

Оценочные материалы для текущего контроля и промежуточной аттестации по дисциплине

Информация о владельце:
ФИО: Косенок Сергей Михайлович
Должность: ректор
Дата подписания: 11.06.2026 09:49:34
Уникальный программный ключ:
e3a68f3e41a62674b54f4998099d3d6b5fd-f836

пине

Программирование на языке Python

Код, направление подготовки	09.03.04 Программная инженерия
готовки	
Направленность (профиль)	Программное обеспечение компьютерных систем
Форма обучения	Очная
Кафедра-разработчик	Автоматики и компьютерных систем
Выпускающая кафедра	Автоматики и компьютерных систем

Типовые задания для контрольной работы:

Вариант 1

1. Перечислите 3–4 популярные библиотеки Python для обработки и визуализации данных. Кратко (2–3 предложения для каждой) опишите их основное назначение и укажите тип задач, для которых они наиболее подходят. (ОПК-2.1)
2. Опишите ключевые различия между процедурным и объектно-ориентированным подходами в программировании на Python. Кратко (1–2 предложения) поясните, в каких случаях предпочтительнее использовать каждый из подходов. (ПК-1.2)
3. Назовите основные типы UML-диаграмм, которые используются для описания архитектуры веб-приложений. Для каждого типа укажите его назначение (1–2 предложения). Перечислите не менее трёх типов диаграмм. (ОПК-6.2)
4. Опишите стандартный процесс развёртывания простого веб-приложения на локальном хосте с использованием одного из популярных Python-фреймворков (например, Flask или Django): перечислите основные шаги (3–5 пунктов) и кратко (1–2 предложения для каждого шага) поясните, что выполняется на каждом этапе. Укажите, какие инструменты Python используются на каждом шаге. (ПК-3.1, ОПК-6.1)

5. Практическое задание.

Разработайте небольшое веб-приложение на Flask, которое выполняет следующие функции:

- отображает главную страницу с приветственным текстом и формой для загрузки CSV-файла;
- после загрузки файла приложение считывает данные с помощью библиотеки Pandas;
- вычисляет среднее значение для числового столбца (если в файле несколько числовых столбцов, выберите первый);
- выводит результат на отдельной странице вместе с первыми 5 строками загруженных данных.

Требования:

- код должен быть структурирован и содержать комментарии к ключевым блокам (2–3 слова для пояснения назначения блока);
- реализуйте обработку ошибок: если файл не CSV или не содержит числовых данных, то приложение должно выводить краткое сообщение об ошибке с соответствующим содержанием;
- приложение должно запускаться на локальном хосте (порт 5000).

Предоставьте полный код приложения (не более 60 строк) и краткое описание (3–4 предложения) того, как проверить его работу. (ОПК-2.1, ОПК-2.2, ОПК-6.1, ПК-3.1)

Вариант 2

1. Объясните назначение и основные возможности библиотеки NumPy. Кратко (1–2 предложения) поясните, в каких задачах обработки данных использование NumPy предпочтительнее, чем стандартных структур данных Python. (ОПК-2.1, ОПК-2.2)
2. Опишите основные методы работы с файлами в Python. Укажите, какие исключения могут возникнуть при работе с файлами, и как их можно обработать (кратко, 2–3 предложения). (ОПК-2.2)
3. Опишите стандартный подход к написанию тестов с использованием модуля unittest или pytest. Кратко (1–2 предложения) объясните, зачем нужно автоматизировать тестирование. (ОПК-6.2, ПК-3.1)
4. Объясните, что такое веб-фреймворк и зачем он нужен. Сравните два популярных Python-фреймворка (Flask и Django) по следующим критериям: сложность освоения, гибкость настройки, наличие встроенных инструментов (например, админ-панели, ORM). Сделайте вывод (1–2 предложения), для каких проектов лучше подходит каждый фреймворк. (ОПК-6.1, ОПК-2.1)

5. Практическое задание.

