

Chapter

14

람다와 스트림

Lambda & Stream

[연습문제]

[14-1] 메서드를 람다식으로 변환하여 아래의 표를 완성하십시오.

메서드	람다식
<pre>int max(int a, int b) { return a > b ? a : b; }</pre>	<pre>(int a, int b) -> a > b ? a : b</pre>
<pre>int printVar(String name, int i) { System.out.println(name+"="+i); }</pre>	<pre>(String name, int i) -> { System.out.println(name + "=" + i); } (name, i) -> { System.out.println(name + "=" + i); } (name, i) -> System.out.println(name + "=" + i)</pre>
<pre>int square(int x) { return x * x; }</pre>	<pre>(int x) -> x * x (x) -> x * x x -> x * x</pre>
<pre>int roll() { return (int) (Math.random() * 6); }</pre>	<pre>() -> { return (int) (Math.random() * 6); } () -> (int) (Math.random() * 6)</pre>
<pre>int sumArr(int[] arr) { int sum = 0; for(int i : arr) sum += i; return sum; }</pre>	<pre>(int[] arr) -> { int sum = 0; for (int i : arr) sum += i; return sum; }</pre>
<pre>int[] emptyArr() { return new int[]{}; }</pre>	<pre>() -> new int[] {}</pre>

[14-2] 람다식을 메서드 참조로 변환하여 표를 완성하십시오.
(‘변환불가’라고 적어야함.)

람다식	메서드 참조
<pre>(String s) -> s.length()</pre>	<pre>String::length</pre>
<pre>() -> new int[] {}</pre>	<pre>() -> new int[] {} int[]::new</pre>
<pre>arr -> Arrays.stream(arr)</pre>	<pre>arr -> Arrays.stream(arr) Arrays::stream</pre>
<pre>(String str1, String str2) -> str1.equals(str2)</pre>	<pre>(String str1, String str2) -> str1.equals(str2) String::equals</pre>
<pre>(a, b) -> Integer.compare(a, b)</pre>	<pre>(a, b) -> Integer.compare(a, b) Integer::compare</pre>
<pre>(String kind, int num) -> new Card(kind, num)</pre>	<pre>(String kind, int num) -> new Card(kind, num) Card::new</pre>
<pre>(x) -> System.out.println(x)</pre>	<pre>(x) -> System.out.println(x) System.out::println</pre>
<pre>() -> Math.random()</pre>	<pre>() -> Math.random() Math::random</pre>
<pre>(str) -> str.toUpperCase()</pre>	<pre>(str) -> str.toUpperCase() String::toUpperCase</pre>
<pre>() -> new NullPointerException()</pre>	<pre>() -> new NullPointerException() NullPointerException::new</pre>
<pre>(Optional opt) -> opt.get()</pre>	<pre>(Optional opt) -> opt.get() Optional::get</pre>
<pre>(StringBuffer sb, String s) -> sb.append(s)</pre>	<pre>(StringBuffer sb, String s) -> sb.append(s) StringBuffer::append</pre>
<pre>(String s) -> System.out.println(s)</pre>	<pre>(String s) -> System.out.println(s) System.out::println</pre>

[14-3] 아래의 괄호안에 알맞은 함수형 인터페이스는?

```
(    ) f; // 함수형 인터페이스 타입의 참조변수 f를 선언.
f = (int a, int b) -> a > b ? a : b;
```

- 1개
2개
3개
4개
5개
- a. Function
 - b. BiFunction
 - c. Predicate
 - d. ☒ BinaryOperator
 - e. IntFunction

```
public static void main(String[] args) {
    int n = 6;
    int[] dice1 = new int[n];
    int[] dice2 = new int[n];
    for (int i = 1; i <= n; i++) {
        dice1[i - 1] = i;
        dice2[i - 1] = i;
    }
    for (int i : dice1) {
        for (int j : dice2) {
            if (i + j != 6) { continue; }
            System.out.println("[" + i + ", " + j + "]");
        }
    }
}
```

[14-4] 두 개의 주사위를 굴려서 나온 눈의 합이 6인 경우를 모두 출력하시오.

[Hint] 배열을 사용하시오.

