**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 «Математической кибернетики и информационных технологий»

Информационные технологии и программирование

Лабораторная работа №7:

Выполнение заданий с аннотациями

Выполнила:

студентка группы БВТ2203

Залялиева Ильвина

Москва 2023

**Цель работы:** решение заданий.

**Задание:** Вам необходимо разработать приложение, которое считывает данные из исходного источника (например, файл, база данных или сетевой ресурс), применяет к ним различные операции с использованием Stream API, и сохраняет результаты в новый источник данных.

1. Создайте аннотацию @DataProcessor, которая будет использоваться для пометки методов обработки данных.
2. Создайте класс DataManager, который будет отвечать за многопоточную обработку данных. Этот класс должен иметь методы:

o registerDataProcessor(Object processor): Регистрирует объект-

обработчик данных с аннотацией @DataProcessor.  
o loadData(String source): Загружает данные из исходного

источника.  
o processData(): Запускает многопоточную обработку данных,

применяя методы с аннотацией @DataProcessor с

использованием Stream API.  
o saveData(String destination): Сохраняет обработанные данные в

новый источник.

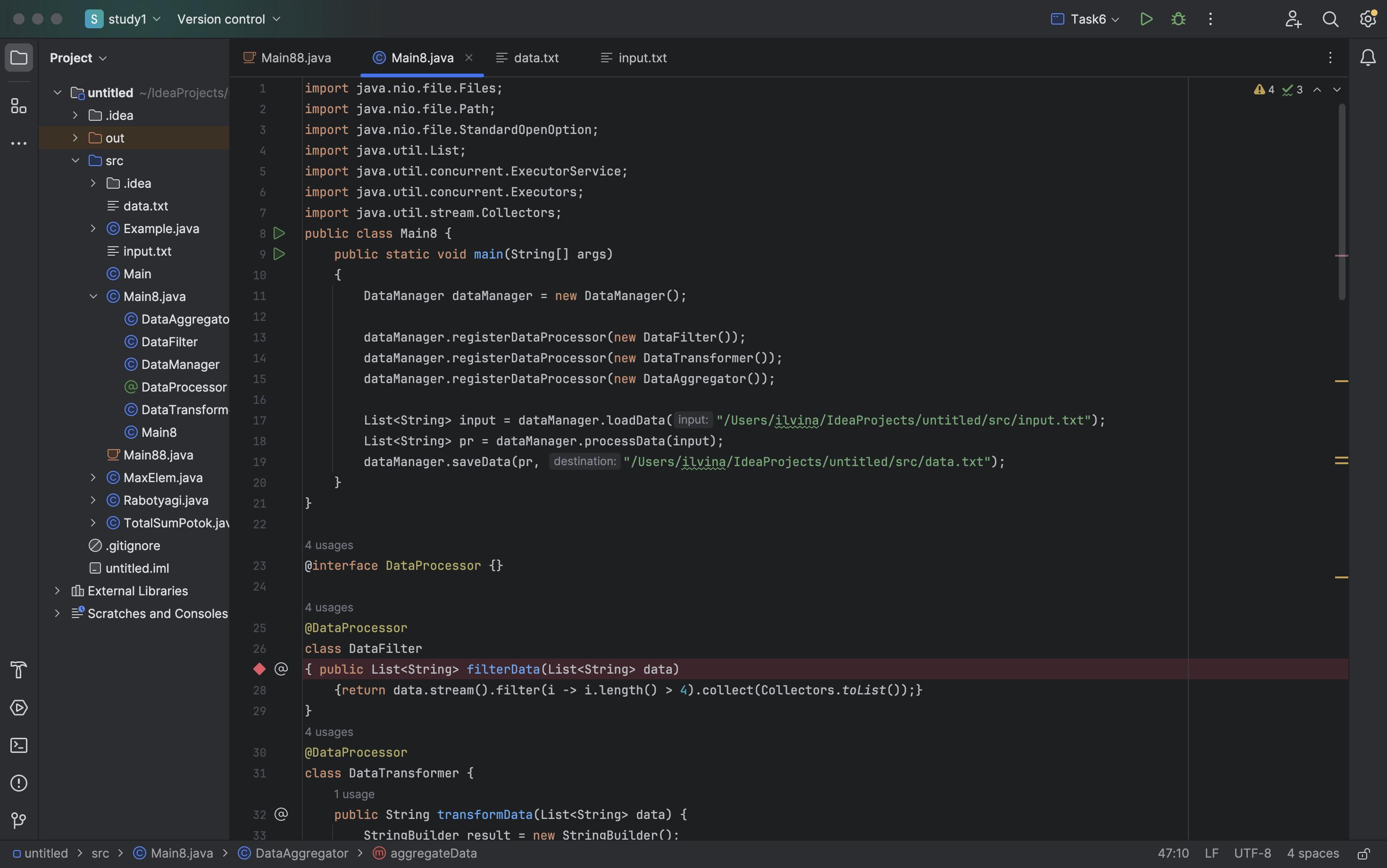
1. Создайте несколько классов, представляющих различные

