МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
МИНИСТЕРСТВО НАУКИ, ВЫСШЕГО ОБРАЗОВАНИЯ И ИННОВАЦИЙ КЫРГЫЗСКОЙ РЕСПУБЛИКИ
Межгосударственная образовательная организация высшего образования
«Кыргызско-Российский Славянский университет имени первого Президента Российской Федерации Б.Н. Ельцина»

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

по дисциплине
«Объектно-ориентированное программирование и визуальное моделирование»

Уровень высшего образования: БАКАЛАВРИАТ
Направление подготовки: 38.03.01 - РФ, 580100 - КР Экономика
Профиль «Математические методы в экономике»
Квалификация: бакалавр
Форма обучения: очная

Бишкек 2026

Фонд оценочных средств предназначен для контроля знаний обучающихся по направлению подготовки 38.03.01 - РФ, 580100 - КР Экономика профиль «Математические методы в экономике».

Фонд оценочных средств рассмотрен и утвержден на заседании Международной бизнес-школы протокол № _1_ от 03.09.2026 г.

Директор Международной бизнес-школы Кочербаева А.А.

Руководитель образовательной программы
Математические методы в экономике Мокроусов Н.В.

Исполнители:

Доцент Мусакулова Ж.А.

1. ПЕРЕЧЕНЬ КОМПЕТЕНЦИЙ С УКАЗАНИЕМ ЭТАПОВ ИХ ФОРМИРОВАНИЯ В ПРОЦЕССЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Освоение дисциплины «Объектно-ориентированное программирование и визуальное моделирование» на языке Python направлено на формирование у обучающихся следующей ключевой профессиональной компетенции в соответствии с ФГОС ВО 38.03.01 «Экономика»:

Формируемые компетенции	Планируемые результаты обучения по дисциплине (индикаторы достижения компетенции)	Виды оценочных средств / Шифр
ПК-3: Способен использовать специализированные программные средства для повышения офисной продуктивности и коммуникаций, проведения оценки и анализа, визуализации результатов анализа и предлагаемых решений.	Знать: − основные принципы объектно-ориентированного программирования (инкапсуляция, наследование, полиморфизм); − синтаксис и семантику языка Python для разработки программных решений; − методы разработки объектно-ориентированных моделей для решения экономических задач; − принципы работы с библиотеками NumPy, Pandas и Matplotlib для построения моделей и визуализации данных.	Блок A, D
	Уметь: − проектировать и реализовывать программы, использующие объектно-ориентированный подход на Python; − использовать язык Python для разработки эффективных вычислительных модулей; − применять объектно-ориентированный подход для анализа и моделирования экономических процессов.	Блок B, D
	Владеть: − навыками построения и использования объектных структур данных (классы, объекты, наследование, свойства); − навыками написания структурированного, читаемого и документированного кода на Python; − навыками применения ООП и библиотек (NumPy, Pandas, Matplotlib) для разработки программных систем, автоматизирующих задачи анализа данных.	Блок C, D

2. ТЕХНОЛОГИЧЕСКАЯ КАРТА ДИСЦИПЛИНЫ

Дисциплина: «Объектно-ориентированное программирование и визуальное моделирование» (Python)

Курс / семестр: 3 / 6 семестр | Общая трудоемкость: 3 ЗЕТ (96 часов) | Форма контроля: Зачет с оценкой

