Задание 1: Декоратор для кэширования результатов функции

Создайте класс декоратор cache, который будет кэшировать результаты вызовов функции. Декоратор должен сохранять результаты в файле формата json по аргументам функции и возвращать закэшированное значение, если функция была вызвана с теми же аргументами.

Дополнительно:

- Реализуйте механизм очистки кэша по истечении определенного времени.
- Добавьте возможность ограничения максимального количества элементов в кэше.

Задание 2: Декоратор для логирования вызовов функций

Напишите функцию декоратор log_calls, который будет записывать информацию о каждом вызове функции в json файл. В лог должны включаться:

- Имя функции
- Аргументы, с которыми была вызвана функция
- Время вызова

Дополнительно:

- Реализуйте возможность фильтрации логов по времени и имени функции.
- Добавьте уровень логирования (info, warning, error).

Задание 3: Декоратор для контроля времени выполнения функции

Создайте класс декоратор timing, который будет измерять время выполнения функции и выводить его в консоль. Если функция выполняется дольше заданного порога (например, 1 секунда), выводите предупреждение.

Дополнительно:

- Добавьте возможность задания порога времени в качестве аргумента декоратора.
- Реализуйте возможность сохранения результатов в файл.

Задание 4: Класс-декоратор для ограничения числа вызовов функции

Создайте класс-декоратор LimitCalls, который будет ограничивать количество вызовов функции до заданного максимума. Если функция вызывается больше раз, чем разрешено, выбрасывайте исключение.

Дополнительно:

- Реализуйте возможность сброса счетчика вызовов.
- Добавьте параметр для задания времени, в течение которого будет действовать ограничение (например, "разрешить 5 вызовов в течение 10 секунд").
-