Разработайте консольное приложение на Python, которое:

- запрашивает у пользователя путь к CSV-файлу;
- после загрузки файла приложение считывает данные с помощью библиотеки Pandas;
- фильтрует строки, где значение в определённом столбце превышает заданный порог (порог и столбец задаются пользователем);
- выводит отфильтрованные данные в консоль в виде таблицы;
- сохраняет отфильтрованные данные в новый CSV-файл.

Требования:

- код должен быть структурирован и содержать комментарии к ключевым блокам (2–3 слова для пояснения назначения блока);
- реализуйте обработку ошибок: если файл не найден или столбец отсутствует, то приложение должно выводить краткое сообщение об ошибке с соответствующим содержанием;
- при разработке программного кода используйте процедурный подход (используйте функции для разделения логики на блоки (загрузка, фильтрация, сохранение)).

Предоставьте полный код приложения (не более 50 строк) и краткое описание (3–4 предложения) того, как проверить его работу. (ОПК-2.2, ПК-1.2)

Вариант 3

1. Что такое функция в Python? Объясните, как можно импортировать отдельные функции из модуля NumPy или Pandas. Приведите три примера импорта функций из указанных модулей и кратко (1–2 предложения для каждого) опишите назначение импортированных функций. (ОПК-2.2)
2. Дайте определения следующим понятиям из объектно-ориентированного программирования: класс, объект, наследование. Кратко (1–2 предложения) объясните, как использование принципа наследования может упростить разработку приложения. (ПК-1.2)
3. Что такое UML-диаграмма классов? Нарисуйте (опишите текстом) простую UML-диаграмму для системы «Библиотека», включающую не менее трёх классов (например, «Книга» (Book), «Читатель» (Reader), «Библиотека» (Library)). Для каждого класса укажите 2–3 атрибута и 1–2 метода. Кратко (1–2 предложения) объясните, как такая диаграмма помогает в проектировании приложения. (ОПК-6.2)
4. Опишите процесс настройки базового автоматизированного тестирования для Python-приложения с использованием unittest или pytest. (ОПК-6.2, ПК-3.1)

5. Практическое задание.

Разработайте веб-приложение на Flask, которое:

- отображает страницу с формой для ввода данных о книге (название, автор, год издания);
- после отправки формы сохраняет данные в список (в памяти приложения);
- отображает список всех добавленных книг на отдельной странице;

- позволяет удалить книгу из списка (через кнопку рядом с каждой записью).

Требования:

- код должен быть структурирован и содержать комментарии к ключевым блокам (2–3 слова для пояснения назначения блока);
- приложение должно запускаться на локальном хосте (порт 5000);
- реализуйте базовую обработку ошибок (например, если поля формы пустые): приложение должно выводить краткое сообщение об ошибке с соответствующим содержанием.

Предоставьте полный код приложения (не более 70 строк) и краткое описание (3–4 предложения) того, как проверить его работу. (ОПК-6.1, ПК-3.1, ПК-1.2)

Вариант 4

1. Опишите назначение библиотеки Pandas и её основные структуры данных (DataFrame и Series). Кратко (1–2 предложения) объясните, почему Pandas удобна для анализа данных. (ОПК-2.1, ОПК-2.2)
2. Что такое механизм исключений в Python? Перечислите три распространённых типа исключений, которые могут возникнуть при работе с файлами, и кратко (1–2 предложения для каждого) опишите, когда они возникают. (ОПК-2.2)
3. Объясните, что такое архитектура веб-приложений и какие основные компоненты она включает. (ОПК-6.1)
4. Сравните подходы процедурного и объектно-ориентированного программирования в контексте разработки веб-приложений на Python. Укажите два преимущества и два недостатка каждого подхода (представьте в виде таблицы). Сделайте вывод (1–2 предложения), какой подход лучше подходит для небольших веб-приложений, а какой – для крупных. (ПК-1.2, ОПК-6.1)
5. **Практическое задание.**

Разработайте скрипт на Python, который:

- генерирует случайный набор данных (например, список пользователей с именами и возрастaми) и сохраняет его в CSV-файл;
- считывает этот файл с помощью Pandas;
- вычисляет средний возраст и находит самого старшего пользователя;
- выводит результаты в консоль и сохраняет их в отдельный текстовый файл.

