функцию существуют?

Что представляет собой возвращаемое значение функции?

Что такое модуль Python и для чего он используется?

Как осуществляется создание и подключение пользовательских модулей?

Что такое класс и объект в объектно-ориентированном программировании?

Какие основные элементы включает описание класса?

Для чего предназначен метод __init__()?

В чем заключается принцип инкапсуляции?

Какие способы ограничения доступа к атрибутам и методам используются в Python?

Что такое наследование? Какие преимущества оно предоставляет при разработке программ?

Что представляет собой иерархия классов?

В чем заключается принцип полиморфизма?

Как реализуется переопределение методов в дочерних классах?

Для чего используется функция super()?

Что такое исключения в Python?

Для чего предназначены конструкции try, except, else и finally?

Какие преимущества предоставляет обработка исключений при разработке программных приложений?

Раздел 2. Визуальное моделирование. Работа с библиотекой NumPy

Для чего предназначена библиотека NumPy?

Что такое массив NumPy и чем он отличается от стандартного списка Python?

Какие основные типы массивов поддерживает NumPy?

Как создаются одномерные и многомерные массивы?

Какие способы изменения формы и размерности массивов существуют?

Что такое индексация и срезы массивов?

Как осуществляется обращение к элементам многомерного массива?

Какие встроенные функции NumPy используются для выполнения математических вычислений?

Как вычисляются сумма, среднее значение и стандартное отклонение элементов массива?

Какие операции линейной алгебры можно выполнять с помощью NumPy?

Как осуществляется умножение матриц?

Для чего используются функции вычисления определителя и решения систем линейных уравнений?

Что такое векторизация вычислений?

В чем заключаются преимущества использования NumPy при обработке больших объемов данных?

Какие методы оптимизации операций с массивами существуют?

Как NumPy применяется для решения прикладных задач экономического анализа?

Раздел 3. Проектирование. Использование библиотеки Pandas

Для чего предназначена библиотека Pandas?

Что представляют собой структуры данных Series и DataFrame?

В чем заключаются основные отличия Series и DataFrame?

Какими способами можно создавать объекты DataFrame?

Как осуществляется импорт данных из файлов CSV и Excel?

Какие методы используются для просмотра структуры и основных характеристик набора данных?

Как выполняется выборка строк и столбцов DataFrame?

Какие способы фильтрации и сортировки данных поддерживает Pandas?

Что такое пропущенные значения и какие методы используются для их обнаружения?

Какие способы обработки и заполнения пропущенных значений существуют?

Для чего используются методы группировки данных?

Как осуществляется агрегирование данных с помощью Pandas?

Какие статистические показатели можно вычислять средствами Pandas?

Как выполняется создание новых столбцов на основе существующих данных?

Для чего используются методы преобразования данных?

Какие возможности Pandas применяются для анализа временных рядов?

Какие инструменты используются для построения графиков и визуализации данных?

Как осуществляется визуализация данных с использованием Pandas и Matplotlib?

Какие этапы включает предварительная обработка экономических данных?

Какие задачи экономического анализа можно решать с использованием библиотеки Pandas?

Какие этапы включает разработка мини-проекта по анализу экономических данных?

2. Задания для проведения текущего контроля

Текущий контроль знаний и умений обучающихся осуществляется в процессе выполнения практических и самостоятельных работ, тестирования, решения индивидуальных заданий и анализа результатов программирования.

Задания по разделу 1. Введение в объектно-ориентированное программирование

Разработать программу на Python для вычисления показателей экономической деятельности предприятия с использованием условных операторов и циклов.

Создать функцию для расчета выручки, затрат и прибыли предприятия. Предусмотреть передачу параметров и возврат результатов вычислений.

Создать пользовательский модуль, содержащий функции для выполнения экономических расчетов, и подключить его к основной программе.

Разработать класс «Товар», содержащий атрибуты наименования, цены и количества товара. Реализовать методы вычисления общей стоимости товара.

Создать класс «Сотрудник» с атрибутами, характеризующими работника предприятия, и методами расчета заработной платы.

