

Java Professional

module #3 Lambdas & Stream API

lecture # 3. Stream API Stream, primitive streams.

Intermediate and Terminal operations

Mentor: Rustam Khakov

Stream, primitive streams. Intermediate and Terminal operations

- Introduction
- Intermediate operations
- Terminal operations
- Important points

What is Java Stream API

Представленный в Java 8, новый инструмент языка Java:

- Позволяет использовать функциональный стиль с коллекциями.
- Последовательность методов для получения желаемого результата.
- Не является структурой данных, принимает входные данные из коллекций, массивов, ввода-вывода.
- Не изменяет исходную структуру данных

Imperative

Explicit Instructions

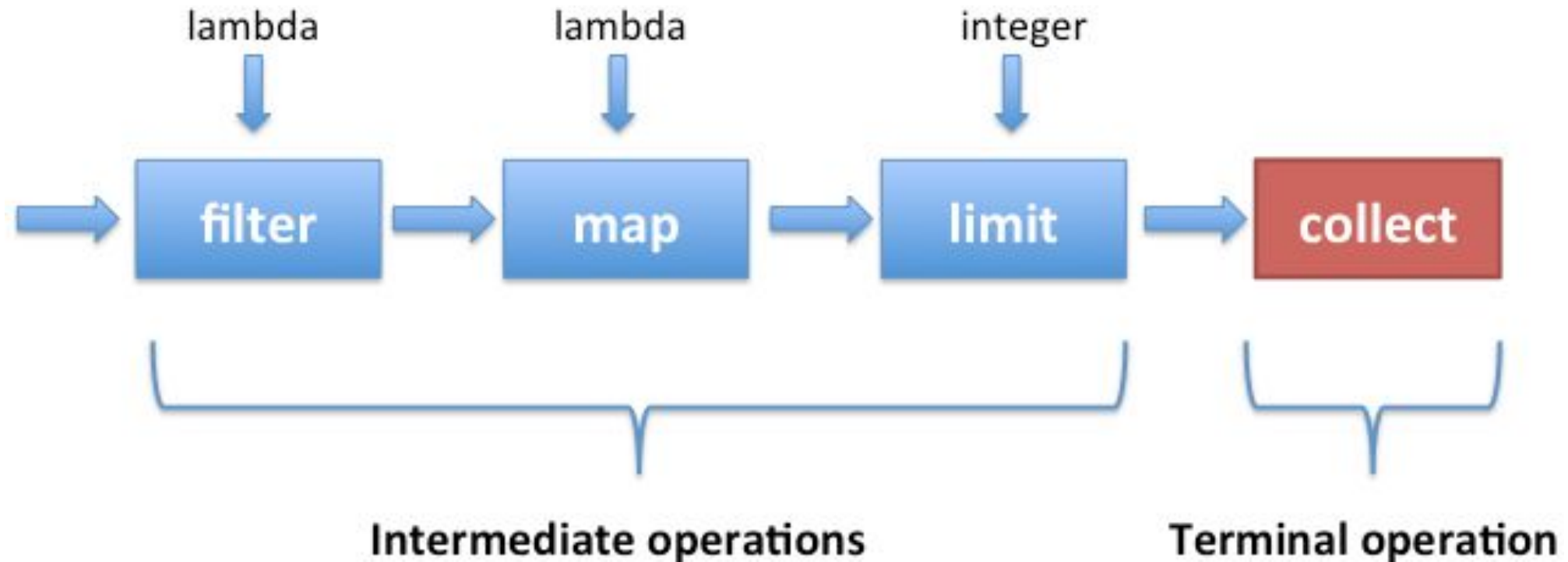
The system is stupid,
you are smart

Declarative

Describe the Outcome

The system is smart,
you don't care

Intermediate and Terminal Operations



- Пока не вызван терминальный метод, ничего не происходит.
- Конвейерные методы являются «ленивыми».

Java Streams

map

filter

reduce

collect



Stream Operations

```
graph TD; SO([Stream Operations]) --- IO[Intermediate Ops]; SO --- TO[Terminal Ops]; SO --- SC[Short-circuiting];
```

Intermediate Ops

filter()
map()
peek()
boxed()

Terminal Ops

collect()
reduce()
max()
min()

Short-circuiting

limit()
findFirst()
findAny()

Operation	Return Type	Type Of Operation	What It Does?
filter()	Stream<T>	Intermediate	Returns a stream of elements which satisfy the given predicate.
map()	Stream<R>	Intermediate	Returns a stream consisting of results after applying given function to elements of the stream.
distinct()	Stream<T>	Intermediate	Returns a stream of unique elements.
sorted()	Stream<T>	Intermediate	Returns a stream consisting of elements sorted according to natural order.
limit()	Stream<T>	Intermediate	Returns a stream containing first <i>n</i> elements.
skip()	Stream<T>	Intermediate	Returns a stream after skipping first <i>n</i> elements.
forEach()	void	Terminal	Performs an action on all elements of a stream.
toArray()	Object[]	Terminal	Returns an array containing elements of a stream.
reduce()	type T	Terminal	Performs reduction operation on elements of a stream using initial value and binary operation.
collect()	Container of type R	Terminal	Returns mutable result container such as List or Set.
min()	Optional<T>	Terminal	Returns minimum element in a stream wrapped in an Optional object.
max()	Optional<T>	Terminal	Returns maximum element in a stream wrapped in an Optional object.
count()	long	Terminal	Returns the number of elements in a stream.
anyMatch()	boolean	Terminal	Returns true if any one element of a stream matches with given predicate.
allMatch()	boolean	Terminal	Returns true if all the elements of a stream matches with given predicate.
noneMatch()	boolean	Terminal	Returns true only if all the elements of a stream doesn't match with given predicate.
findFirst()	Optional<T>	Terminal	Returns first element of a stream wrapped in an Optional object.
findAny()	Optional<T>	Terminal	Randomly returns any one element in a stream.

Важное

1. «Поток» состоит из источника, за которым следуют промежуточные методы, (промежуточных методов может быть много или не быть совсем)
2. Не изменяет оригинальную коллекцию
3. Не должен иметь внутреннего состояния - точно так же как и функции
4. Могут быть последовательными и параллельными.
5. Методы не должны быть null
6. Нельзя переиспользовать стрим - вызывать можно только один раз