Название модулей дисциплины согласно РПД	Контроль	Форма контроля	Мин. баллы	Макс. баллы	График контроля
Модуль 1. Раздел 1. Введение в ООП на Python (переменные, функции, классы, инкапсуляция, наследование, исключения)	Текущий контроль	Домашние задания № 1-4	7	12	7 нед.
	Рубежный контроль	Тестирование по Модулю 1	5	8	8 нед.
Модуль 2. Раздел 2. Визуальное моделирование. Работа с NumPy (массивы, векторизованные вычисления, линейная алгебра)	Текущий контроль	Лабораторная работа № 1	4	8	12 нед.
	Рубежный контроль	Рубежный тест / КР	4	6	13 нед.
Модуль 3. Раздел 3. Проектирование. Использование Pandas и Matplotlib (Series, DataFrame, анализ данных, построение графиков)	Текущий контроль	Лабораторная работа № 2	4	8	16 нед.
	Рубежный контроль	Контрольная работа / Проект	6	10	17 нед.
ВСЕГО за семестр (текущий и рубежный контроль)	Модульный	Накопительный рейтинг	40	70	1–17
Промежуточный контроль (Зачет с оценкой)	Промежуточный	Итоговый тест	20	30	Сессия
СЕМЕСТРОВЫЙ РЕЙТИНГ ПО ДИСЦИПЛИНЕ	Итоговый	Суммарный балльно-рейтинговый показатель	60	100	—

ВСЕГО за семестр (Текущий и рубежный контроль): зачетный минимум — 40 баллов, зачетный максимум — 70 баллов.

Промежуточный контроль (Зачет с оценкой): Итоговый тест (30 вопросов) — минимум 20 баллов, максимум 30 баллов.

Семестровый рейтинг по дисциплине: минимум — 60 баллов, максимум — 100 баллов.

3. ТИПОВЫЕ КОНТРОЛЬНЫЕ ЗАДАНИЯ И ИНЫЕ МАТЕРИАЛЫ, НЕОБХОДИМЫЕ ДЛЯ ОЦЕНКИ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ (ОЦЕНОЧНЫЕ СРЕДСТВА)

Блок А (ЗНАТЬ). Перечень контрольных вопросов по дисциплине (6 семестр)

1. Структура и особенности языка Python. Базовые типы данных: int, float, str, bool. Изменяемые и неизменяемые типы.
2. Структуры данных Python: списки (list), кортежи (tuple), словари (dict), множества (set). Основные методы и операции.
3. Управляющие конструкции в Python: условные операторы (if...elif...else), циклы (for, while), операторы break, continue, pass.
4. Функции в Python: синтаксис определения (def), позиционные и именованные аргументы, параметры по умолчанию.
5. Использование *args и **kwargs для передачи переменного числа аргументов в функции Python.
6. Анонимные функции (lambda) и встроенные функции высшего порядка (map, filter, sorted).
7. Области видимости переменных в Python: правила LEGB (Local, Enclosing, Global, Built-in). Ключевые слова global и nonlocal.
8. Модульная структура программы в Python. Создание собственных модулей, импорт с использованием import, from ... import.
9. Основные концепции объектно-ориентированного программирования (ООП): парадигма, преимущества, объектный подход к проектированию.
10. Классы и объекты в Python. Объявление класса (class), конструктор __init__, параметр self.
11. Атрибуты класса и атрибуты экземпляра: различие, область применения, особенности обращения через self и имя класса.
12. Инкапсуляция в Python: публичные, защищенные (_protected) и приватные (__private) атрибуты. Механизм искажения имен (name mangling).
13. Использование декораторов @property, @setter и @deleter для управляемого доступа к атрибутам класса.
14. Наследование в Python: базовые и дочерние классы, одиночное и множественное наследование. Функция super().
15. Полиморфизм в Python. Динамическая типизация (утиная типизация / duck typing) и переопределение методов.
16. Магические (dunder) методы в Python: __str__, __repr__, __eq__, __len__, __getitem__, __call__. Их назначение.
17. Перегрузка арифметических операторов в Python через методы __add__, __sub__, __mul__, __truediv__.
18. Обработка исключений в Python: конструкция try ... except ... else ... finally. Типы встроенных ошибок (ValueError, TypeError, ZeroDivisionError).
19. Создание и возбуждение пользовательских классов исключений (raise Exception).

20. Чтение и запись файлов в Python: контекстный менеджер with open() as f. Форматы TXT, CSV, JSON.