Реализовать принцип инкапсуляции при разработке класса «Банковский счет». Предусмотреть методы пополнения счета, снятия денежных средств и получения текущего баланса.

Разработать иерархию классов «Сотрудник», «Менеджер», «Бухгалтер», используя механизм наследования.

Реализовать полиморфизм на примере классов, описывающих различные виды экономических операций.

Разработать программу обработки пользовательского ввода с использованием функций и обработкой возможных исключений.

Создать объектно-ориентированную модель учета товаров на предприятии с возможностью добавления товаров, изменения их характеристик и расчета общей стоимости запасов.

Задания по разделу 2. Визуальное моделирование. Работа с NumPy

Создать одномерный массив NumPy, содержащий значения ежемесячной выручки предприятия.

Сформировать двумерный массив, содержащий показатели доходов и расходов предприятия за несколько месяцев.

Выполнить индексацию и срезы массива для получения отдельных значений и групп данных.

Рассчитать сумму, среднее значение, минимальное и максимальное значения элементов массива.

Вычислить стандартное отклонение показателей экономической деятельности предприятия.

Выполнить арифметические операции над массивами одинаковой размерности.

Изменить форму и размерность массива с использованием соответствующих методов NumPy.

Сформировать матрицу экономических показателей и выполнить операции сложения и умножения матриц.

Вычислить определитель заданной матрицы и решить систему линейных уравнений с использованием NumPy.

Разработать программу для анализа динамики экономических показателей на основе массивов NumPy.

Задания по разделу 3. Проектирование. Использование Pandas

Создать DataFrame, содержащий сведения о товарах, их стоимости и количестве.

Загрузить набор экономических данных из CSV-файла и определить его основные характеристики.

Импортировать данные из Excel-файла и выполнить их предварительную обработку.

Выполнить фильтрацию и сортировку данных по нескольким условиям.

Обнаружить пропущенные значения в наборе данных и выполнить их обработку.

Создать новые столбцы DataFrame на основе существующих показателей.

Выполнить группировку экономических данных по заданным категориям и рассчитать агрегированные показатели.

Рассчитать основные статистические характеристики набора данных.

Построить графики изменения экономических показателей с использованием Pandas и Matplotlib.

Импортировать данные о продажах предприятия, выполнить их обработку и провести сравнительный анализ показателей за различные периоды.

Сформировать временной ряд на основе экономических показателей и проанализировать динамику их изменения.

Разработать мини-проект по анализу экономических данных с использованием библиотеки Pandas.

3. Задания для проведения промежуточной аттестации

Промежуточная аттестация проводится в форме зачета с оценкой и предусматривает проверку теоретических знаний и практических навыков применения языка Python, принципов объектно-ориентированного программирования, библиотеки NumPy и инструментов анализа данных Pandas.

Теоретические задания:

Раскрыть основные принципы объектно-ориентированного программирования и привести примеры их практического применения.

Объяснить механизм создания классов и объектов в Python.

Охарактеризовать принципы инкапсуляции, наследования и полиморфизма.

Рассмотреть особенности использования функций и пользовательских модулей Python.

Объяснить назначение и особенности обработки исключений.

Охарактеризовать структуру массивов NumPy и основные операции над ними.

Раскрыть возможности NumPy для выполнения математических расчетов и операций линейной алгебры.

Объяснить назначение структур Series и DataFrame библиотеки Pandas.

Охарактеризовать основные методы импорта, обработки, фильтрации и анализа данных.

Рассмотреть возможности визуализации экономических данных с использованием Pandas и Matplotlib.

Практические задания:

Разработать объектно-ориентированную программу для учета доходов и расходов предприятия с использованием классов, объектов и методов.

Создать систему учета товаров с применением принципов инкапсуляции и наследования.

Разработать программу для расчета экономических показателей с использованием функций, модулей и обработки исключений.

Создать массив NumPy на основе заданных экономических показателей и выполнить их статистическую обработку.

Решить систему линейных уравнений, описывающую экономическую задачу, с использованием библиотеки NumPy.

Загрузить экономический набор данных в DataFrame, выполнить проверку качества данных, фильтрацию, сортировку и обработку пропущенных значений.

