МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ

КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ

Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования

**«Московский технический университет связи и информатики»**

Кафедра «Математическая кибернетика и информационные технологии»

Лабораторная работа №8

**по дисциплине**

**«Информационные технологии и программирование»**

Выполнил: студент гр. БВТ2201

Аблязов И.В.

Проверил:

Москва, 2023 г.

# ОСНОВНАЯ ЧАСТЬ

В данной лабораторной работе необходимо разработать приложение, которое считывает данные из исходного источника, применяет к ним различные операции с использованием Stream API, и сохраняет результаты в новый источник данных.

Для начала была создана аннотация @DataProcessor.

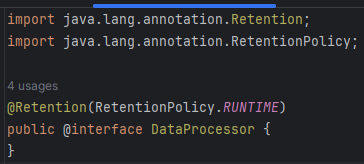


Рисунок 1 Аннотация DataProcessor

После были реализованы три разных класса, представляющие различные обработчики данных: NameSort – сортировка входных данных по имени, ValueSort – сортировка входных данных по количеству и MoreThamTen – вывод только тех данных, значение которых больше 10. Классы использует в себе Stream API для улучшенной работы

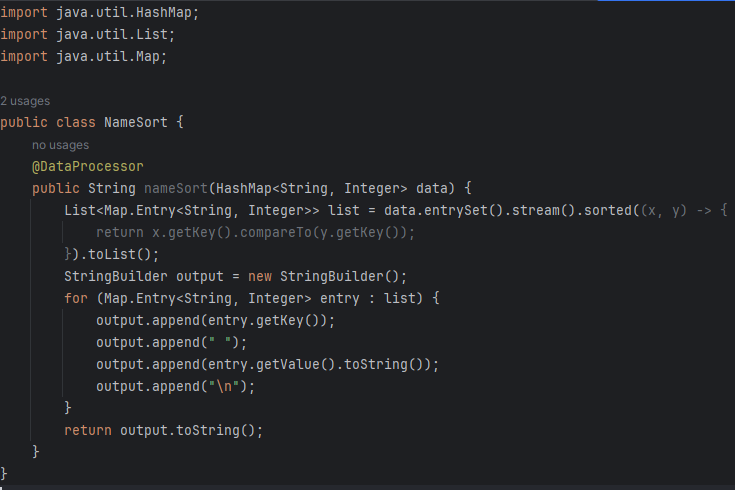


Рисунок 2 Класс NameSort

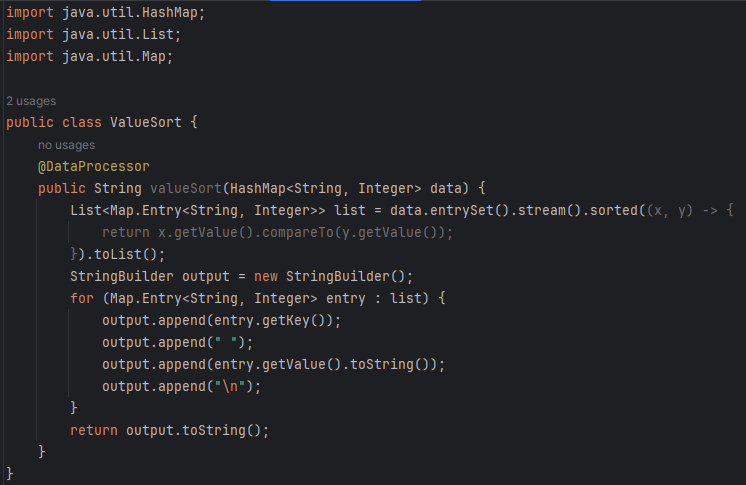


Рисунок 3 Класс ValueSort

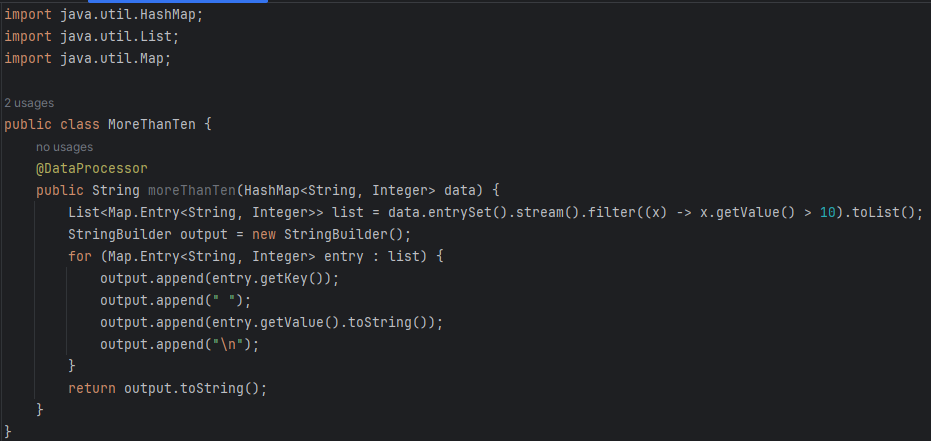


Рисунок 4 Класс MoreThanTen

