Задание 1.

Ситуация: мы работаем с текстовыми файлами и часто открываем и закрываем их. Чтобы избежать ошибок, связанных с забытым закрытием файла, используем менеджеры контекста.

Задача — написать код, который создаёт файл, записывает в него строку, а затем считывает содержимое и выводит на экран. Использовать конструкцию with для работы с файлом.

Задание 2

Ситуация: мы работаем с подключениями к базе данных и должны гарантировать, что соединение будет закрыто после работы независимо от того, возникли ли ошибки.

Задача — реализовать менеджер контекста для управления соединением с базой данных (имитация).

Задание 3

Задача: реализовать менеджер контекста для работы с кэшом. Менеджер должен хранить кэш в словаре и автоматически очищать его при выходе из блока with.

Задание 4

Задача: написать класс ConfigManager, который будет использоваться как менеджер контекста для временного изменения конфигурации приложения. Класс должен содержать атрибут конфигурации (строку), а при инициализации получать новое значение для него. Во время выполнения блока with конфигурация должна меняться на новую, а при выходе из него принимать начальное значение.

Задание 5

Задача: реализовать менеджер контекста, который будет пытаться (использовать библиотеку random) несколько раз установить соединение с API в случае неудачи. Если соединение не удаётся установить после нескольких попыток, программа должна выбросить исключение.