Требования:

- код должен быть структурирован и содержать комментарии к ключевым блокам (2–3 слова для пояснения назначения блока);
- реализуйте обработку ошибок (например, если файл не создан): приложение должно выводить краткое сообщение об ошибке с соответствующим содержанием;
- при разработке программного кода используйте процедурный подход (используйте функции для разделения логики на блоки (генерация, загрузка, вычисление, сохранение)).

Предоставьте полный код скрипта (не более 60 строк) и краткое описание (3–4 предложения) того, как проверить его работу. (ОПК-2.2, ПК-1.2)

Вариант 5

1. Объясните, что такое JSON и как он используется в веб-разработке. Опишите, как в Python работать с JSON-данными. (ОПК-2.2, ОПК-6.1)
2. Перечислите шаги по созданию, активации и деактивации виртуального окружения. Кратко (2–3 предложения) опишите, чем может быть полезно использование виртуальных окружений при разработке и развёртывании веб-приложений. (ПК-3.1)
3. Что такое UML-диаграмма последовательности (Sequence Diagram)? Кратко (1–2 предложения) объясните, как эта диаграмма помогает разработчикам понимать логику взаимодействия компонентов. (ОПК-6.2)
4. Сравните подходы к тестированию веб-приложений: ручное тестирование и автоматизированное тестирование. Для этого составьте таблицу с двумя колонками («Ручное тестирование» и «Автоматизированное тестирование») и тремя строками («Преимущества», «Недостатки»,

«Подходящие сценарии»). Кратко (1–2 предложения) сделайте вывод, когда предпочтительнее использовать каждый подход. (ОПК-6.2, ПК-3.1)

5. Практическое задание.

Разработайте консольное приложение на Python, которое:

- запрашивает у пользователя путь к файлу, содержащему произвольный текст из некоторого набора слов;
- считывает содержимое файла, удаляя знаки препинания и приводя слова к нижнему регистру;
- с помощью библиотеки Pandas создаёт DataFrame, где каждая строка – это слово из файла;
- подсчитывает частоту встречаемости каждого слова и выводит 5 самых частых слов с указанием количества их вхождений;
- сохраняет результаты в новый CSV-файл с колонками «слово» и «частота»;
- для обработки текста можно использовать модуль `re` или встроенные методы строк.

Требования:

- код должен быть структурирован и содержать комментарии к ключевым блокам (2–3 слова для пояснения назначения блока);
- реализуйте обработку ошибок (если файл не найден, имеет неверный формат или пуст): приложение должно выводить краткое сообщение об ошибке с соответствующим содержанием;
- при разработке программного кода используйте процедурный подход (используйте функции для разделения логики на блоки (чтение файла, обработка текста, подсчёт частоты, сохранение результата)).

Предоставьте полный код приложения (не более 60 строк) и краткое описание (3–4 предложения) того, как проверить его работу. (ОПК-2.1, ОПК-2.2, ПК-1.2)

Типовые вопросы и практические задания к экзамену:

Задание на экзамене содержит 2 теоретических вопроса и практическую задачу.

Задание для показателя оценивания дескриптора «Знает»	Вид задания	Уровень сложности
<p><u>Вариант 1</u></p> <p>1. Перечислите три популярные библиотеки Python для обработки и визуализации данных. Кратко охарактеризуйте функциональное назначение каждой из них.</p> <p>2. Определите понятие «бэкенд» и перечислите основные задачи, решаемые бэкенд-частью при обработке пользовательских запросов.</p>	теоретический	репродуктивный
<p><u>Вариант 2</u></p> <p>1. Опишите алгоритм создания одномерного массива в библиотеке NumPy при работе с числовыми данными.</p> <p>2. Перечислите три основных типа UML-диаграмм, используемых при проектировании ПО, и опишите назначение каждой из них при документировании архитектуры системы.</p>		
<p><u>Вариант 3</u></p> <p>1. Опишите алгоритм добавления нового столбца в существующий DataFrame Pandas. Перечислите операции преобразования данных</p>		

2. Опишите элементы диаграммы классов UML и их роль при моделировании статической структуры системы.

