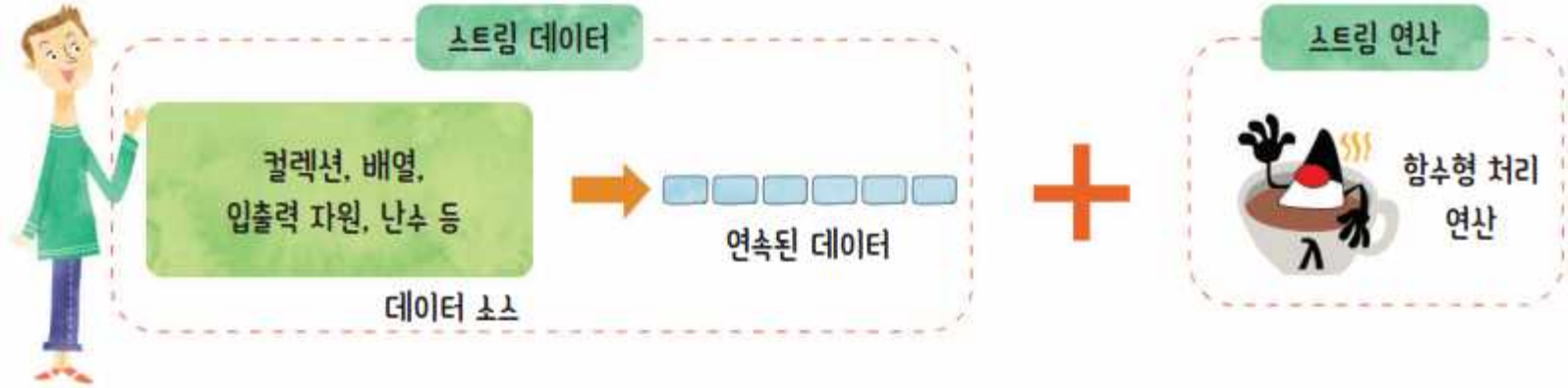
12장 스트림

목차

스트림 소개

■ 스트림 의미

- JDK 8부터 새롭게 추가된 기능으로 데이터 집합체를 반복적으로 처리
- 스트림을 이용하면 다수의 스레드 코드를 구현하지 않아도 데이터를 병렬로 처리
- 스트림은 스트림 데이터와 스트림 연산의 개념을 모두 포함

