

Создание Stream

Способы создания Stream.

Метод empty()

Первый способ — использовать метод `empty()`. Этот метод возвращает пустой стрим без элементов. Его удобно использовать, чтобы не возвращать `null` из метода.

```
Stream<String> streamEmpty = Stream.empty();
```

Также можно создать стрим на основе перечисления:

```
Stream<String> streamFromArray = Stream.of("a", "b", "c");
```

В параметры метода `of` можно передавать сколько угодно аргументов, и вы получите из них стрим.

Способы создания Stream.

Метод stream класса Collection

Если уже есть существующая коллекция, то можно воспользоваться методом stream класса Collection.

```
List<Double> doubleList = List.of(1D, 2D, 3D);  
Stream<Double> stream = doubleList.stream();
```

Способы создания Stream.

Статический метод Arrays.stream()

Похожий механизм есть и для массивов. Это статический метод `Arrays.stream()`. С помощью него можно получать стримы из примитивов — классы `IntStream`, `LongStream`, `DoubleStream`.

```
double[] dArray = new double[]{3.4, 5.6};  
DoubleStream stream = Arrays.stream(dArray);
```

Способы создания Stream.

Создание «бесконечного» стрима

Также вы можете создать «бесконечный» стрим. Для этого нужно задать способ генерации следующего значения, и Stream API создаст значения за нас:

```
Stream<Double> stream = Stream.generate(() -> Math.random());
```

Если попробовать вывести все значения такого стрима, то вы получите аналог бесконечного цикла:

```
Stream.generate(Math::random).forEach(System.out::println);
```

Поэтому лучше ограничить размер каким-то разумным значением:

```
Stream<Double> stream = Stream.generate(Math::random).limit(10);
```

Написание аналога обычного цикла в Java

Также можно написать аналог обычного цикла в Java. Предположите что вам нужно создать цикл со счётчиком от 0 до 9. Это легко сделать с помощью оператора for:

```
for (int i = 0; i < 10; i++) {  
    //здесь находится наша логика  
}
```

Ещё проще это сделать с помощью метода range класса IntStream:

```
IntStream.range(0, 10) .....
```

Вывод

Таким образом, способы создания стрима очень разнообразны и полностью отвечают его философии — **операции над наборами данных**.

Стрим можно создать, используя различные источники данных, — массив, коллекцию, обычную группу объектов, а также сгенерировать программно.