

1. Особенности flatMap:

flatMap преобразует каждый элемент потока в поток других объектов. Затем он "сплющивает" все эти потоки в один поток.

Это полезно, когда каждый элемент входного потока может соответствовать нескольким элементам выходного потока.

Обычно используется для преобразования потока коллекций в поток элементов.

2. Виды Collectors в Stream API:

toList(): сбор данных в список.

toSet(): сбор данных в множество.

toMap(): сбор данных в карту.

joining(): соединение элементов в строку.

averagingInt(), averagingLong(), averagingDouble(): вычисление среднего значения.

summingInt(), summingLong(), summingDouble(): вычисление суммы.

maxBy(), minBy(): вычисление максимального и минимального значения с использованием компаратора.

groupingBy(): группировка элементов по категориям.

partitioningBy(): разбиение на две части по предикату.

и многие другие.

3. Для чего используется IntStream:

IntStream представляет поток примитивных int-значений, что может быть эффективнее, чем использование потока объектов Integer.

Он предоставляет множество полезных операций для работы с примитивными значениями, таких как sum(), average(), range(), и т. д.

4. Как можно пройтись в диапазоне от 1 до 100 не используя for:

Используя IntStream:

```
IntStream.rangeClosed(1, 100).forEach(System.out::println);
```

5. Из чего состоит Collector, когда реализовывают его:

Collector определяется четырьмя функциями:

supplier: возвращает новый мутабельный результат контейнера.

accumulator: инкорпорирует элемент типа T в мутабельный результат контейнера.

combiner: объединяет два значения в одно.

finisher: выполняет окончательное преобразование результата.

При создании пользовательского Collector нам необходимо предоставить реализацию для этих функций.