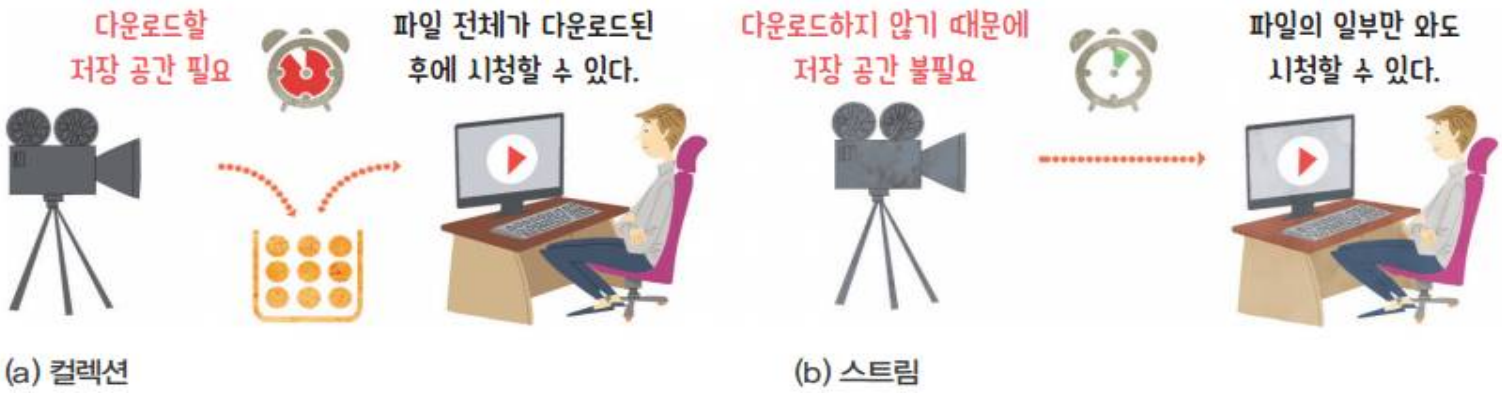


스트림 소개

■ 컬렉션과 스트림

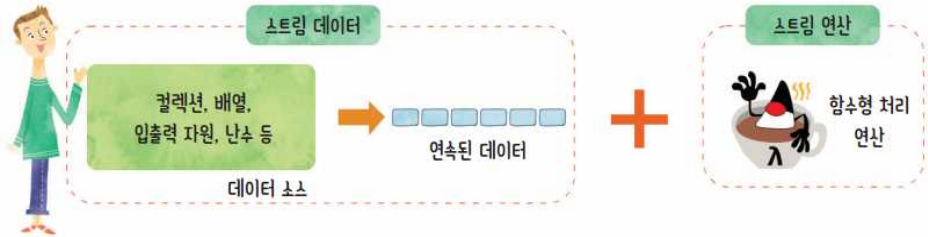
구분	컬렉션	스트림
처리 방식	다운로드	스트리밍
저장 공간	필요	불필요
반복 방식	외부 반복	내부 반복
코드 구현	명령형	선언형
원본 데이터	변경	변경하지 않고 소비
연산 병렬화	어려움	쉬움

● 데이터 처리 방식



(a) 컬렉션 (b) 스트림

스트림 소개



컬렉션과 스트림

- 컬렉션이 데이터의 공간적 집합체라면, 스트림은 데이터의 시간적 집합체이다.
- 컬렉션은 데이터 원소의 효율적인 관리와 접근에 맞게 설계되어 있지만, 스트림은 데이터 원소에서 수행할 함수형 연산에 맞게 설계되어 있다.
- 따라서 스트림은 원소에 직접 접근하거나 조작하는 수단을 제공하지 않는다.
- 스트림을 사용하면 코드가 간단해지고 오류 발생 확률이 줄어든다.

- 예제 : sec01/StreamDemo

⇒0~29 사이의 난수 20개 중 10보다 큰 난수를 찾아 정렬한 후 출력

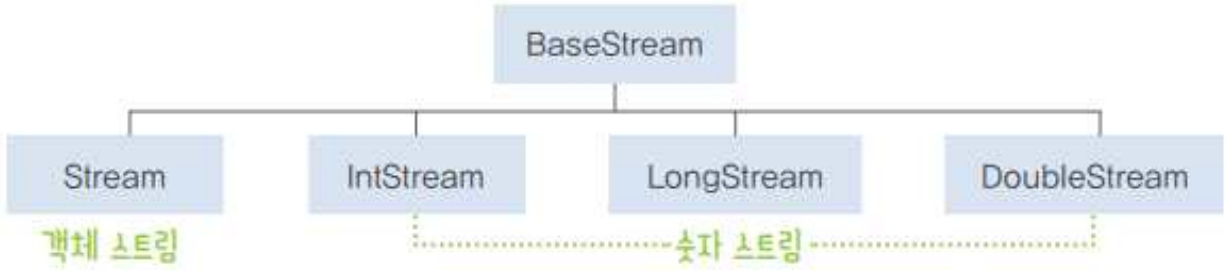
```
[19, 20, 23, 24, 27, 28]
19 20 23 24 27 28
```

```
package sec01;
import java.util.ArrayList;
import java.util.Collections;
import java.util.List;
import java.util.Random;

public class StreamDemo {
    public static void main(String[] args) {
        List<Integer> list = new ArrayList<>();
        Random r = new Random();

        for (int i = 0; i < 10; i++) //난수 생성 list에 추가
            list.add(r.nextInt(30));

//컬렉션으로 처리
        List<Integer> gt10 = new ArrayList<>();
        for (int i : list)
            if (i > 10)
                gt10.add(i);

        Collections.sort(gt10);
        System.out.println(gt10);

//스트림으로 처리 람다식으로 표현
        list.stream().filter(i -> i > 10).sorted()
                .forEach(x -> System.out.print(x + " "));
    }
// 컬렉션에서 스트림 생성, 10보다 큰 원소 추출, 정렬, 출력
}
```