обработчики данных, и пометьте их аннотацией @DataProcessor. Например, можно создать классы для фильтрации, трансформации и агрегации данных.

1. Используйте многопоточность из java.util.concurrent для эффективной обработки данных параллельно.
2. Протестируйте ваше приложение, загрузив данные из исходного источника, применив различные обработчики с помощью Stream API, и сохраните результаты в новый источник.

Ход работы

1. Импортирую библиотеки. Создаю основной класс, в котором будут применяться описанные выше задачи.



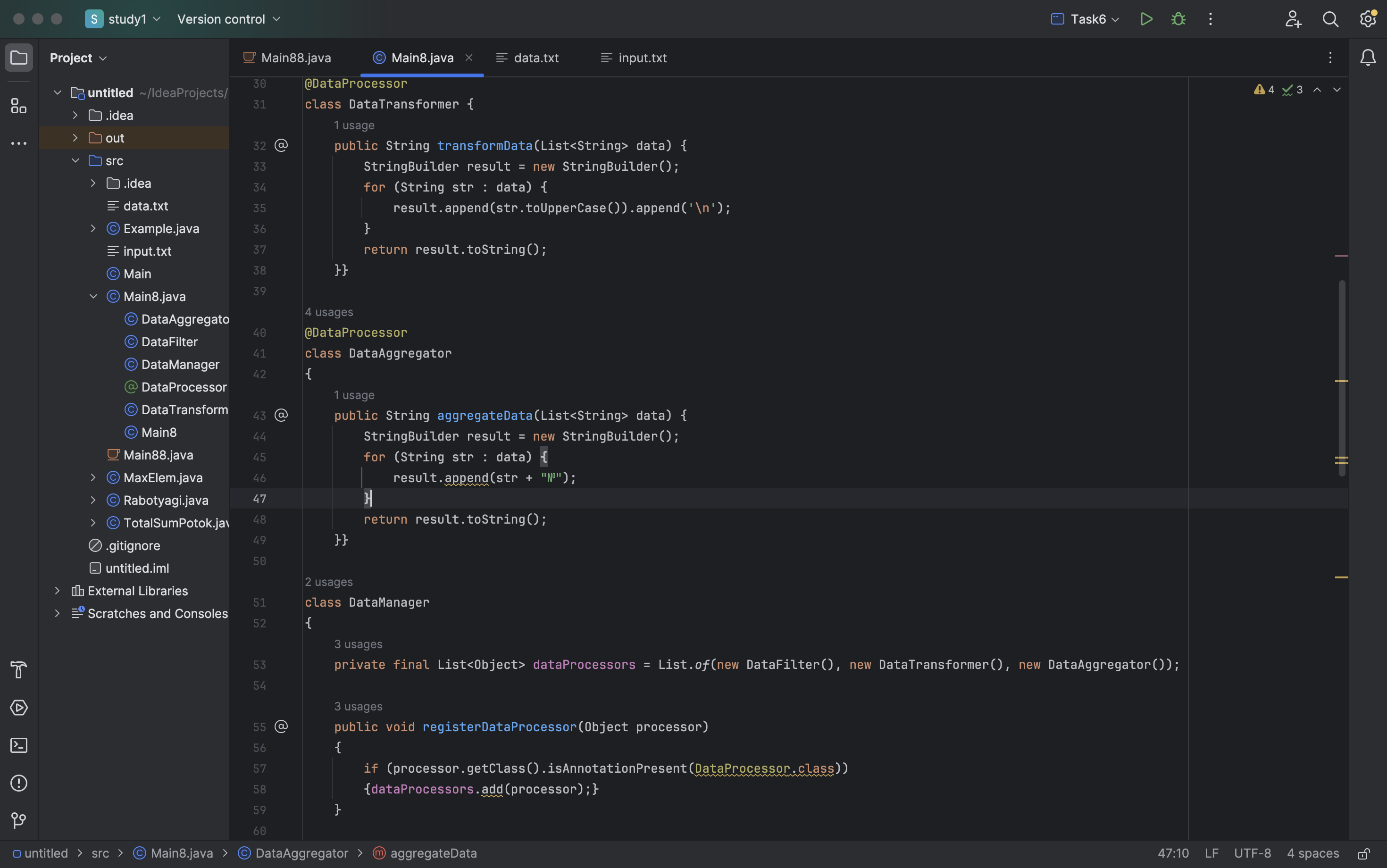
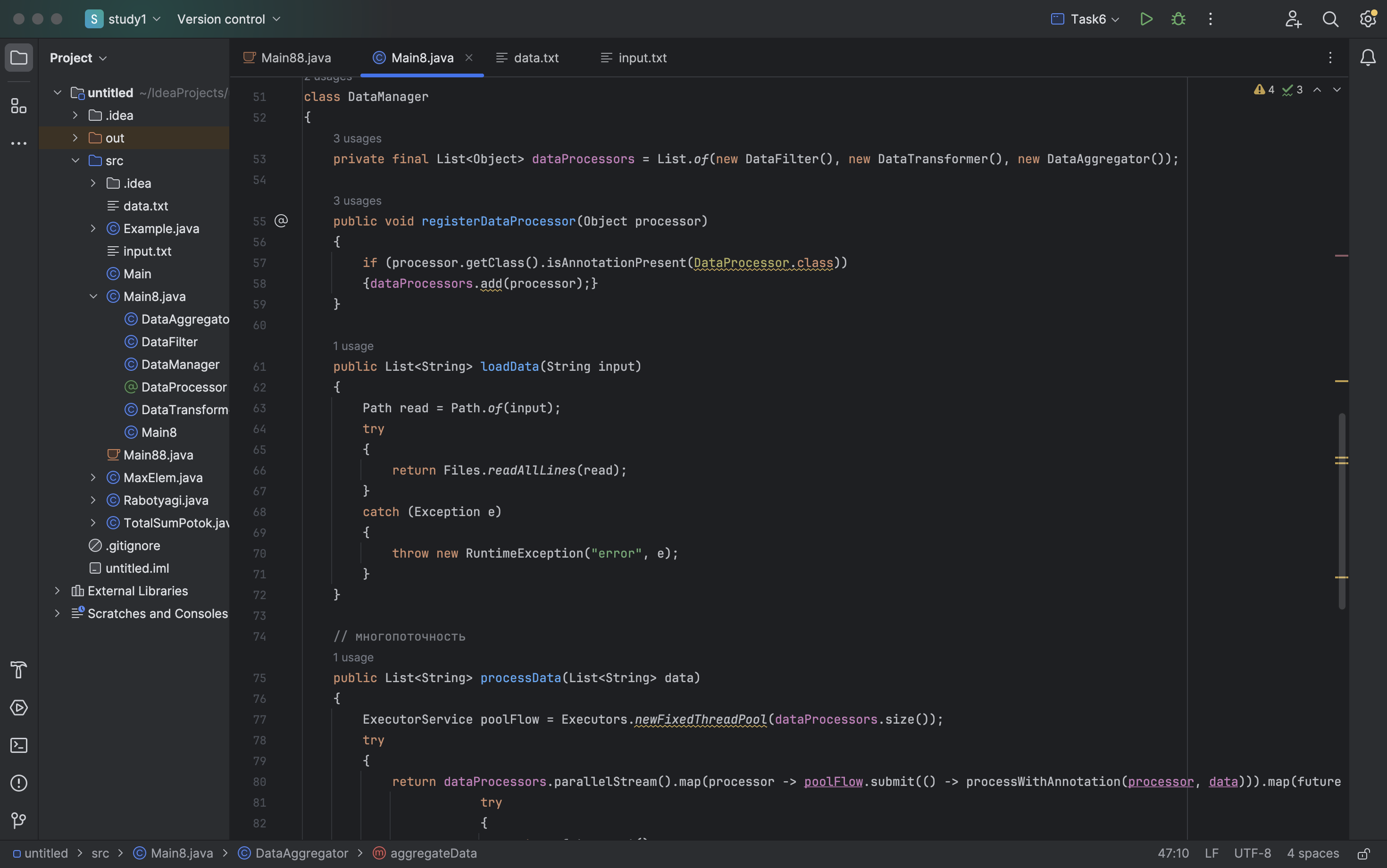
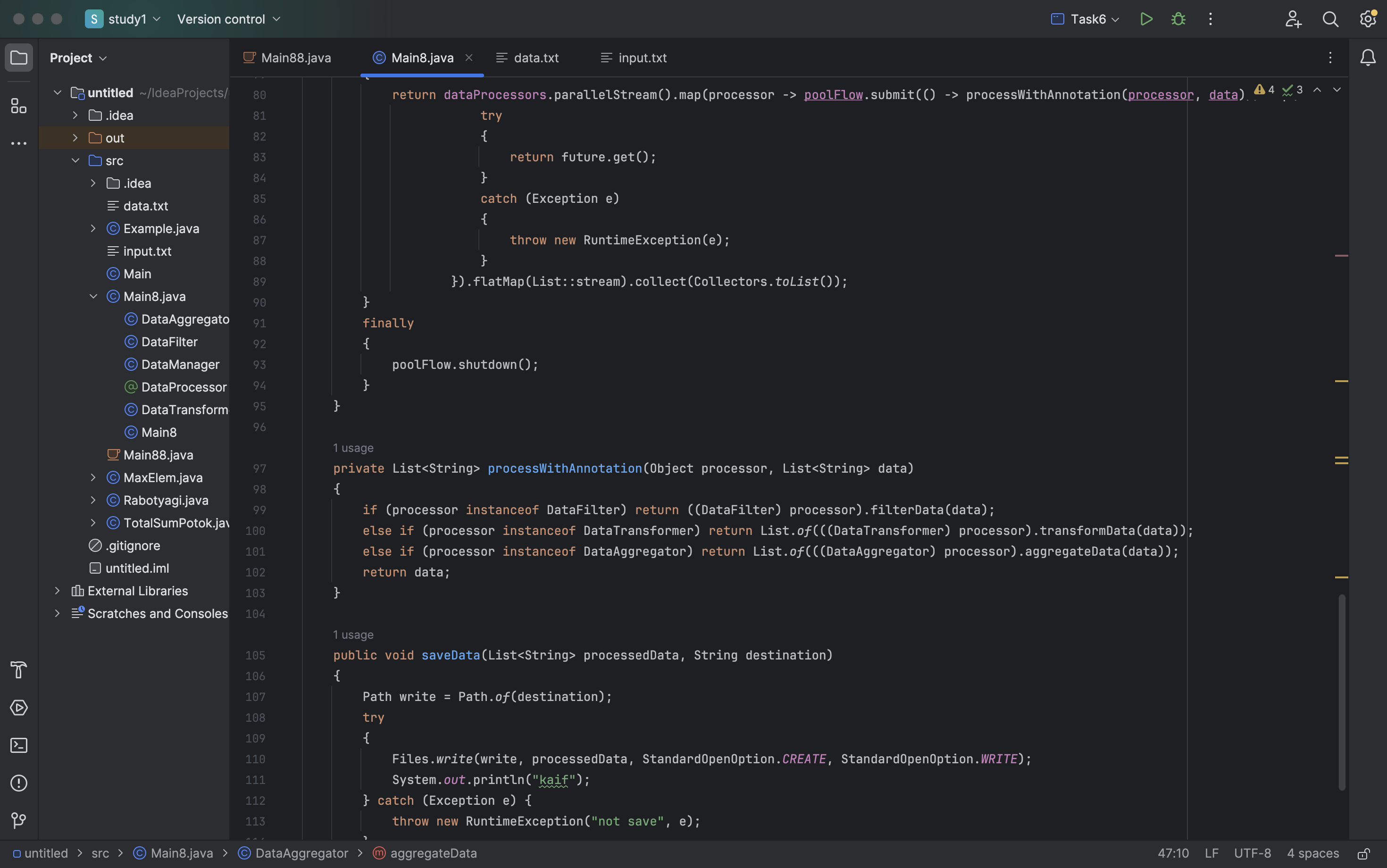


