Задача 1: Перебор данных из списка заказов

Ситуация: мы работаем над приложением для кафе. У нас есть список заказов клиентов, и мы хотим обработать их по одному, чтобы избежать перегрузки системы при обработке всех заказов сразу.

Задача: создайте программу, которая:

- Использует итератор для последовательного перебора списка заказов.
- Выводит каждый заказ по одному с помощью next().
- Обрабатывает ошибку StopIteration, если заказы закончились.

Задача 2: Перебор шагов инструкции

Ситуация: мы работаем над приложением для кухни, где повара должны выполнять шаги рецепта по одному. Нам нужно создать программу, которая помогает повару видеть текущий шаг инструкции, а после завершения всех шагов сообщает, что рецепт закончен.

Задача: создайте программу, которая:

- Использует итератор для последовательного перебора шагов рецепта.
- Выводит текущий шаг по одному с помощью next().
- Обрабатывает ошибку StopIteration, если шаги закончились.

Задача 3: Генератор чисел с проверкой кратности

Ситуация: мы пишем программу, которая помогает школьникам тренироваться в математике. Нам нужно создать генератор чисел, который выдаёт последовательность чисел, кратных определённому значению (например, 3). Ученики смогут указать верхний предел последовательности.

Задача: напишите генератор multiples(limit, divisor), который принимает два аргумента:

- limit — максимальное число последовательности.
- divisor — число, на которое проверяется кратность.

Генератор должен возвращать числа от 1 до limit, которые делятся на divisor без остатка.

Ожидаемый результат:

Если пользователь ввёл limit = 10 и divisor = 3, вывод:

Числа от 1 до 10, кратные 3:
3
6
9

Задача 4: Генератор чётных чисел

Ситуация: мы хотим разработать генератор, который выводит только чётные числа в заданном диапазоне. Это полезно для экономии времени и ресурсов, когда нужны только определённые данные из большого набора.

Задача: напишите генератор even_numbers(start, end), который принимает два аргумента:

- start — начало диапазона.
- end — конец диапазона.

Генератор должен возвращать только чётные числа в указанном диапазоне.

Ожидаемый результат:

Если пользователь ввёл start = 5 и end = 12, вывод:

Чётные числа от 5 до 12:
6
8
10
12