Вариант 4

1. Перечислите категории способов преобразования DataFrame Pandas. Для каждой категории приведите пример метода, выполняющего преобразование и опишите действие, которое он выполняет.
2. Опишите структуру простого тестового примера для функции сложения двух чисел с использованием pytest при решении задачи проверки корректности математических операций.

Вариант 5

1. Опишите последовательность действий для чтения текстового файла в Python построчно в контексте обработки больших текстовых файлов.
2. Сравните процедурный и объектно-ориентированный подходы к программированию на Python, выделив основные различия организации кода и управления данными при использовании каждой из них.

Вариант 6

1. Опишите механизм обработки ошибок открытия файла в Python с использованием конструкции try-except как средства обеспечения устойчивости программы к ошибкам ввода-вывода.
2. Перечислите инструменты мониторинга и логирования, которые могут использоваться для Python-веб-приложений после развёртывания, и кратко объясните назначение каждого из них.

Вариант 7

1. Перечислите основные компоненты классической трёхуровневой архитектуры веб-приложения и опишите их роль в архитектуре.
2. Перечислите основные принципы процедурного подхода к разработке программного обеспечения и приведите примеры их реализации в Python.

Вариант 8

1. Опишите основные функциональные возможности фреймворка Django

2. Объясните назначение диаграммы последовательностей UML и приведите пример сценария взаимодействия компонентов, который удобно отобразить с её помощью при проектировании веб-приложения.

Вариант 9

1. Опишите алгоритм фильтрации строк в DataFrame Pandas по заданному условию при обработке табличных данных.
2. Определите понятия «класс» и «объект» в Python и опишите их взаимосвязь.

Вариант 10

1. Перечислите основные режимы открытия файлов в Python и объясните их назначение в контексте различных сценариев работы с файлами.
2. Объясните концепцию фикстуры (fixture) в pytest и опишите её назначение.

Вариант 11

1. Опишите принцип работы клиент-серверной архитектуры на примере взаимодействия компонентов веб-приложения на Python.
2. Опишите способ обозначения наследования на UML-диаграмме классов и объясните его семантику.

Вариант 12

1. Опишите основные функциональные возможности фреймворка Flask.
2. Опишите особенности реализации принципа наследования в Python.

Вариант 13

1. Опишите порядок выполнения поэлементного умножения двух массивов одинаковой размерности в библиотеке NumPy.
2. Определите понятие виртуального окружения (virtual environment) в Python и объясните его назначение.

Вариант 14

1. Сравните методы read(), readline() и readlines() при работе с файлами в Python, указав особенности их применения при чтении файлов с разным объёмом данных.
2. Перечислите опции командной строки для запуска тестов с помощью pytest, указав их назначение.

<p><u>Вариант 15</u></p> <ol style="list-style-type: none"> Опишите роль веб-сервера в архитектуре веб-приложения и назовите один популярный веб-сервер для Python-приложений для развёртывания веб-сервисов. Назовите один инструмент для контейнеризации Python приложений и опишите его особенности. <p><u>Вариант 16</u></p> <ol style="list-style-type: none"> Перечислите библиотеки Python, предназначенные для визуализации данных, и укажите их функциональные возможности по построению графиков и диаграмм. Перечислите ключевые преимущества фреймворка pytest перед стандартным модулем unittest. 		
--	--	--