Рисунок 1, 2 – аннотации.

1. Создаю класс DataManager, который будет отвечать за многопоточную обработку данных.





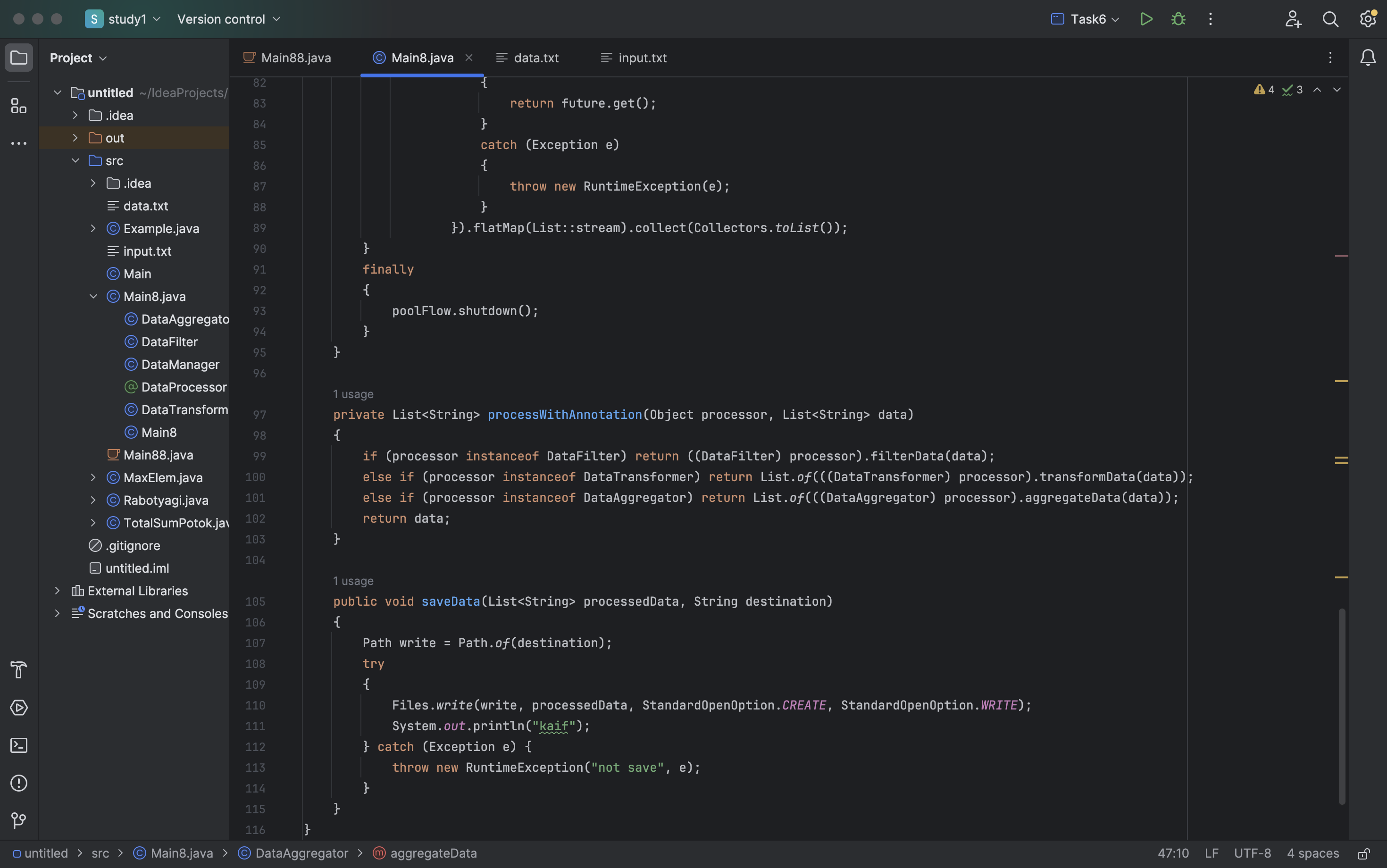
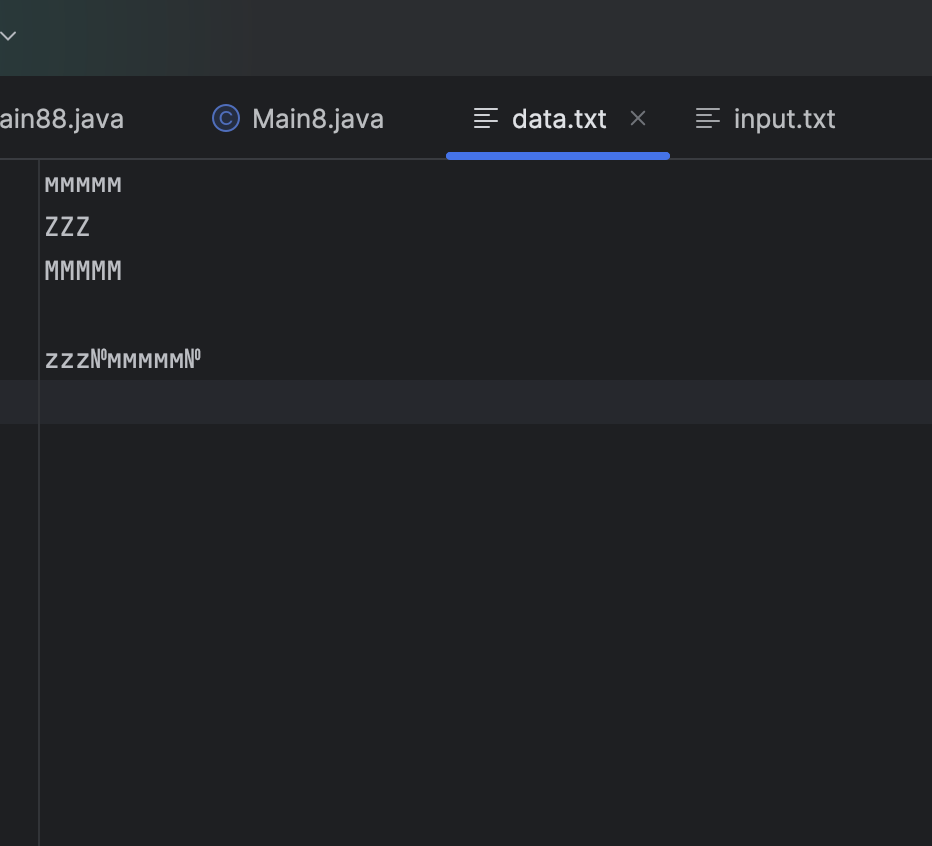


