

Терминальные операторы

Терминальные операторы

Терминальные операторы преобразуют поток элементов в нужный вам окончательный вид.

С одним из терминальных операторов вы уже знакомы, это **forEach**.

Он применяет Consumer к каждому элементу стрима.

Терминальные операторы

Среди терминальных операций есть операции по нахождению наименьшего и наибольшего значения.

Это методы **max()** и **min()**.

Как и в случае с методом **sorted()**, необходимо передать в параметры метода компаратор, который покажет, по какому признаку следует сравнивать объекты.

Терминальные операторы

Наиболее часто стрим используют, чтобы взять одну коллекцию, что-то сделать с её элементами, и сложить их в другую коллекцию.

Для этого используется метод **collect()**.

В качестве аргумента он принимает объект класса `Collector`, который и будет определять, каким образом вы хотите собрать ваши элементы стрима в коллекцию.

Терминальные операторы

С помощью терминальных операторов можно не только получить какой-то элемент или набор элементов, но и проверить соответствие элементов какому-либо предикату.

Для этого используются методы **anyMatch()**, **noneMatch()** и **allMatch()**.

Вывод

Как видите, Stream API в Java имеет множество полезных инструментов для манипуляции коллекциями и другими источниками данных. С помощью Stream API можно переписать любой код в функциональном стиле, но важно не злоупотреблять этим.