Провести группировку и агрегирование данных о продажах предприятия, рассчитать основные статистические показатели.

Построить графическое представление динамики экономических показателей и сформулировать выводы по результатам визуального анализа.

Разработать программный модуль для обработки и анализа временных рядов экономических данных.

Выполнить комплексное задание по разработке программы анализа экономических данных с использованием Python, NumPy и Pandas.

5.2. Темы курсовых работ (проектов)

Дисциплина не предусматривает написание курсовой работы

5.3. Фонд оценочных средств

Фонд оценочных средств по дисциплине «Объектно-ориентированное программирование и визуальное моделирование» предназначен для оценки уровня сформированности компетенции ПК-2: способности обоснованно использовать современные инструменты и методы сбора, подготовки и анализа данных.

Фонд оценочных средств включает совокупность контрольных материалов, позволяющих оценить теоретические знания, практические умения и навыки обучающихся в области объектно-ориентированного программирования на языке Python, математических вычислений с использованием библиотеки NumPy, а также обработки, анализа и визуализации данных с использованием библиотеки Pandas.

Оценивание результатов освоения дисциплины осуществляется посредством текущего контроля и промежуточной аттестации.

1. Средства текущего контроля

Для проверки текущего уровня освоения дисциплины используются следующие оценочные средства:

Устный опрос – проверка знаний основных понятий и принципов программирования на Python, объектно-ориентированного подхода, методов работы с массивами NumPy и инструментов анализа данных Pandas.

Тестирование – проверка теоретических знаний по основным разделам дисциплины, включая синтаксис Python, функции, классы, объекты, принципы ООП, операции с массивами и методы обработки данных.

Практические задания – проверка умения разрабатывать программы на языке Python, создавать классы и объекты, применять механизмы наследования, инкапсуляции и полиморфизма, выполнять математические расчеты и обработку данных.

Лабораторные работы – оценка способности самостоятельно решать прикладные задачи с использованием Python, NumPy и Pandas, проводить обработку и анализ экономических данных.

Самостоятельные работы – проверка способности самостоятельно разрабатывать программные решения, выполнять анализ данных и применять изученные инструменты для решения экономических задач.

Контрольная работа – комплексная проверка теоретических знаний и практических навыков по основным разделам дисциплины.

2. Средства промежуточной аттестации

Промежуточная аттестация проводится в форме зачета с оценкой и предусматривает оценку теоретической и практической подготовки обучающихся.

В качестве оценочных средств используются:

теоретические вопросы по основным разделам дисциплины;

практические задания по программированию на Python;

задания на разработку объектно-ориентированных моделей;

задания по выполнению математических и статистических расчетов с использованием NumPy;

задания по импорту, обработке, анализу и визуализации данных с использованием Pandas;

комплексные задания, направленные на решение прикладных экономических задач.

3. Оценочные материалы по разделам дисциплины

Раздел 1. Введение в объектно-ориентированное программирование

Оцениваются знания синтаксиса и основных конструкций Python, умение разрабатывать функции и модули, создавать классы и объекты, применять принципы инкапсуляции, наследования и полиморфизма, а также осуществлять обработку исключений.

Формы контроля: устный опрос, тестирование, практические и самостоятельные задания, лабораторные работы.

Раздел 2. Визуальное моделирование. Работа с NumPy

Оцениваются знания структуры массивов NumPy, умение создавать и преобразовывать одномерные и многомерные массивы,

выполнять индексацию, срезы, математические вычисления и операции линейной алгебры, применять библиотеку для решения прикладных экономических задач.

Формы контроля: тестирование, практические задания, лабораторные и самостоятельные работы.

Раздел 3. Проектирование. Использование Pandas

Оцениваются знания структур Series и DataFrame, умение импортировать данные из различных форматов, выполнять фильтрацию, сортировку, обработку пропущенных значений, группировку, агрегирование, статистический анализ и визуализацию данных.

Формы контроля: практические и лабораторные работы, самостоятельные задания, контрольная работа, выполнение мини-проекта.

