

2025년 상반기 K-디지털 트레이닝

스트림 요소 처리

[KB] IT's Your Life



○ 스트림을 이용하여 다음과 같이 출력하는 코드를 완성하세요.

```
package ch17.sec01;
import java.util.HashSet;
import java.util.lterator;
import java.util.Set;
import java.util.stream.Stream;
public class StreamExample {
         public static void main(String[] args) {
                  Set<String> set = new HashSet< >();
                  set.add("홍길동");
                  set.add("신용권");
                  set.add("감자바");
```

```
홍길동
신용권
감자바
```

StreamExample.java

```
package ch17.sec01;
                                               홍길동
import java.util.HashSet;
                                               신용권
import java.util.lterator;
                                               감자바
import java.util.Set;
import java.util.stream.Stream;
public class StreamExample {
         public static void main(String[] args) {
                 //Set 컬렉션 생성
                 Set<String> set = new HashSet< >();
                 set.add("홍길동");
                 set.add("신용권");
                 set.add("감자바");
                 //Stream을 이용한 요소 반복 처리
                 Stream<String> stream = set.stream();
                 stream.forEach( name -> System.out.println(name) );
```

○ 다음과 같이 Student 클래스를 작성하세요.

```
package ch17.sec03;
public class Student {
         private String name;
         private int score;
         public Student (String name, int score) {
                   this.name = name;
                   this.score = score;
         public String getName() { return name; }
         public int getScore() { return score; }
```

○ 스트림을 이용하여 다음과 같이 출력되도록 코드를 완성하세요.

스트림 처리시 메서드 체인닝을 사용함.

```
package ch17.sec03;
import java.util.Arrays;
import java.util.List;
public class StreamPipeLineExample {
         public static void main(String[] args) {
                  List<Student> list = Arrays.asList(
                                     new Student("홍길동", 10),
                                     new Student("신용권", 20),
                                     new Student("유미선", 30)
```

평균 점수: 20.0

StreamPipeLineExample.java

```
package ch17.sec03;
import java.util.Arrays;
import java.util.List;
public class StreamPipeLineExample {
         public static void main(String[] args) {
                 List<Student> list = Arrays.asList(
                                   new Student("홍길동", 10),
                                   new Student("신용권", 20),
                                   new Student("유미선", 30)
                 //방법1
                 Stream<Student> studentStream = list.stream();
                 //중간 처리(학생 객체를 점수로 매핑)
                 IntStream scoreStream = studentStream.mapToInt(student -> student.getScore());
                 //최종 처리(평균 점수)
                 double avg = scoreStream.average().getAsDouble();
                 System.out.println("평균 점수: " + avg);
```

평균 점수: 20.0

StreamPipeLineExample.java

평균 점수: 20.0

🗸 롬복을 이용하여 전체 매개변수를 가지는 생성자, Getter, Setter, toString을 정의하세요.

```
package ch17.sec04.exam01;

public class Product {
        private int pno;
        private String name;
        private String company;
        private int price;
}
```

Product.java

```
package ch17.sec04.exam01;

@AllArgsConstructor
@Data
public class Product {
    private int pno;
    private String name;
    private String company;
    private int price;
}
```

☑ 스트림을 이용하여 list에 담긴 Product를 출력하세요(toString 이용).

```
package ch17.sec04.exam01;
import java.util.ArrayList;
import java.util.List;
public class StreamExample {
         public static void main(String[] args) {
                  //List 컬렉션 생성
                  List<Product> list = new ArrayList<>();
                  for(int i=1; i<=5; i++) {
                            Product product = new Product(i, "상품"+i, "멋진회사", (int)(10000*Math.random()));
                            list.add(product);
```

StreamExample.java

```
{pno:1, name:상품1, company:멋진회사, price:6188}
package ch17.sec04.exam01;
                                            {pno:2, name:상품2, company:멋진회사, price:2510}
                                            {pno:3, name:상품3, company:멋진회사, price:9932}
import java.util.ArrayList;
                                            {pno:4, name:상품4, company:멋진회사, price:4317}
import java.util.List;
                                            {pno:5, name:상품5, company:멋진회사, price:170}
import java.util.stream.Stream;
public class StreamExample {
         public static void main(String[] args) {
                  //List 컬렉션 생성
                  List<Product> list = new ArrayList<>();
                  for(int i=1; i<=5; i++) {
                           Product product = new Product(i, "상품"+i, "멋진회사", (int)(10000*Math.random()));
                           list.add(product);
                  //객체 스트림 얻기
                  Stream<Product> stream = list.stream();
                  stream.forEach(p -> System.out.println(p));
```

○ 아래와 같이 출력되도록 스트림을 이용한 코드를 완성하세요.

```
package ch17.sec04.exam02;
import java.util.Arrays;
import java.util.stream.IntStream;
import java.util.stream.Stream;
public class StreamExample {
         public static void main(String[] args) {
                   String[] strArray = { "홍길동", "신용권", "김미나"};
                   int[] intArray = { 1, 2, 3, 4, 5 };
```

```
홍길동,신용권,김미나,
1,2,3,4,5,
```