Также был реализован класс DataManager у которого есть такие поля, как processors (список задач), service (многопоточный сервис), data (входные данные в виде HashMap), outputData.

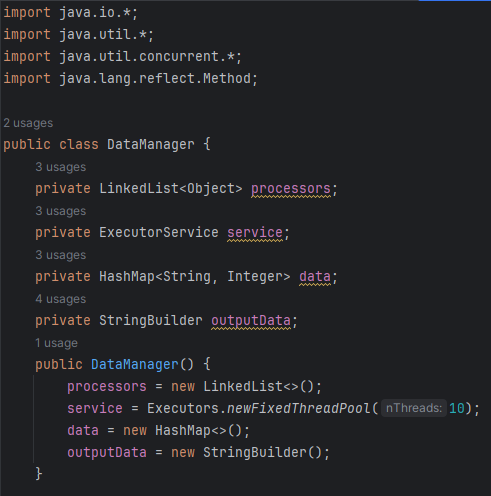


Рисунок 5 Поля и конструктор DataManager

Метод registerDataProcessor(Object processor) добавляет в DataManager новую задачу.

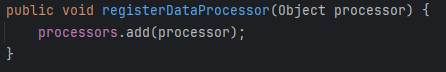


Рисунок 6 Метод registerDataProcessor

Метод loadData(String source) получает путь к исходному файлу и считывает его содержимое.

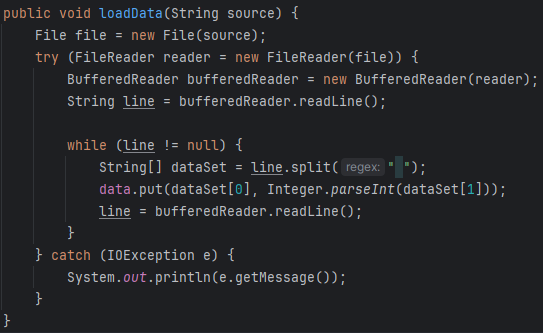


Рисунок 7 Метод loadData

Метод processData() выполняет все поставленные задачи, которые помечены аннотацией DataProcessor.



Рисунок 8 Метод processData

Метод saveData(String outputFile) выводит результат работы менеджера в отдельный файл.

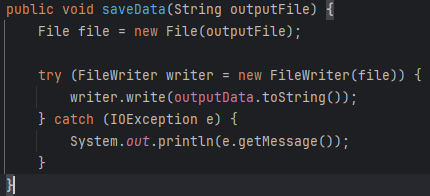


Рисунок 9 Метод saveData

Протестируем нашу программу с помощью кода, показанного на рисунке 10.



Рисунок 10 Тестирование программы

Получаем следующий результат.

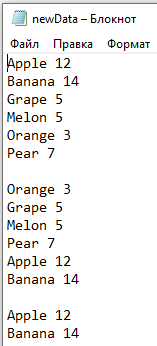


Рисунок 11 Релузьтат работы программы

# ВЫВОД

Данная лабораторная работа помогла лучше освоить язык Java. Была проделана работа с аннотациями, stream API. Также были закреплены знания о работе с потоками. Все знания были закреплены на практике и сохранены на GitHub: <https://github.com/HungryM1NT/ITIP>