

# **Домашнее задание 1**

**Тема:** Введение в бэкенд. Асинхронность в Python

**Форма проверки:** самопроверка

**Имя преподавателя:** Геннадий Осипов

**Время выполнения:** 2 академических часа

**Цель задания:**

Научиться создавать простое FastAPI-приложение.

**Инструменты для выполнения ДЗ:** Visual Studio Code, PyCharm, Python, FastAPI.

**Правила приёма работы:**

Прикрепите ссылку на выполненное задание в GitHub. Убедитесь, что доступ к репозиторию открыт. Отправьте работу на проверку.

**Критерии оценки:**

Задание считается выполненным:

- если прикреплена ссылка на код с выполненным заданием,
- доступ к файлу открыт,
- код выполняется и даёт правильный ответ к задаче.

Задание считается невыполненным:

- если ссылка на код с выполненным заданием не прикреплена,
- отсутствует доступ по ссылке,
- код не выполняется.

**Дедлайн:** 7 дней после открытия задания на платформе

## **Задание**

Создайте FastAPI-приложение, которое показывает, как работает асинхронность в Python.

Приложение должно реализовывать эндпоинт: POST /calculate/

**Что делает:** принимает список чисел и задержек, считает квадраты этих чисел асинхронно.

1. Пример запроса:

```
{  
    "numbers": [5, 3, 10],  
    "delays": [1, 2, 0.5]  
}
```

2. Пример ответа:

```
{  
    "results": [  
        {"number": 5, "square": 25, "delay": 1, "time":  
        1.02},  
        {"number": 3, "square": 9, "delay": 2, "time":  
        2.01},  
        {"number": 10, "square": 100, "delay": 0.5,  
        "time": 0.51}  
    ],  
    "total_time": 2.03,  
    "parallel_faster_than_sequential": true  
}
```

## Чек-лист самопроверки

- Приложение реализует метод, возвращающий квадраты чисел и общее время выполнения.
- Реализованный метод — асинхронный.
- Входные и выходные модели описаны классами (Pydantic).