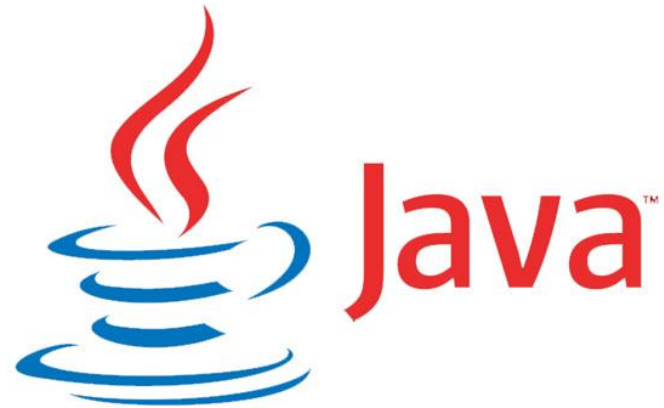


제11장 람다식과 스트림

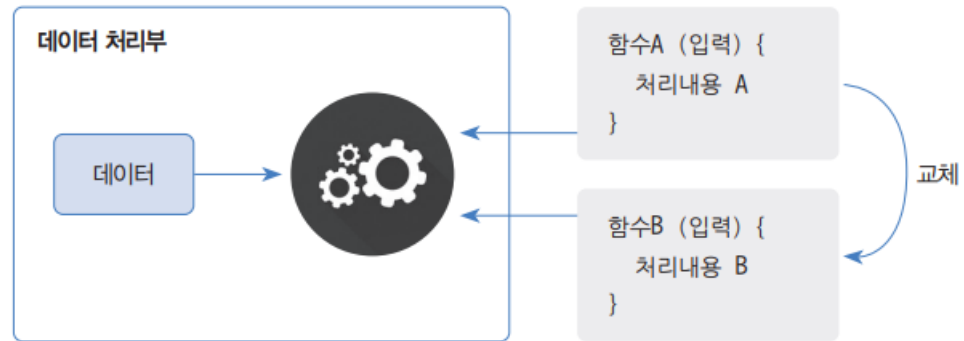


목차

1. 람다식
2. 스트림
3. 내부 반복자
4. 스트림 필터
5. 스트림 매핑

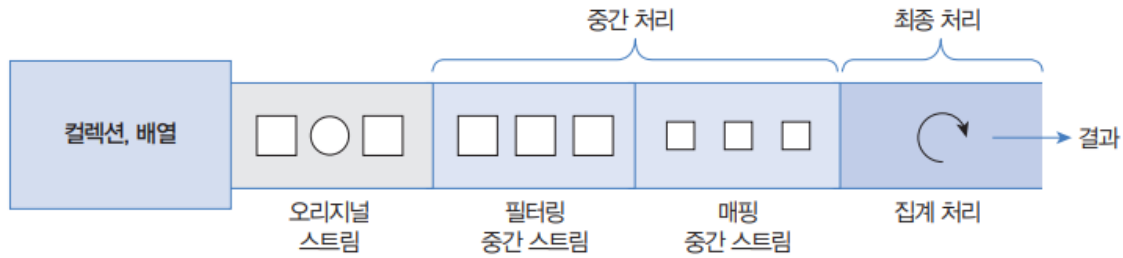
1. 람다식

- 람다식^{Lambda}은 객체지향 프로그래밍에 함수형 프로그래밍을 지원하기 위한 구문 형식
- 함수형 프로그래밍은 함수를 정의하고 이 함수를 데이터 처리부로 보내 데이터를 처리하는 프로그래밍 기법
- 자바는 인터페이스 익명 객체를 람다식으로 구현