4. Критерии оценивания результатов освоения дисциплины

При оценивании результатов освоения дисциплины учитываются:

полнота и системность теоретических знаний;

правильность применения синтаксиса и конструкций языка Python;

владение принципами объектно-ориентированного программирования;

умение самостоятельно разрабатывать и отлаживать программный код;

правильность применения библиотек NumPy и Pandas;

способность выполнять обработку, анализ и визуализацию данных;

умение применять изученные методы для решения прикладных экономических задач;

самостоятельность и обоснованность принимаемых решений.

5. Шкала оценивания промежуточной аттестации

«Отлично» – обучающийся демонстрирует полное и системное знание теоретического материала, уверенно применяет принципы объектно-ориентированного программирования, самостоятельно разрабатывает программы на Python, эффективно использует библиотеки NumPy и Pandas для обработки и анализа данных, правильно интерпретирует полученные результаты.

«Хорошо» – обучающийся демонстрирует достаточный уровень теоретических знаний и практических навыков, умеет разрабатывать программы и применять основные инструменты NumPy и Pandas, допускает отдельные несущественные ошибки, которые способен самостоятельно исправить.

«Удовлетворительно» – обучающийся демонстрирует базовые знания основных разделов дисциплины, выполняет типовые практические задания с отдельными затруднениями, нуждается в дополнительной помощи при разработке программ и обработке данных.

«Неудовлетворительно» – обучающийся не демонстрирует необходимого уровня теоретических знаний, не владеет основными конструкциями языка Python, не умеет применять принципы объектно-ориентированного программирования и выполнять базовые операции обработки и анализа данных.

5.4. Перечень видов оценочных средств

В процессе освоения дисциплины «Объектно-ориентированное программирование и визуальное моделирование» используются следующие виды оценочных средств:

Устный опрос – проверка знаний основных понятий и принципов программирования на языке Python, объектно-ориентированного программирования, работы с библиотеками NumPy и Pandas.

Тестирование – проверка усвоения теоретического материала по основным разделам дисциплины, включая основы Python, принципы ООП, математические вычисления и методы анализа данных.

Практические задания – оценка навыков разработки программ на Python, создания функций, классов и объектов, применения принципов инкапсуляции, наследования и полиморфизма.

Лабораторные работы – оценка способности применять изученные методы и инструменты Python, NumPy и Pandas для

решения прикладных задач, выполнения математических расчетов, обработки и анализа экономических данных.

Самостоятельная работа – оценка способности самостоятельно изучать программные инструменты, разрабатывать программные решения, выполнять обработку и анализ данных, решать индивидуальные задачи.

Контрольная работа – проверка уровня усвоения теоретического материала и сформированности практических навыков по изученным разделам дисциплины.

Зачет с оценкой – итоговая оценка теоретических знаний и практических умений обучающихся по результатам освоения дисциплины.

6. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

6.1. Рекомендуемая литература

6.1.1. Основная литература

	Авторы, составители	Заглавие	Издательство, год
Л1.1	Сузи Р. А.	Язык программирования Python: учебное пособие	Москва: Интернет-Университет Информационных Технологий (ИНТУИТ) 2016
Л1.2	Charbit M.	Digital Signal Processing with Python Programming: учебное пособие	S. l: s. n. 2017
Л1.3	Амоа К. А., Рындин Н. А., Скворцов Ю. С.	Разработка программных пакетов на языке Python: Учебное пособие	Воронеж: Воронежский государственный технический университет, ЭБС АСВ 2020
Л1.4	Шелудько В. М.	Основы программирования на языке высокого уровня Python: Учебное пособие	Ростов-на-Дону, Таганрог: Издательство Южного федерального университета 2017
Л1.5	Маккинли Уэс, Слинкина А.	Python и анализ данных	Саратов: Профобразование 2019