Задание для показателя оценивания дескриптора «Умеет»	Вид задания	Уровень сложности
<p><u>Вариант 1</u></p> <p>Задание.</p> <p>1) Напишите программу на Python, которая:</p> <ul style="list-style-type: none"> считывает данные из CSV-файла «Данные о продажах» (<i>sales_data.csv</i>) (колонки: Дата (<i>date</i>), Товар (<i>product</i>), Количество (<i>quantity</i>), Цена (<i>price</i>)). Содержание файла сгенерировать самостоятельно в соответствии с указанной структурой, типом содержимого полей, создав для этого соответствующую функцию. Количество записей для генерации не менее 100; с помощью библиотеки <i>pandas</i> вычисляет общую выручку по каждому продукту; выводит результат в консоль в виде таблицы; сохраняет результат в новый CSV-файл «Совокупный доход» (<i>total_revenue.csv</i>). <p>При разработке программного кода используйте процедурный подход (используйте функции для разделения логики на блоки).</p> <p>2) Дополнительно: создайте UML-диаграмму классов, описывающую архитектуру приложения (классы для чтения данных, обработки и сохранения).</p> <p><u>Вариант 2</u></p> <p>Задание.</p> <p>1) Создайте простое веб-приложение на Flask, которое:</p> <ul style="list-style-type: none"> отображает страницу с формой для ввода имени и возраста; 	<p>практический</p>	<p>конструктивный, творческий</p>

- после отправки формы проверяет, что возраст является целым числом от 20 до 80, а имя состоит только из букв русского алфавита. Для проверки должны быть реализованы соответствующие функции проверки;
- если данные корректны, выводит приветственное сообщение: «Привет, [имя]! Вам [возраст] лет.»;
- если данные некорректны, показывает соответствующее сообщение об ошибке;
- должно запускаться на локальном хосте (порт 5000).

2) Дополнительно: напишите два теста с использованием `pytest` для функции проверки возраста.

Вариант 3

Задание.

Дан текстовый файл `log.txt` с записями вида:

```
Дата      Время  УрЛог Сообщение
2023-10-01 10:00:00 INFO User login
2023-10-01 10:05:00 ERROR Database connection failed
```

Напишите скрипт на Python, который:

- читает файл построчно;
- извлекает дату, время, уровень логирования и сообщение;
- группирует записи по уровню логирования (*INFO*, *ERROR*, *WARNING*);
- для каждого уровня логирования: подсчитывает количество записей, находит самое раннее и самое позднее время записи;
- выводит в консоль: общее количество записей, количество записей для каждого уровня, самое раннее и самое позднее время записи для каждого уровня, список всех сообщений уровня *ERROR* (при наличии);
- реализует обработку ошибок (если файл не найден, имеет неверный формат или пуст): приложение должно выводить краткое сообщение об ошибке с соответствующим содержанием.

При разработке программного кода используйте процедурный подход (используйте функции для разделения логики на блоки).

Вариант 4

Задание.

1) Используя библиотеку `numpy`, создайте массив 10×10 случайных чисел от 0 до 100. Выполните следующие операции:

- найдите среднее значение и стандартное отклонение массива;
- замените все значения, превышающие среднее, на 0;
- для измененного массива найдите: количество нулевых элементов, новое среднее значение, новое стандартное отклонение;

- выведите в консоль: исходный и изменённый массивы, все вычисленные статистические показатели (количество нулевых элементов, среднее значение, стандартное отклонение).

2) Дополнительно: постройте столбчатую диаграмму (с помощью *matplotlib*) для сравнения среднего значения и стандартного отклонения. Для диаграммы необходимо задать: заголовок, подписи осей, сетку.

При разработке программного кода используйте процедурный подход (используйте функции для разделения логики на блоки).

Вариант 5

Задание.

1) Разработайте класс «Учетная запись в банке» (*BankAccount*) с методами:

- *__init__(owner, initial_balance=0)* (инициализация счёта с указанием владельца и начального баланса);
- *deposit(amount)* (пополнение счёта, проверка на положительность суммы);
- *withdraw(amount)* (снятие средств, проверка на достаточность баланса и положительность суммы);
- *get_balance()* (возвращает текущий баланс);
- *transfer(to_account, amount)* (перевод средств на другой счёт, с проверкой корректности операции).

