

Итоговое домашнее задание

Дисциплина	Программирование на Python
Тема	Итоговое домашнее задание
Форма проверки	Проверка преподавателем
Имя преподавателя	Филипп Воронов
Время выполнения	4 часа
Цель задания	Проверить знания и умения по дисциплине «Программирование на Python»
Инструменты для выполнения ДЗ	Python, API VirusTotal, Vulners API
Правила приёма работы	Выполните задание и загрузите код в GitHub-репозиторий. Прикрепите в LMS ссылку на GitHub-репозиторий с выполненным заданием. Важно: убедитесь, что по ссылке открыт доступ
Критерии оценивания	Максимальное количество баллов за итоговое задание — 10. Критерии оценивания приведены после описания задания. Задание считается не выполненным , если репозиторий с заданием не прикреплен или к нему нет доступа по ссылке
Дедлайн	Срок сдачи — 24.12.2024

Описание задания

Напишите HTTP-сервер, который управляет списком дел пользователя.

Каждое дело должно характеризоваться четырьмя полями:

- title — название задачи;
- priority — текстовое описание приоритета (low, normal, high);
- isDone — выполнено ли задание;
- id — уникальное число — идентификатор задачи.

Реализуйте API со следующими функциями:

1. **Создание задачи.** Отправляется POST-запрос на сервер по пути /tasks с телом в виде JSON с полями title и priority. Сервер должен создать и сохранить задачу, выставить isDone в False, выдать задаче уникальный ID и в ответ отправить JSON со всеми четырьмя полями.
2. **Получение списка всех задач.** Отправляется GET-запрос на сервер по пути /tasks, в ответ приходит список задач в формате JSON. Пример: [{"title": "Gym", "id": 1, "priority": "low", "isDone": false}, {"title": "Buy a laptop", "id": 2, "priority": "high", "isDone": true}].
3. **Отметка о выполнении задачи.** Отправляется POST-запрос на сервер по пути /tasks/id/complete, где id — это уникальный идентификатор задачи. В ответ приходит пустое тело ответа с кодом 200, если всё успешно, и кодом 404, если такой задачи нет.

После каждого запроса, который меняет список задач, сервер должен всё сохранять в файл tasks.txt. При старте сервер должен проверять наличие этого файла и, если он есть, восстанавливать данные о задачах из него.

Критерии оценивания

Критерий	Описание и количество баллов	Максимальный балл
Функциональность	<ul style="list-style-type: none">• Задание полностью выполнено, API работает корректно и выполняет все необходимые функции без ошибок — 3 балла.• API выполняет только часть функций, но работает без ошибок — 1–2 балла.• Задание выполнено не полностью или с ошибками, из-за которых API не функционирует — 0–1 балл	3
Качество кода	<ul style="list-style-type: none">• Код читабелен, написан логично, с соблюдением принципов ООП — 3 балла.• Код написан достаточно читабельно и логично, но с небольшими нарушениями стандартов — 1–2 балла.• Код написан некорректно, трудно читаем и не соответствует принципам ООП — 0 баллов	3

Разнообразие функций и использование библиотек	<ul style="list-style-type: none"> Решение демонстрирует широкий спектр функций и использование разных библиотек — 2 балла. Решение содержит ограниченное количество функций или в нём использованы только базовые библиотеки — 1 балл. Решение выполняет только одну конкретную функцию без использования библиотек — 0 баллов 	2
Дополнительные функции и качество решения	<ul style="list-style-type: none"> Решение дополнительно реализует необходимые функции или предполагает эффективный и творческий подход, демонстрирующий глубокое понимание Python — 0–2 балла 	2
Итого		10