Рисунок 3, 4, 5 – оставшаяся часть кода.

1. Чтение и запись в файлы обработанного материала.



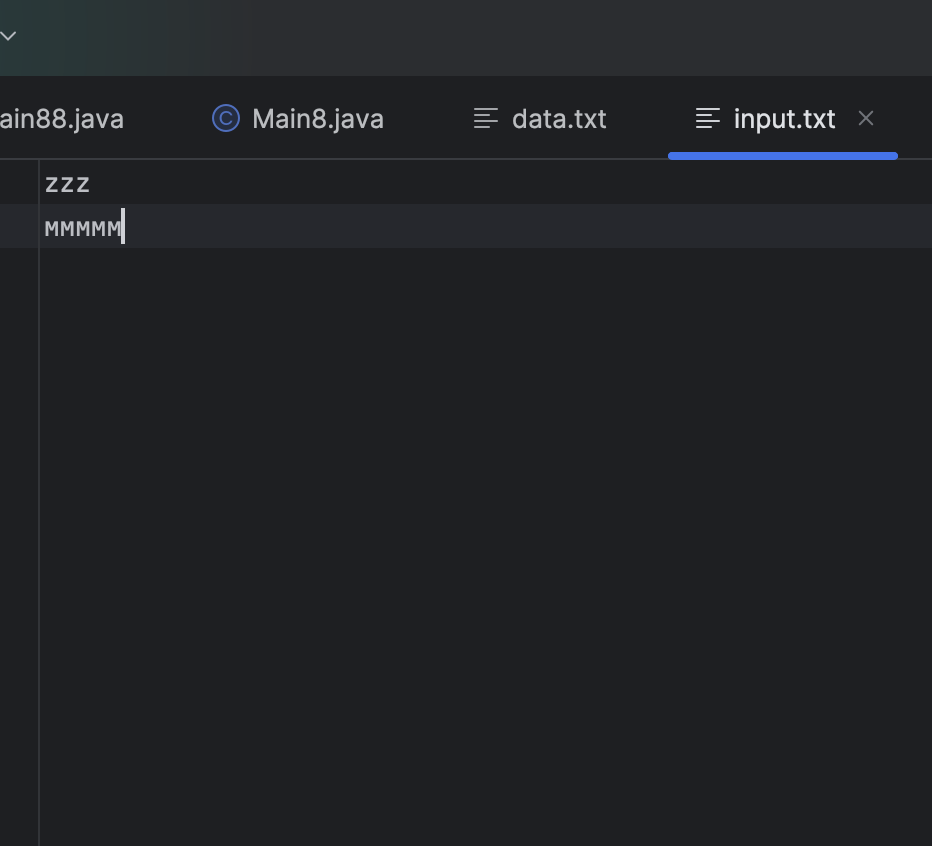


Рисунок 6, 7 – примеры тестирования.