스트림 종류와 생성

스트림 종류

- lava.base모듈에 포함된 java.util.stream 패키지의 인터페이스이며 다음 과같은 인테페이스로 구성



숫자 스트림과 객체 스트림의 차이

- 숫자 스트림은 평균, 합계를 반환하는 average( ), sum( )이라는 메서드가 있다.
- 객체 스트림이 제공하는 최종 연산이 Optional 타입을 반환한다면 숫자 스트림은 OptionalInt, OptionalLong, OptionalDouble 타입을 반환한다.
- 숫자 스트림은 데이터 스트림의 기본 통계 요약 내용을 나타내는 summaryStatistics() 메서드를 제공한다.

- IntStream과 Stream<Integer> ?
  IntStream은 int 타입 스트림이며, Stream<Integer>는 Integer타입의 객체 스트림이다.
  IntStream은 Stream<Integer>로 대체할 수 있지만 박싱 및 언박싱과 같은 부담으로 인하여 성능이 떨어지며 average(), sum()과 같은 메서드를 사용할 수 없다.

스트림 종류와 생성

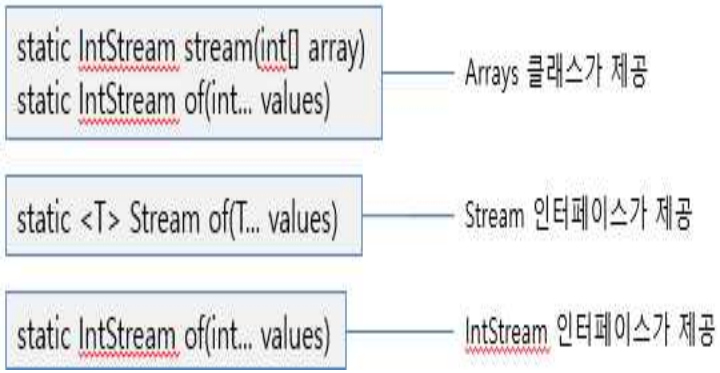
■ 스트림 생성

- 스트림은 주로 데이터 소스가 될 수 있는 컬렉션, 배열, 입출력 채널을 이용하여 생성할 수 있다. 그 외에도 Random클래스가 제공하는 메서드나 스트림의 정적 메서드를 통해서도 생성 할 수 있다.

- 컬렉션으로부터 스트림 생성(방법1)

```
default Stream<E> stream()
default Stream<E> parallelStream()
```

- => 예제 12-1
- 배열로부터 스트림 생성(방법2)



- 예제 : sec02/Array2StreamDemo

```
package sec02;

import java.util.Arrays;
import java.util.stream.DoubleStream;
import java.util.stream.IntStream;
import java.util.stream.Stream;

public class Array2StreamDemo {
    public static void main(String[] args) {
        int[] ia = {2, 3, 5, 7, 11, 13};
        IntStream is = Arrays.stream(ia);

        String[] strings = {"The", "pen", "is", "mighter",
"than", "the", "sword"};
        Stream<String> ss =
                Stream.of(strings);

        double[] da = {1.2, 3.14, 5.8, 0.2};
        DoubleStream ds =
                DoubleStream.of(da);
    }
}
```

스트림 종류와 생성

스트림 생성

- 기타 데이터로부터 스트림 생성
  - Random의 ints( ), longs( ), doubles( )
  - 숫자 스트림과 객체 스트림의 iterate( )와 generate( )
  - IntStream과 LongStream의 range( ) 혹은 rangeClosed( )
  - 입출력 파일이나 폴더로부터도 스트림을 생성
  - ...