21. Библиотека NumPy: назначение, достоинства векторизованных вычислений по сравнению с базовыми списками Python.

22. Объект ndarray в NumPy: основные атрибуты (shape, ndim, dtype, size, itemsize).

23. Способы создания массивов NumPy: np.array(), np.zeros(), np.ones(), np.arange(), np.linspace(), np.eye().

24. Индексация и срезы (slicing) в одномерных и многомерных массивах NumPy. Булевая индексация.

25. Изменение формы и объединение массивов NumPy: reshape(), flatten(), np.concatenate(), np.vstack(), np.hstack().

26. Поэлементные операции и векторизованные математические функции в NumPy (np.sin, np.exp, np.log, np.sqrt).

27. Линейная алгебра в NumPy (модуль numpy.linalg): вычисление определителя (det), обратной матрицы (inv), решение СЛАУ (solve).

28. Статистические функции в NumPy: вычисление суммы (sum), среднего (mean), медианы (median), дисперсии (var) и стандартного отклонения (std).

29. Библиотека Pandas: основные структуры данных Series и DataFrame. Конструкторы и особенности индексирования.

30. Чтение и сохранение данных в Pandas: pd.read_csv(), pd.read_excel(), to_csv(), to_excel(). Параметры функций.

31. Доступ к данным в DataFrame: атрибуты .loc[] (по меткам) и .iloc[] (по позициям). Фильтрация данных по условиям.

32. Обработка пропущенных значений в Pandas: выявление (isna), удаление (dropna), заполнение (fillna, interpolate).

33. Преобразование и очистка данных в Pandas: переименование колонок, изменение типов (astype), удаление дубликатов (drop_duplicates).

34. Группировка и агрегирование данных в Pandas: метод groupby(), агрегирующие функции sum(), mean(), count(), agg().

35. Объединение и слияние наборов данных в Pandas: функции pd.concat() и pd.merge() (аналог JOIN в SQL).

36. Создание сводных таблиц (pivot_table) в Pandas для комплексного анализа экономических показателей.

37. Работа с временными рядами в Pandas: тип DatetimeIndex, функция pd.to_datetime(), ресемплирование (resample).

38. Визуализация данных с помощью библиотеки Matplotlib: построение графиков функцией plt.plot(), гистограмм plt.hist(), диаграмм рассеяния plt.scatter().

39. Оформление графиков в Matplotlib: заголовок (title), подписи осей (xlabel, ylabel), сетка (grid), легенда (legend), настройка цветов и типов линий.

40. Интеграция Pandas и Matplotlib: вызов метода df.plot() для быстрой визуализации экономических таблиц и построения мультиграфиков.

Блок В (УМЕТЬ). Типовые практические задания и домашние работы на Python

Домашнее задание №1-2. «Базовый синтаксис Python, структуры данных и обработка ввода»

Задание: Написать консольную программу на Python для анализа бюджета домохозяйства. Программа должна принимать словарь с категориями расходов (название и сумма), рассчитывать общие расходы, долю каждой категории в процентах, находить категорию с максимальными затратами и выводить отформатированный отчет.
Требования: Использовать циклы, условия, форматирование строк f-strings, обработку исключений при вводе чисел.

Домашнее задание №3-4. «Функциональное программирование и модульная организация»

Задание: Разработать модуль на Python financial_math.py, содержащий функции для вычисления параметров кредитования: аннуитетного платежа, переплаты по процентам и построения графика погашения. В основном скрипте импортировать модуль и выполнить расчет для трех вариантов процентных ставок.
Требования: Оформить docstrings для функций, применить параметры по умолчанию и передачу аргументов через *args.

Домашнее задание №5-6. «Проектирование базовых классов и объектов»

Задание: Создать класс Product (Товар), содержащий атрибуты: наименование, закупочная цена, торговая наценка и количество на складе. Реализовать методы расчета розничной цены, общей стоимости партии и метод продажи товара с уменьшением остатка.
Требования: Конструктор __init__, корректное использование self, вывод информации о товаре через метод get_info().