2) Дополнительно:

1. Создайте UML-диаграмму классов, описывающую архитектуру приложения (класс *BankAccount*, возможные наследники — например, «Учетная запись счета» *SavingsAccount* с методом *add_interest()*).
2. Создайте экземпляры двух счетов, выполните несколько операций пополнения, снятия и перевода между ними, выведите баланс после каждой операции.
3. Напишите три теста с *pytest* для класса *BankAccount*:
 - тест для метода *withdraw* (проверка снятия суммы, превышающей баланс);
 - тест для метода *deposit* (проверка отрицательной суммы);
 - тест для метода *transfer* (проверка корректности перевода и обновления балансов).
4. Сохраните историю операций (тип операции, сумма, дата и время) в файл «Лог транзакций» (*transaction_log.txt*).

Вариант 6

Задание.

1) Создайте веб-приложение на Flask с тремя маршрутами:

- / — главная страница с приветствием и ссылкой на страницы статистики и визуализации;

- `/stats` — страница, которая читает данные из JSON-файла `users.json` (массив объектов с полями `name`, `age`, `city`, `salary`), вычисляет:
 - средний возраст;
 - среднюю зарплату;
 - количество пользователей по городам (группировка);
- `/plot` — страница, отображающая столбчатую диаграмму распределения зарплат по городам (используйте `matplotlib` и `pandas` для агрегации данных).

2) Дополнительно:

1. Разверните приложение на локальном хосте (`localhost:5000`).
2. Напишите два теста с `pytest` для функции агрегации данных (проверка среднего возраста и группировки по городам).
3. Создайте UML-диаграмму последовательностей для сценария «Просмотр статистики пользователей».

Вариант 7

Задание.

1) Напишите скрипт, который:

1. Создает 10 текстовых файлов с именами `file_1.txt`, `file_2.txt`, ..., `file_10.txt`. В каждый файл записывает случайное количество строк (от 5 до 15) с текстом «Это строка [номер] в файле [имя файла]».
2. Собирает содержимое всех файлов в один итоговый файл `combined.txt`, добавляя имя исходного файла и номер строки перед каждой строкой (например, «`file_1.txt 1 Это строка 1 в файле file_1.txt`», ..., «`file_10.txt 100 Это строка 1 в файле file_10.txt`»).
3. Анализирует `combined.txt`:
 - подсчитывает общее количество строк;
 - находит самые длинные и самые короткие строки.
4. Показывает прогресс выполнения (например, «Создан файл `file_3.txt`», «Анализ завершён») в консоли.

2) Дополнительно:

1. Для файла `combined.txt` составить частотный словарь слов (без учёта регистра и знаков препинания)
2. Сохраните результаты анализа в JSON-файл `analysis_report.json` со структурой:

```
{
  "total_lines": 120,
  "longest_line": "...",
  "shortest_line": "...",
  "word_frequency": {"это": 120, "строка": 120, ...}
}
```

3. Реализуйте обработку ошибок (если файл не найден, имеет неверный формат или пуст): приложение должно

выводить краткое сообщение об ошибке с соответствующим содержанием.

При разработке программного кода используйте процедурный подход (используйте функции для разделения логики на блоки).

Вариант 8

Задание.

1) Создайте простое веб-приложение на Flask для анализа и визуализации температурных данных. Приложение должно содержать два маршрута:

1. / — главная страница с:

- кратким описанием функционала приложения;
- формой для ручного ввода температур за неделю (7 полей для чисел);
- кнопкой «Построить графики» (открывает страницу результатов).

2. /analyze — страница результатов, при запуске которой:

- считываются данные из формы (или использует предустановленные значения, если форма не заполнена): `temperatures = [20, 22, 19, 24, 26, 23, 21]` # температура за неделю; `days = ['Понедельник', 'Вторник', 'Среда', 'Четверг', 'Пятница', 'Суббота', 'Воскресенье'];`
- вычисляется: максимальную и минимальную температуру, день с максимальной температурой, среднее значение температуры;
- строится график: линейный график температуры по дням (с маркерами, заголовком, подписями осей, сеткой);
- на странице отображается: вычисленные статистические показатели, построенный график.

