

Stream API



1. Stream 의 특징

- 주의 : 우리가 배울 stream 은 java.io 의 Stream 과 다른 내용 이다.
- 주의 : Java8 부터 지원하는 기능이다.
- Stream 을 통해 Array 나 Collection Framework 의 데이터를 쉽고 빠르게 처리 할 수 있다.

특징	설명
원본 데이터를 변경하지 않는다.	Stream 은 원본의 훼손 위험이 없다.
일회용 이다.	Stream 사용 후 자동 반납되므로 은 재사용이 불가능 하다.
내부 반복으로 작업을 처리 한다.	Stream 에서는 별도의 반복 문 이 불필요 하다.
병렬 처리가 가능하다.	Stream 에서는 병렬처리로 복잡한 연산을 빠르게 처리 할 수 있다.

2. Stream 생성

- Stream 생성 방법은 다양 하다.
- 대표적으로 Collection 과 Array 를 통해 생성 한다.

방법	설명
List.stream();	Collection 을 stream 화
Arrays.stream(Array);	Array 를 stream 화
Stream.of(Array);	Array 를 stream 화
Stream.of(value, value,...);	값 들을 stream 화
IntStream.range(0,10);	순수 타입을 Stream 화

3. Stream 가공

- 생성된 Stream 은 일련의 가공과정을 거쳐 새로운 데이터를 만들어 낸다.

기능	설명
필터링(filter)	특정 조건의 데이터만 추출 할 수 있다.
데이터 변환(map)	들어오는 데이터를 변환하여 반환 할 수 있다.
정렬(sort)	정렬을 할 수 있다.
중복 제거(distinct)	중복을 제거 할 수 있다.
특정 연산 수행(peek)	데이터에 영향을 주지 않는 수행(데이터 확인 같은...) 을 할 수 있다.
원시스트림<-> 스트림화(mapTo)	원시스트림과 스트림간의 변경이 가능 하다.

4. Stream 결과 만들기

- Stream 가공을 통해 만들어진 결과에 대해 최종 데이터를 구해 낼 수 있다.

기능	설명
집계 기능	max, min, sum, average, count
데이터 수집 기능	데이터를 Collection Framework 형태로 변경 할 수 있다.
조건 검사	해당 데이터가 조건을 만족하는지 확인 할 수 있다.
개별 연산 수행	forEach 를 통해 개별 데이터를 꺼내 연산을 진행 할 수 있다.

5. FlatMap

- FlatMap 은 다중 구조로 되어있는 Map 을 단 차원으로 평평(flat)하게 만드는 것이다.

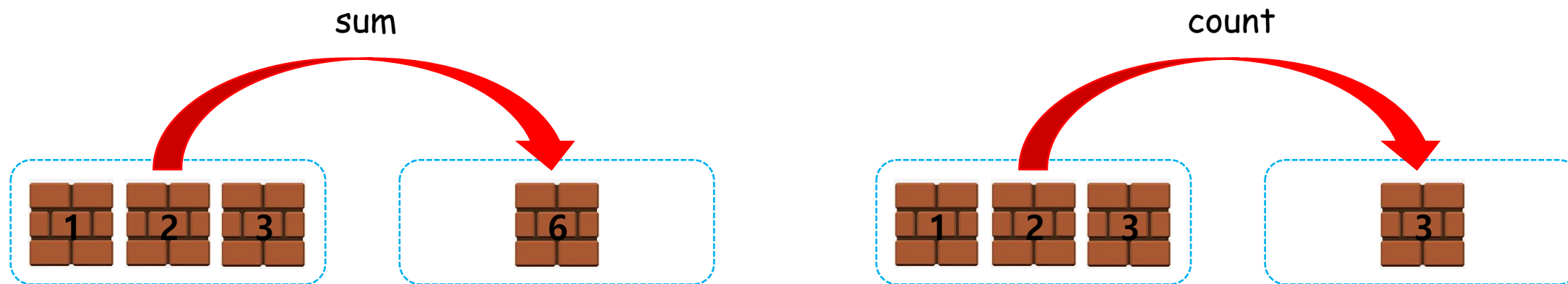
```
[["H","E","L","L","O"], ["W","O","R","L","D"]]
```



```
["H","E","L","L","O", "W","O","R","L","D"]
```

6. Reduce

- Reduce 는 누산기와 연산으로 컬렉션 내의 값을 더 작은 컬렉션 이나 단일 값으로 만든다.



`reduce(accumulator, identity)`

accumulator : 받아온 인자 값을 연산하여 반환

identity : 계산 처리시 사용할 초기값

7. Null Safe

- Java 에서 개발 시 많이 나타나는 Exception 은 Null Point Exceptoin 일 것이다.
- Java8 부터 지원하는 Optional 은 Null 처리를 가독 성 있게 해주었고
- Stream 에서도 이 기능을 사용 하고 있다.

기능	설명
Optional.ofNullable()	Null 일수도 있는 값을 넣어 Optional 객체를 생성
Optional.ifPresent()	Optional 의 값이 Null 이 아닐 경우 실행
Optional.orElse()	Optional 의 값이 Null 일 경우 기본값을 반환
Optional.orElseGet()	Optional 의 값이 Null 일 경우 특정 기능을 수행

8. 실행 순서와 병렬 처리

- Stream 은 수평적 처리가 아닌 수직적 처리 구조를 가지고 있다.
- 이로서 얻는 장점과 단점을 생각해 보자