6.1.2. Дополнительная литература

	Авторы, составители	Заглавие	Издательство, год
Л2.1	Шолле Ф.	Глубокое обучение на Python: учебное пособие	СПб.: Питер 2018
Л2.2	Буйначев С. К., Боклаг Н. Ю., Песин Ю. В.	Основы программирования на языке Python: Учебное пособие	Екатеринбург: Уральский федеральный университет, ЭБС АСВ 2014
Л2.3	Дроботун Н. В., Рудков Е. О., Баев Н. А.	Алгоритмизация и программирование. Язык Python: Учебное пособие	Санкт-Петербург: Санкт-Петербургский государственный университет промышленных технологий и дизайна 2020
Л2.4	Букунов С. В., Букунова О. В.	Объектно ориентированное программирование на языке Python: Учебное пособие	Санкт-Петербург: Санкт-Петербургский государственный архитектурно-строительный университет, ЭБС АСВ 2020

6.1.3. Методические разработки

	Авторы, составители	Заглавие	Издательство, год
Л3.1	Рик Гаско	Простой Python просто с нуля	Москва: СОЛОН-Пресс 2019
Л3.2	Шелудько В. М.	Язык программирования высокого уровня Python. Функции, структуры данных, дополнительные модули: Учебное пособие	Ростов-на-Дону, Таганрог: Издательство Южного федерального университета 2017
Л3.3	Глебов В. И., Криволапов С. Я.	Практикум по математической статистике. Проверка гипотез с использованием Excel, MatCalc, R и Python: Учебное пособие	Москва: Прометей 2019

6.2. Перечень ресурсов информационно-телекоммуникационной сети "Интернет"

Э1	Сайт Python предоставляет официальную документацию, учебники и руководства для изучения Python.	https://www.python.org/
Э2	Специализация на Coursera, разработанная для тех, кто хочет изучить Python и его применение в анализе данных.	https://www.coursera.org/specializations/python
Э3	GeeksforGeeks предлагает обширные ресурсы, включая статьи, учебники и задачи для практики на Python.	https://www.geeksforgeeks.org/

6.3. Перечень информационных и образовательных технологий	
6.3.1 Компетентностно-ориентированные образовательные технологии	
6.3.1.1	1. Традиционные: Лекции, лабораторные работы.
6.3.1.2	2. Инновационные образовательные технологии - дискуссионный разбор выполненных заданий на практических занятиях.
6.3.1.3	3. Информационные образовательные технологии (использование googleColab)
6.3.2 Перечень информационных справочных систем и программного обеспечения	
6.3.2.1	1. PyCharm - популярная интегрированная среда разработки (IDE) для Python
6.3.2.2	2. IDLE (интегрированная среда разработки и обучения) : IDLE — это стандартная среда разработки, поставляемая с Python.
6.3.2.3	3. Visual Studio Code (VSCode) : VSCode — текстовый редактор с поддержкой Python через расширение.
6.3.2.4	4. Google Colab — сервис, созданный Google, который предоставляет возможность работать с кодом на языке Python через Jupyter Notebook.
6.3.2.5	5. https://www.python.org/doc/ - Официальная документация Python.
6.3.2.6	6. https://www.codecademy.com/learn/learn-python-3 - интерактивный онлайн-курс Python.

7. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)	
7.1	Аудитории с мультимедийным оборудованием (проектор, экран, аудиосистема) для проведения лекций и семинаров.
7.2	Компьютерные классы, оснащённые современными персональными компьютерами или ноутбуками с установленным необходимым программным обеспечением.
7.3	Интернет-доступ в учебных помещениях для работы с облачными сервисами и онлайн-ресурсами.

8. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)

Дисциплина «Объектно-ориентированное программирование и визуальное моделирование» направлена на формирование у обучающихся теоретических знаний и практических навыков разработки программных решений на языке Python, применения принципов объектно-ориентированного программирования, выполнения математических расчетов, обработки, анализа и визуализации данных.

Освоение дисциплины предполагает сочетание лекционных и практических занятий, самостоятельной работы обучающихся, выполнения лабораторных заданий и подготовки к промежуточной аттестации.

1. Общие рекомендации по изучению дисциплины

При изучении дисциплины обучающимся рекомендуется:

последовательно изучать теоретический материал в соответствии с тематическим планом дисциплины;

закреплять теоретические знания путем выполнения практических и лабораторных работ;

самостоятельно изучать дополнительные материалы по языку программирования Python и используемым библиотекам;

регулярно выполнять задания, связанные с разработкой и отладкой программ;

использовать официальную документацию Python, NumPy и Pandas;

анализировать результаты выполнения программ и формулировать выводы;

своевременно выполнять задания для самостоятельной работы и готовиться к текущему контролю.