StreamExample.java

```
package ch17.sec04.exam02;
import java.util.Arrays;
import java.util.stream.lntStream;
import java.util.stream.Stream;
public class StreamExample {
         public static void main(String[] args) {
                   String[] strArray = { "홍길동", "신용권", "김미나"};
                   Stream<String> strStream = Arrays.stream(strArray);
                   strStream.forEach(item -> System.out.print(item + ","));
                   System.out.println();
                   int[] intArray = { 1, 2, 3, 4, 5 };
                   IntStream intStream = Arrays.stream(intArray);
                   intStream.forEach(item -> System.out.print(item + ","));
                   System.out.println();
                                                     홍길동,신용권,김미나,
                                                     1,2,3,4,5,
```

☑ 1에서 100까지의 합을 구하고, 다음과 같은 출력하세요.

```
package ch17.sec04.exam03;
import java.util.stream.IntStream;
public class StreamExample {
    public static int sum; // 합계를 구하기 위한 변수
    public static void main(String[] args) {
    }
}
```

총합: 5050

StreamExample.java

```
package ch17.sec04.exam03;

import java.util.stream.IntStream;

public class StreamExample {
    public static int sum;

    public static void main(String[] args) {
        IntStream stream = IntStream.rangeClosed(1, 100);
        stream.forEach(a -> sum += a);
        System.out.println("善합: " + sum);
    }
}
```

총합: 5050

💟 다음과 같이 출력되도록 코드를 완성하세요.

```
package ch17.sec05;
import java.util.ArrayList;
import java.util.List;
                                                            홍길동
                                                            신용권
public class FilteringExample {
                                                            감자바
        public static void main(String[] args) {
                                                            신민철
                List<String> list = new ArrayList<>();
                list.add("홍길동");
                                                            신용권
                list.add("신용권");
                                                            신용권
                list.add("감자바");
                                                            신민철
                list.add("신용권");
                list.add("신민철");
                                                            신용권
                                                            신민철
                //중복 요소 제거
                //신으로 시작하는 요소만 필터링
                //중복 요소를 먼저 제거하고, 신으로 시작하는 요소만 필터링
```

FilteringExample.java

```
package ch17.sec05;
import java.util.ArrayList;
import java.util.List;
public class FilteringExample {
         public static void main(String[] args) {
                  //List 컬렉션 생성
                  List<String> list = new ArrayList<>();
                  list.add("홍길동");
                  list.add("신용권");
                  list.add("감자바");
                  list.add("신용권");
                  list.add("신민철");
```

FilteringExample.java

```
홍길동
//중복 요소 제거
                                                 신용권
list.stream()
                                                  감자바
        .distinct()
                                                 신민철
        .forEach(n -> System.out.println(n));
System.out.println();
                                                 신용권
                                                 신용권
//신으로 시작하는 요소만 필터링
                                                 신민철
list.stream()
        .filter(n -> n.startsWith("신"))
                                                 신용권
        .forEach(n -> System.out.println(n));
                                                  신민철
System.out.println();
//중복 요소를 먼저 제거하고, 신으로 시작하는 요소만 필터링
list.stream()
        .distinct()
        .filter(n -> n.startsWith("신"))
        .forEach(n -> System.out.println(n));
```

○ 다음과 같이 출력되도록 코드를 작성하세요.

```
package ch17.sec10;
import java.util.Arrays;
public class AggregateExample {
        public static void main(String[] args) {
                int[] arr = {1, 2, 3, 4, 5};
                //2의 배수 카운팅
                //2의 배수 총합
                //2의 배수 평균
                //2의 배수 중 최대값
                //2의 배수 중 최소값
                //첫 번째 3의 배수
```

2의 배수 개수: 2 2의 배수의 합: 6 2의 배수의 평균: 3.0 최대값: 4 최소값: 2

첫 번째 3의 배수: 3

AggregateExample.java

```
package ch17.sec10;
import java.util.Arrays;
public class AggregateExample {
         public static void main(String[] args) {
                  //정수 배열
                  int[] arr = \{1, 2, 3, 4, 5\};
                  //카운팅
                  long count = Arrays.stream(arr)
                                    .filter(n -> n%2==0)
                                    .count();
                  System.out.println("2의 배수 개수: " + count);
                  //총합
                  long sum = Arrays.stream(arr)
                                                             2의 배수 개수: 2
                                    .filter(n -> n%2==0)
                                                              2의 배수의 합: 6
                                    .sum();
                  System.out.println("2의 배수의 합: " + sum);
```

AggregateExample.java

```
//평균
double avg = Arrays.stream(arr)
                  .filter(n -> n%2==0)
                  .average()
                  .getAsDouble();
System.out.println("2의 배수의 평균: " + avg);
//최대값
int max = Arrays.stream(arr)
                  .filter(n -> n%2==0)
                  .max()
                  .getAsInt();
System.out.println("최대값: " + max);
//최소값
int min = Arrays.stream(arr)
                  .filter(n -> n%2==0)
                                           2의 배수의 평균: 3.0
                  .min()
                                           최대값: 4
                  .getAsInt();
                                            최소값: 2
System.out.println("최소값: " + min);
```

AggregateExample.java