Домашнее задание №7-8. «Инкапсуляция и управление доступом к данным через свойства»

Задание: Создать класс BankAccount (Банковский счет) с приватными атрибутами __account_number, __balance и __holder_name. Реализовать методы пополнения (deposit), снятия (withdraw) с проверкой достаточности средств и декораторы @property и @balance.setter для защиты от установки отрицательного баланса.
Требования: Запрет прямого изменения баланса извне, генерирование исключений ValueError при некорректных операциях.

Домашнее задание №9-10. «Иерархия классов, наследование и полиморфизм»

Задание: Разработать иерархию классов для учета сотрудников фирмы. Базовый класс Employee (имя, оклад) с методом calculate_salary(). Дочерние классы: Manager (получает процент от продаж) и Developer (получает премию за каждый проект). Создать список объектов разных классов и полиморфно выдать ведомость выплат.
Требования: Переопределение метода calculate_salary(), использование super().__init__().

Домашнее задание №11-12. «Векторизованные вычисления и матрицы в NumPy»

Задание: С помощью библиотеки NumPy смоделировать матрицу прямых затрат A размера 4x4 для межотраслевого баланса Леонтьева. Задать вектор конечного спроса Y. Вычислить матрицу полных затрат B = (I - A)^(-1) и определить необходимый валовый выпуск X = B * Y.
Требования: Проверка условия продуктивности матрицы, использование np.eye(), np.linalg.inv(), np.dot().

Домашнее задание №13-14. «Анализ и визуализация данных в Pandas и Matplotlib»

Задание: Загрузить набор данных по продажам сети магазинов (колонки: Дата, Магазин, Товар, Продажи, Прибыль). Провести очистку пропусков, сгруппировать данные по магазинам, найти суммарную прибыль и построить столбиковую диаграмму (bar chart) в Matplotlib с подписями осей и легендой.
Требования: Использование pd.read_csv(), groupby(), agg(), plt.savefig().

Блок С (ВЛАДЕТЬ). Комплексные лабораторные работы и проекты

Лабораторная работа №1-5. «Объектно-ориентированное моделирование экономической системы фирмы на Python»

Цель: Освоить полное проектирование объектной структуры приложения для моделирования финансово-хозяйственной деятельности предприятия.
Постановка задачи:
Разработать систему классов: Company, Department, Project, Expense. Класс Company агрегирует список отделов и проектов. Реализовать методы расчета совокупных доходов, операционных расходов, чистой прибыли, а также налога на прибыль (20%). Предусмотреть сохранение итоговой финансовой отчетности фирмы в текстовый файл и загрузку из файла.
Требования к коду: Полное соблюдение принципов ООП, документация типов typing/docstrings, обработка ошибок ввода-вывода, написание тестового скриптаmain.py.

Лабораторная работа №6-9. «Математико-статистический анализ экономических данных с помощью NumPy и Pandas»

Цель: Приобрести навыки комплексной обработки больших табличных массивов экономических данных, очистки данных и расчете статистических показателей.
Постановка задачи:
Дан файл macroeconomics_data.csv с квартальными экономическими показателями (ВВП, инфляция, инвестиции, безработица) за 10 лет. Выполнить:
1) Загрузку и заполнение пропущенных значений методами линейной интерполяции;
2) Расчет базовых и цепных темпов роста ВВП;
3) Корреляционный анализ между уровнем инфляции и безработицей (кривая Филлипса) с помощью np.corrcoef() и df.corr();
4) Построение сводной таблицы pivot_table среднего значения показателей по годам.
Результат: Работающий скрипт и аналитическая записка с выводами.