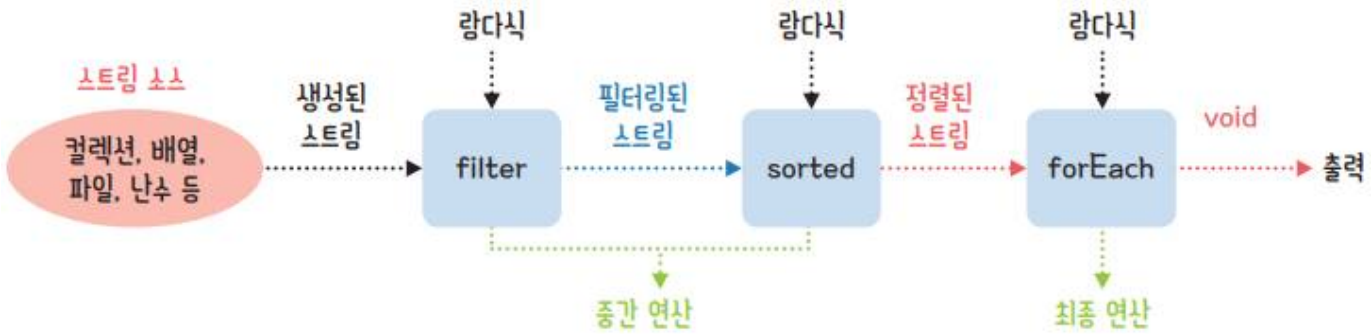
- 예제 : sec02/Etc2StreamDemo

```
package sec02;

import java.util.Random;
import java.util.stream.IntStream;
import java.util.stream.Stream;

public class Etc2StreamDemo {
    public static void main(String[] args) {
        IntStream is1 = IntStream.iterate(1, x -> x + 2);

        IntStream is2 = new Random().ints(0, 10);

        Stream<Double> ds =
Stream.generate(Math::random);

        IntStream is3 = IntStream.range(1, 5);
    }
}
```

스트림 연산과 옵션 타입

■ 스트림 파이프라인

- 스트림 연산의 결과가 Stream 타입이면 연속적으로 호출할 수 있다. 스트림 연산의 연속 호출은 여러 개의 스트림이 연결되어 스트림 파이프라인을 형성