Особое внимание следует уделять формированию навыков самостоятельного программирования, пониманию логики работы программ и применению изученных инструментов для решения прикладных экономических задач.

2. Методические рекомендации по изучению лекционного материала

Лекционные занятия предназначены для формирования систематизированных знаний по основным разделам дисциплины.

При подготовке к лекциям обучающимся рекомендуется предварительно ознакомиться с соответствующей темой, повторить ранее изученные конструкции языка Python и основные понятия программирования.

Во время лекционных занятий необходимо вести конспект, фиксировать определения, основные понятия, синтаксические

конструкции языка Python, примеры программного кода и алгоритмы решения задач.

После изучения каждой темы рекомендуется:

повторить основные теоретические положения;

разобраться в приведенных примерах программного кода;

самостоятельно воспроизвести рассмотренные программные решения;

ответить на контрольные вопросы по изученной теме;

определить, каким образом изученные методы могут применяться для решения экономических задач.

3. Методические рекомендации по выполнению практических и лабораторных работ

Практические и лабораторные занятия направлены на закрепление теоретических знаний и приобретение практических навыков программирования и анализа данных.

При выполнении практических и лабораторных работ рекомендуется придерживаться следующей последовательности:

Ознакомиться с темой, целью и содержанием практической работы.

Изучить теоретический материал, необходимый для выполнения задания.

Определить исходные данные и сформулировать последовательность решения поставленной задачи.

Разработать программный код на языке Python.

Выполнить программу и проверить правильность полученных результатов.

При обнаружении ошибок провести отладку программного кода.

Проанализировать результаты выполнения задания.

Подготовить отчет о выполненной работе с необходимыми пояснениями и выводами.

Программный код должен быть структурированным, понятным и содержать необходимые комментарии. При разработке программ рекомендуется использовать осмысленные имена переменных, функций, классов и методов.

При выполнении работ по разделу 1 «Введение в объектно-ориентированное программирование» обучающимся необходимо освоить основные конструкции языка Python, научиться создавать функции и пользовательские модули, разрабатывать классы и объекты, применять принципы инкапсуляции, наследования и полиморфизма. Особое внимание следует уделять практическому применению объектно-ориентированного подхода при моделировании экономических объектов и процессов.

При выполнении работ по разделу 2 «Визуальное моделирование. Работа с NumPy» необходимо освоить способы создания и преобразования массивов, индексацию и срезы, выполнение математических и статистических вычислений, а также решение задач линейной алгебры. Рекомендуется самостоятельно экспериментировать с различными массивами и проверять результаты выполнения операций.

При выполнении работ по разделу 3 «Проектирование. Использование Pandas» обучающимся необходимо освоить работу со структурами Series и DataFrame, импорт данных из файлов CSV и Excel, фильтрацию, сортировку, обработку пропущенных значений, группировку и агрегирование данных. Особое внимание следует уделять анализу экономических показателей и графическому представлению полученных результатов с использованием Pandas и Matplotlib.

4. Методические рекомендации по организации самостоятельной работы

Самостоятельная работа является важной составляющей освоения дисциплины и направлена на углубление теоретических знаний, закрепление практических навыков и развитие способности самостоятельно решать профессионально ориентированные задачи.

В процессе самостоятельной работы обучающимся рекомендуется:

повторять лекционный материал и изучать рекомендуемую литературу;

самостоятельно разрабатывать программы на Python по изученным темам;

выполнять дополнительные упражнения по созданию функций, классов и объектов;

изучать возможности библиотек NumPy и Pandas;

самостоятельно подбирать наборы данных для выполнения аналитических заданий;

выполнять обработку и статистический анализ экономических данных;

изучать способы визуального представления результатов анализа;

готовить материалы для выполнения комплексных заданий и мини-проектов.

При выполнении самостоятельных заданий рекомендуется не ограничиваться воспроизведением рассмотренных на занятиях примеров, а изменять исходные данные, условия задач и структуру программ, проверяя