2) Дополнительно:

1. Разверните приложение на локальном хосте (`localhost:5000`).

2. Создайте функцию `export_to_csv(data, filename)`, которая:

- принимает словарь с данными (дни и температуры);
- сохраняет их в CSV-файл «Данные по температуре» (`temperature_data.csv`);
- добавляет кнопку «Экспорт в CSV» на страницу /analyze, которая запускает эту функцию.

3. Напишите один тест с `pytest` для функции вычисления среднего значения температуры (проверьте на тестовом массиве `[20, 25, 30]` — ожидаемый результат 25.0).

Пример вывода на странице /analyze:

Статистические показатели:

* Максимальная температура: 26 °C (Пятница)

* Минимальная температура: 19 °C (Среда)

* Средняя температура: 22.1 °C

[Изображение графика temperature_analysis.png]

<Кнопка «Экспорт в CSV»>

Вариант 9

Задание.

1) Создайте класс «Книга» (*Book*) с атрибутами «Название» (*title*), «Автор» (*author*), «Год издания» (*year*), «Жанр» (*genre*) и методами:

- *info()* — возвращает строку вида «*[title]* (*[year]*) – *[author]* (*[genre]*)»;
- *is_classic()* — возвращает *True*, если книга старше 50 лет.

2) Дополнительно:

1. Напишите функцию, которая:
 - принимает список словарей с данными о книгах. *Список словарей предварительно либо считывается из заранее подготовленного файла, либо вводится пользователем с помощью вспомогательной функции, создающей словари и объединяющей их в список;*
 - создаёт экземпляры класса *Book* для каждого словаря;
 - возвращает два списка: классические книги и современные.
2. Протестируйте функцию на примере списка из пяти книг (две классические, три современные).
3. Создайте UML-диаграмму классов для системы библиотеки (классы «Книга» (*Book*), «Библиотека» (*Library*), методы добавления/удаления книг, поиска по жанру).
4. Сохраните список классических книг в JSON-файл *classics.json*, а современных — в *modern.json*.

Вариант 10

Задание.

1) Разработайте веб-приложение на Flask для учёта и анализа успеваемости студентов. Приложение должно позволять вводить и просматривать оценки по нескольким предметам.

1. Создайте маршрут / (главная страница), который отображает:
 - приветственное сообщение «Система учёта успеваемости»;
 - ссылку на страницу добавления оценок (*/add_grade*);
 - ссылку на страницу статистики (*/stats*).
2. Создайте маршрут */add_grade*, который:
 - отображает форму с полями:

- «Имя студента» (текстовое поле);
- «Предмет» (выпадающий список: Математика, Физика, Программирование, Английский);
- «Оценка» (число от 2 до 5);
- при отправке формы сохраняет данные в CSV-файл `grades.csv` (колонки: «Имя студента» (`student_name`), «Предмет» (`subject`), «Оценка» (`grade`));
- после сохранения выводит сообщение «*Оценка успешно добавлена!*» и показывает ссылку на главную страницу.

3. Создайте маршрут `/stats`, который:

- читает данные из `grades.csv`;
 - с помощью библиотеки `pandas` вычисляет для каждого студента:
 - средний балл по всем предметам;
 - количество сданных предметов;
 - для каждого предмета вычисляет средний балл по студентам;
 - отображает на странице:
 - таблицу с результатами по студентам (имя, средний балл, количество предметов);
 - таблицу средних баллов по предметам.
4. Разверните приложение на локальном хосте (`localhost:5000`).

2) Дополнительно:

1. При первом запуске, если файла `grades.csv` не существует, приложение должно создавать пустой файл с заголовками колонок.
2. Обработайте ошибку ввода некорректной оценки (меньше 2 или больше 5): выводите под формой сообщение «Ошибка: оценка должна быть от 2 до 5».