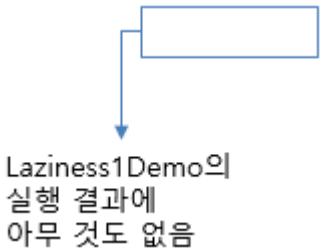


■ 느긋한 연산과 조급한 연산

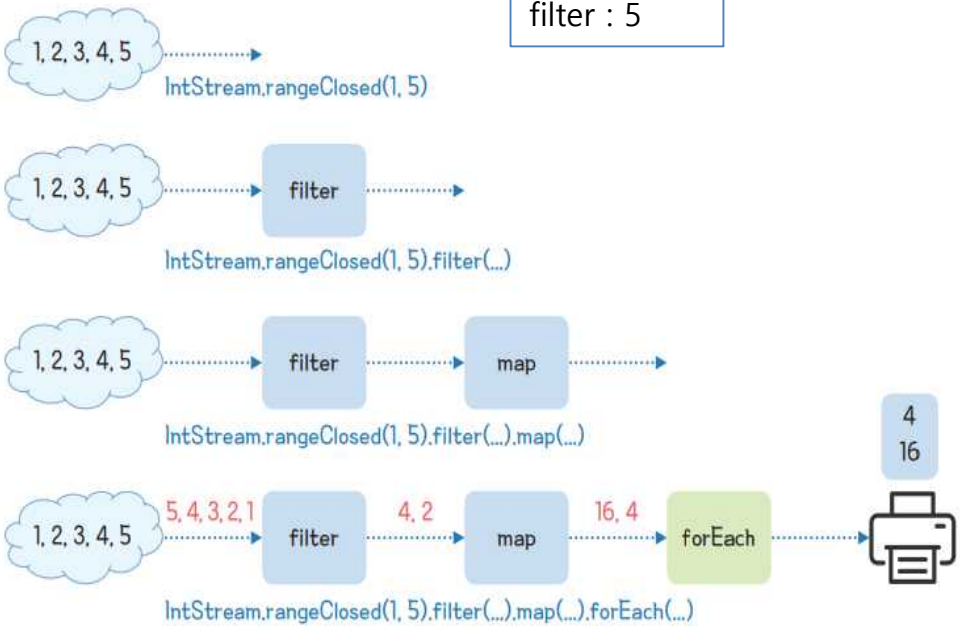
- 느긋한 연산은 조급한 연산이 데이터 소스에게 원소를 요구할 때까지 아무 연산도 수행하지 않고 기다리는 특징
- 스트림의 최종 연산은 조급한 연산이지만 중간 연산은 느긋한 연산이다.
- 최종 연산이 호출되기 전까지 중간 연산은 아무런 작업을 수행하지 않는다.
- 스트림의 중간 연산이 느긋한 연산이기 때문에 다운로드 방식처럼 저장 공간이 따로 필요 없다. 따라서 스트림 연산은 빅데이터뿐만 아니라 무한 스트림에도 대응할 수 있다.

스트림 연산과 옵션 타입

- 느긋한 연산과 조급한 연산
  - 예제 : sec03/Laziness1Demo, sec03/Laziness2Demo




```
filter : 1
filter : 2
map : 2
forEach : 4
filter : 3
filter : 4
map : 4
forEach : 16
filter : 5
```




```
package sec03;
import java.util.stream.IntStream;
public class Laziness1Demo {
   public static void main(String[] args) {
      IntStream is = IntStream.rangeClosed(1, 5);

      is.filter(x -> {
         System.out.println("filter : " + x);
         return x % 2 == 0;
      }).map(x -> {
         System.out.println("map : " + x);
         return x * x;
      });
   }
} //스트림 생성=> 필터링 연산=> 매핑 연산 출력연산 X
```

```
package sec03;
import java.util.stream.IntStream;
public class Laziness2Demo {
   public static void main(String[] args) {
      IntStream is = IntStream.rangeClosed(1, 5);

      is.filter(x -> {
         System.out.println("filter : " + x);
         return x % 2 == 0;
      }).map(x -> {
         System.out.println("map : " + x);
         return x * x;
      }).forEach(x ->
            System.out.println("forEach : " + x)
      );
   }
} //스트림 생성=> 필터링 연산=> 매핑 연산 출력연산 o
```