Лабораторная работа №10-13. «Интегрированный мини-проект: Визуальное моделирование и построение аналитического дашборда»

Цель: Освоить методы создания интегрированных отчетов по визуализации экономических процессов средствами Pandas и Matplotlib.

Постановка задачи:

На основе исторических данных о котировках акций и курсах валют построить визуальный отчет, содержащий дашборд из 4 графиков (2x2 на одном рисунке):

1) Динамика цен закрытия акций (линейный график с выделением минимума и максимума);
2) Скользящее среднее (Simple Moving Average 20 и 50 дней);
3) Гистограмма распределения дневной доходности;
4) Диаграмма рассеяния (Scatter plot) связи между курсом доллара и индексом рынка.

Требования: Качественное оформление всех панелей, общая тема оформления, сохранение изображения высокого разрешения dashboard.png и вывод итоговых статистик.

Блок D. Перечень тестовых заданий для промежуточной аттестации (Зачет с оценкой)

Тест промежуточной аттестации включает 30 вопросов закрытого типа с выбором одного правильного ответа из четырех предложенных. Тест сформирован из следующего банка оценочных средств:

Вопрос №1. Какой из перечисленных типов данных в Python является неизменяемым (immutable)?

1) list (список)
2) dict (словарь)
3) tuple (кортеж)
4) set (множество)

Вопрос №2. Какое ключевое слово используется в Python для создания класса?

1) struct
2) class
3) object
4) def

Вопрос №3. Для чего в методах класса Python первым параметром передается self?

1) Для указания на модуль программы
2) Для обращения к текущему экземпляру (объекту) класса
3) Это обязательное системное имя базового класса
4) Для автоматического вызова конструктора

Вопрос №4. Какой специальный метод вызывается при создании нового экземпляра класса (конструктор)?

1) __new__()
2) __init__()
3) __create__()
4) __construct__()

Вопрос №5. Какой соглашение по наименованию атрибутов в Python указывает на то, что атрибут является защищенным (protected)?

1) Имя начинается с одного подчеркивания (_name)
2) Имя начинается с двух подчеркиваний (__name)
3) Имя заканчивается подчеркиванием (name_)
4) Имя пишется прописными буквами (NAME)

Вопрос №6. Какая функция позволяет вызвать метод базового родительского класса из дочернего?

1) parent()
2) super()
3) base()
4) ancestor()

Вопрос №7. Что произойдет при попытке обратиться к приватным атрибутам __balance объекта извне класса напрямую?

1) Вернется значение None
2) Возникнет ошибка AttributeError
3) Атрибут станет публичным
4) Программа автоматически завершится без сообщения

Вопрос №8. Какой декоратор применяется для создания геттера свойства (property) в классе?

1) @staticmethod
2) @classmethod
3) @property
4) @getter

Вопрос №9. Какой магический метод отвечает за пользовательское строковое представление объекта (например, при вызове print(obj))?

1) __str__()
2) __repr__()
3) __string__()
4) __format__()

Вопрос №10. Что означает понятие «Полиморфизм» в контексте ООП в Python?

1) Возможность класса иметь несколько конструкторов
2) Способность объектов с одинаковым интерфейсом иметь разную внутреннюю реализацию
3) Сокрытие внутренних деталей реализации
4) Наследование нескольких классов одновременно

Вопрос №11. Какой оператор используется для обработки исключений в Python?

1) try ... catch
2) try ... except
3) do ... except

4) try ... handle

Вопрос №12. Какой блок в конструкции try-except выполняется ВСЕГДА, независимо от того, возникло исключение или нет?

1) else
2) finally
3) catch
4) always

Вопрос №13. Что делает встроенная функция type(x) в Python?

1) Возвращает размер объекта x в байтах
2) Возвращает тип данных объекта x
3) Преобразует x в строку
4) Проверяет, является ли x числом

Вопрос №14. Каким способом создается одномерный массив NumPy из 5 нулей?