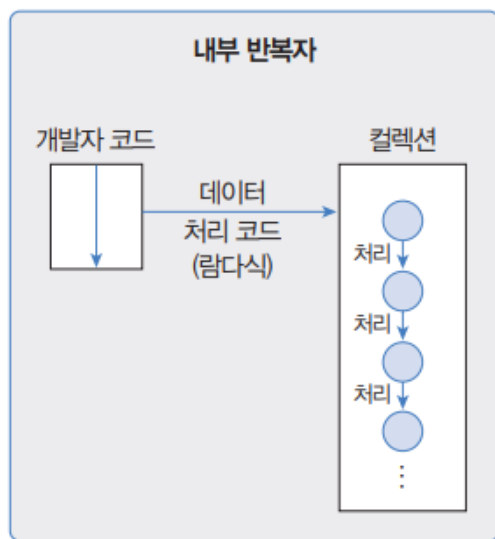
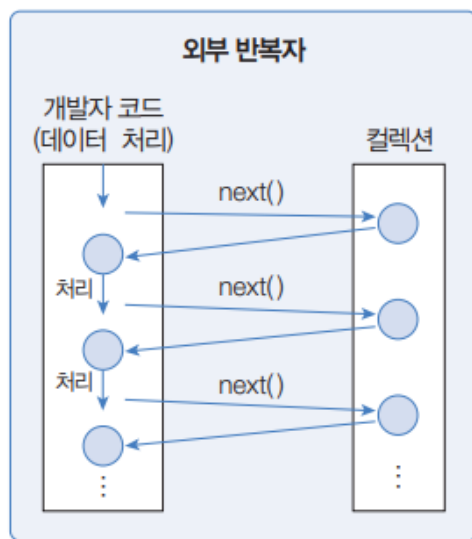
2. 스트림

- 스트림은 배열이나 컬렉션 원소를 람다식으로 처리할 수 있는 반복자를 제공
- 스트림을 사용하면 병렬처리가 가능하기 때문에 대용량의 데이터 처리에 더 나은 성능을 제공
- 여러 종류의 스트림을 연결(파이프라인)해서 원하는 작업을 수행



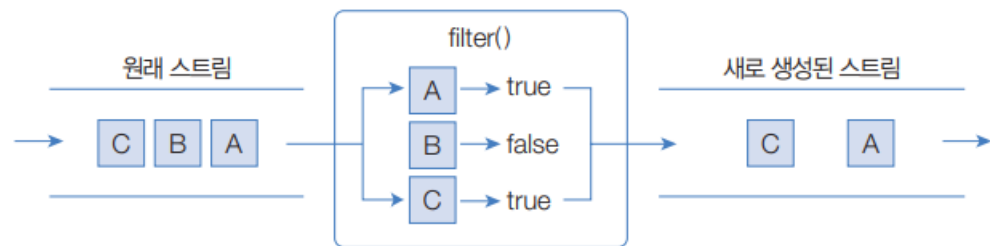
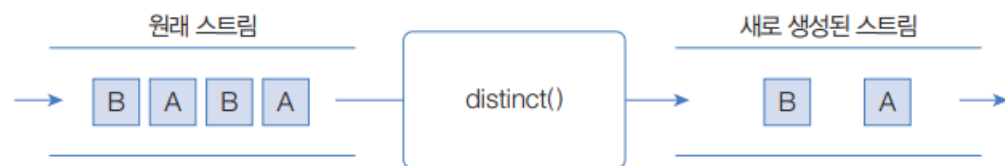
3. 내부 반복자

- 요소 처리 방법을 컬렉션 내부로 주입시켜서 요소를 반복 처리
- 람다식을 가지고 컬렉션 내부에서 요소를 반복 처리
- 내부 반복자는 더 나은 성능을 위해 작업 스레드로 병렬 작업 수행 가능



4. 스트림 필터

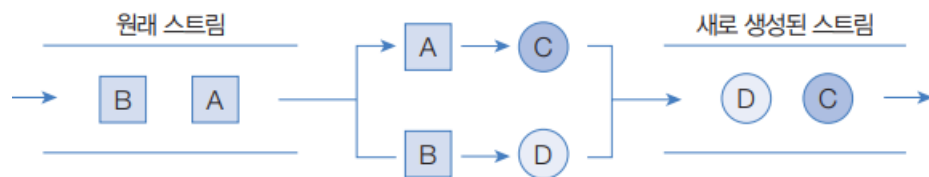
- 스트림 필터는 요소를 걸러내는 중간 처리 기능
- `distinct()`는 요소의 중복을 제거
- `filter()`는 매개값으로 주어진 Predicate가 `true`를 리턴하는 요소만 필터링



5. 스트림 매핑

- 스트림의 요소를 다른 요소로 변환하는 중간 처리 기능
- 매핑 메서드 `mapXxx()`, `flatMapXxx()` 등 제공

`mapXxx()` : 요소를 다른 요소로 변환



`flatMapXxx()` : 요소를 여러 개의 요소로 변환