스트림 활용

■ 필터링

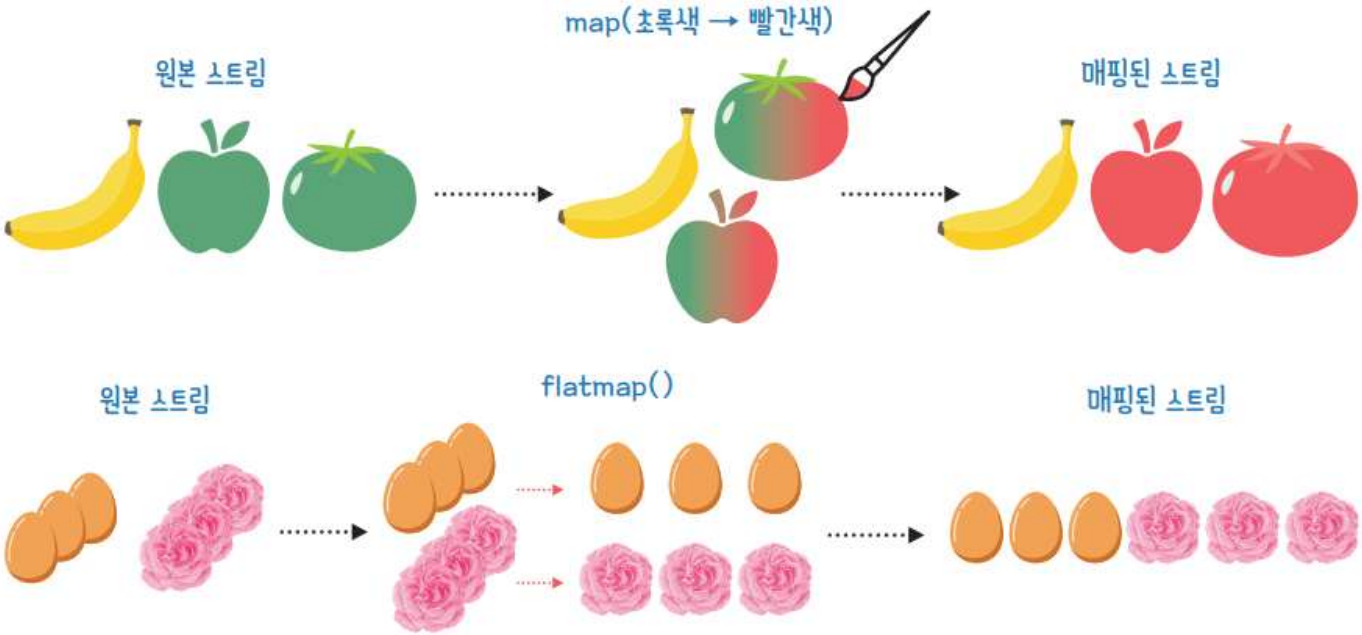
● 예제 : sec04/FilterDemo

문자열 스트림 : c2 c3
정수 스트림 : 2 4
인구가 1억(100백만) 이상의 2개 나라 : (USA) (China)

```
Stream.of("a1", "b1", "b2", "c1", "c2", "c3")
  .filter(s -> s.startsWith("c"))
  .skip(1).forEach(Util::print);
```

스트림 활용

■ 매핑



● 예제 : sec04/Map1Demo

```
A1 B1 B2 C1 C2
2 4 2 6 6 4 8
b1 b2 b3
```

quiz_1_식별자:스트림을 이용

List컬렉션을 데이터 소스로 사용하여 스트림을 생성 한 후
다양한 스트림 연산 작업을 수행하는 프로그램을 작성하시오.

```
package w13_111;

import java.util.List;
import java.util.stream.Stream;

public class quiz_1_111 {
    public static void main(String[] args) {
        Stream<String> ss;
        List<String> names = List.of("홍길동", "배장화", "임꺽정", "연흥부", "김선달", "황진이");

        ss = names.stream();
        ss.???(n -> n.charAt(0) < '이').???(n -> System.out.print(n + " "));
        System.out.println();
        ss = names.stream();
        ss.???().???(n -> System.out.print(n + " "));
        System.out.println();
}
}
```

```
배장화 연흥부 김선달
김선달 배장화 연흥부 임꺽정 홍길동 황진이
```

quiz_1_식별자:스트림을 이용

List컬렉션을 데이터 소스로 사용하여 스트림을 생성 한 후
다양한 스트림 연산 작업을 수행하는 프로그램을 작성하시오.

```
package w13_111;

import java.util.List;
import java.util.stream.Stream;

public class quiz_1_111 {
    public static void main(String[] args) {
        Stream<String> ss;
        List<String> names = List.of("홍길동", "배장화", "임꺽정", "연흥부", "김선달", "황진이");

        ss = names.stream();
        ss.filter(n -> n.charAt(0) < '이').forEach(n -> System.out.print(n + " "));
        System.out.println();
        ss = names.stream();
        ss.sorted().forEach(n -> System.out.print(n + " "));
        System.out.println();
}
}
```

```
배장화 연흥부 김선달
김선달 배장화 연흥부 임꺽정 홍길동 황진이
```