1) np.zeros(5)
2) np.array(0, 5)
3) np.ones(5) * 0
4) np.empty(0, 5)

Вопрос №15. Какой атрибут массива NumPy возвращает его размерность (число строк и столбцов в виде кортежа)?

1) .ndim
2) .size
3) .shape
4) .dtype

Вопрос №16. Какая функция NumPy используется для вычисления вырожденности/определителя квадратной матрицы?

1) np.linalg.det()
2) np.matrix.det()
3) np.det()
4) np.linalg.inv()

Вопрос №17. Какая функция NumPy решает систему линейных алгебраических уравнений A * x = B?

1) np.solve(A, B)
2) np.linalg.solve(A, B)
3) np.linalg.calc(A, B)
4) np.matrix.solve(A, B)

Вопрос №18. Какой результат даст поэлементное умножение двох массивов a * b в NumPy (не матричное)?

1) Скалярное произведение
2) Матричное произведение

3) Поэлементное произведение соответствующих элементов

4) Ошибку типа

Вопрос №19. Какая функция используется в NumPy для генерации массива равномерно распределенных чисел в заданном интервале?

1) np.arange()
2) np.linspace()
3) np.range()
4) np.random()

Вопрос №20. Какая структура данных Pandas представляет собой одноименный одномерный массив с метками (индексами)?

1) DataFrame
2) Series
3) Panel
4) IndexArray

Вопрос №21. Основной двухмерной структурой данных в Pandas с именованными колонками и строками является:

1) Series
2) Matrix
3) DataFrame
4) Tensor

Вопрос №22. Какой метод Pandas используется для чтения данных из файла CSV?

1) pd.load_csv()
2) pd.read_csv()
3) pd.open_csv()
4) pd.parse_csv()

Вопрос №23. Какой индексатор в Pandas выбирает строки и столбцы strictly по их целочисленным позициям (номерам)?

1) .loc[]
2) .iloc[]
3) .ix[]
4) .at[]

Вопрос №24. Какой метод в Pandas заполняет пропущенные значения (NaN) заданным константным значением или методом?

1) df.dropna()
2) df.fillna()
3) df.isna()
4) df.replace_nan()

Вопрос №25. Метод groupby() в Pandas предназначен для:

1) Сортировки значений в колонке

2) Группировки строк по значениям одного или нескольких столбцов для агрегации

3) Слияния двух таблиц

4) Удаления дубликатов

Вопрос №26. Какой метод Pandas позволяет создавать сводные таблицы, аналогичные сводным таблицам в Excel?

1) df.pivot_table()

2) df.cross_tab()

3) df.summary()

4) df.group_pivot()

Вопрос №27. Какая основная библиотека Python используется для построения статических,нимированных и интерактивных графиков?

1) NumPy

2) Matplotlib

3) SciPy

4) SymPy

Вопрос №28. Какая функция Matplotlib непосредственно отображает построенный график на экране?

1) plt.draw()

2) plt.show()

3) plt.plot()

4) plt.display()

Вопрос №29. С помощью какой функции в Matplotlib добавляется заголовок графика?

1) plt.xlabel()

2) plt.title()

3) plt.header()

4) plt.legend()

Вопрос №30. Какой параметр в методе df.plot(kind=...) в Pandas задает построение гистограммы?

1) kind='line'

2) kind='bar'

3) kind='hist'

4) kind='scatter'

КЛЮЧИ ПРАВИЛЬНЫХ ОТВЕТОВ К ТЕСТУ (БЛОК D):

1-3; 2-2; 3-2; 4-2; 5-1; 6-2; 7-2; 8-3; 9-1; 10-2; 11-2; 12-2; 13-2; 14-1; 15-3; 16-1; 17-2; 18-3; 19-2; 20-2; 21-3; 22-2; 23-2; 24-2; 25-2; 26-1; 27-2; 28-2; 29-2; 30-3.

4. МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ, ОПРЕДЕЛЯЮЩИЕ ПРОЦЕДУРЫ ОЦЕНИВАНИЯ ЗНАНИЙ, УМЕНИЙ, НАВЫКОВ И ОПЫТА ДЕЯТЕЛЬНОСТИ

Оценка качества освоения дисциплины включает текущий контроль успеваемости, рубежный контроль и промежуточную аттестацию.

Шкала оценивания выполнения лабораторных и практических работ (умения и навыки)

Показатель	Низкий (0-30)	Фрагментарный (31-59)	Поверхностный (60-69)	Достаточный (70-84)	Высокий (85-100)	Вес
Решение задачи (код Python, алгоритм)	Задача не решена, синтаксические ошибки	Ход верен, но программа не работает	Работает с ошибками на тестах	Работает верно, есть мелкие недочеты	Код идеален, отптимален, документирован	0.6
Оформление и структура кода	Требования PEP8 не выполнены	Код не структурирован, нет комментариев	Частичные комментарии, слабые имена	Соответствует PEP8, есть docstrings	Полное соответствие PEP8 и архитектуре	0.3
Своевременность сдачи	Сдано с опозданием (0 баллов)	Сдано в срок (100 баллов)	—	—	—	0.1
ИТОГОВАЯ ОЦЕНКА	Формула расчета: 0.6 * X1 + 0.3 * X2 + 0.1 * X3					1.0

Балльно-рейтинговая шкала перевода семестрового рейтинга в итоговую оценку

Семестровый рейтинг (баллы)	Академическая оценка	Уровень сформированности компетенции
85 – 100 баллов	«отлично»	Высокий уровень
70 – 84 балла	«хорошо»	Достаточный уровень
60 – 69 баллов	«удовлетворительно»	Поверхностный уровень
менее 60 баллов	«неудовлетворительно»	Низкий / Фрагментарный уровень

5. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ И ВЫПОЛНЕНИЮ КОНТРОЛЬНЫХ ЗАДАНИЙ

Организация учебной работы

Общая трудоемкость дисциплины составляет 3 ЗЕТ (96 учебных часов), включая 16 часов лекций, 32 часа практических (лабораторных) занятий и 47.9 часов самостоятельной работы студентов. Самостоятельная работа предполагает изучение теоретического материала, работу с документацией Python, выполнение практических заданий и подготовку к тестированию.

Инструментарий разработки (IDE и библиотеки)

Для выполнения практических и лабораторных работ рекомендуется использовать современные среды разработки (IDE):

− PyCharm (Community / Professional Edition) — рекомендованная среда для проектирования ООП-приложений;

− Visual Studio Code (VS Code) — с установленными расширениями Python и Jupyter;

− Google Colab / Jupyter Notebook — для интерактивного анализа данных в NumPy и Pandas.

Необходимый стек библиотек: Python 3.9+, NumPy, Pandas, Matplotlib.

Порядок выполнения и защиты лабораторных работ

1. Каждая лабораторная работа выполняется индивидуально согласно варианту.

2. Код должен соответствовать стандарту оформление кода PEP8, содержать подписи авторов, комментарии и docstrings к классам и функциям.

3. Сдача работы происходит в форме демонстрации работающего скрипта преподавателю, ответа на теоретические вопросы по алгоритмам и ООП-архитектуре, а также предоставления исходного файла .py или .ipynb.

4. При защите оценивается правильность работы алгоритма, качество архитектуры классов и обоснованность выбора методов библиотек NumPy/Pandas.

Подготовка к рубежному контролю и зачету

Рубежный контроль проводится в виде тестирования по завершении модулей 1 и 2, а также контрольной работы по модулю 3. Промежуточная аттестация (зачет с оценкой) проходит в форме итогового тестирования (30 вопросов). Для успешной сдачи необходимо проработать контрольные вопросы Блока A, повторить синтаксис Python, методы библиотек NumPy и Pandas и проверить ключи к тестовым заданиям Блока D.