[실행결과]

```
[1, 5]
[2, 4]
[3, 3]
[4, 2]
[5, 1]
```

```
public static void main(String[] args) {
    Stream<Integer> die = IntStream.rangeClosed(1, 6).boxed(); // raw type -> Integer
    die.flatMap(i -> Stream.of(1, 2, 3, 4, 5, 6).map(i2 -> new int[] { i, i2 }))
        .filter(iArr -> iArr[0] + iArr[1] == 6)
        .forEach(iArr -> System.out.println(Arrays.toString(iArr)));
}
```

[14-5] 문자열 배열 strArr의 모든 문자열의 길이를 더한 결과를 출력하시오.

```
String[] strArr = { "aaa", "bb", "c", "dddd" };
```

[실행결과]

sum=10

```
public static void main(String[] args) {
    String[] strArr = { "aaa", "bb", "c", "dddd" };
    Stream<String> strStream = Stream.of(strArr);
    //int sum = strStream.mapToInt(s -> s.length()).sum();
    int sum = strStream.mapToInt(String::length).sum();
    System.out.println("sum=" + sum);
}
```

[14-6] 문자열 배열 strArr의 문자열 중에서 가장 긴 것의 길이를 출력하시오.

```
String[] strArr = { "aaa", "bb", "c", "dddd" };
```

[실행결과]

4

```
public static void main(String[] args) {
    String[] strArr = { "aaa", "bb", "c", "dddd" };
    Stream<String> strStream = Stream.of(strArr);
    strStream.map(String::length)
        .sorted(Comparator.reverseOrder()) // descending order
        .limit(1).forEach(System.out::println);
}
```

[14-7] 임의의 로또번호(1~45)를 정렬해서 출력하시오.

【실행결과】

```
1
20
25
33
35
42
```

```
public static void main(String[] args) {
    new Random().ints(1, 46)
        .distinct()
        .limit(6)
        .sorted()
        .forEach(System.out::println);
}
```

[14-8] 다음은 불합격(150점 미만)한 학생의 수를 남자와 여자로 구별하여 출력하는 프로그램이다. (1)에 알맞은 코드를 넣어 완성하시오.

【연습문제】/ch14/Exercise14_8.java

```
import java.util.*;
import java.util.function.*;
import java.util.stream.*;
import static java.util.stream.Collectors.*;
import static java.util.Comparator.*;

class Student {
    String name;
    boolean isMale; // 성별
    int hak;        // 학년
    int ban;        // 반
    int score;

    Student(String name, boolean isMale, int hak, int ban, int score) {
        this.name = name;
        this.isMale = isMale;
        this.hak = hak;
        this.ban = ban;
        this.score = score;
    }

    String getName() { return name; }
    boolean isMale() { return isMale; }
    int getHak() { return hak; }
    int getBan() { return ban; }
    int getScore() { return score; }

    public String toString() {
        return String.format("[%s, %s, %d학년 %d반, %3d점]", name, isMale ?
                                "남":"여", hak, ban, score);
    }

    // groupingBy() 에서 사용
    enum Level { HIGH, MID, LOW } // 성적을 상, 중, 하 세 단계로 분류
}

class Exercise14_8 {
    public static void main(String[] args) {
        Student[] stuArr = {
```

```

        new Student("나자바", true, 1, 1, 300),
        new Student("김지미", false, 1, 1, 250),
        new Student("김자바", true, 1, 1, 200),
        new Student("이지미", false, 1, 2, 150),
        new Student("남자바", true, 1, 2, 100),
        new Student("안지미", false, 1, 2, 50),
        new Student("황지미", false, 1, 3, 100),
        new Student("강지미", false, 1, 3, 150),
        new Student("이자바", true, 1, 3, 200),

        new Student("나자바", true, 2, 1, 300),
        new Student("김지미", false, 2, 1, 250),
        new Student("김자바", true, 2, 1, 200),
        new Student("이지미", false, 2, 2, 150),
        new Student("남자바", true, 2, 2, 100),
        new Student("안지미", false, 2, 2, 50),
        new Student("황지미", false, 2, 3, 100),
        new Student("강지미", false, 2, 3, 150),
        new Student("이자바", true, 2, 3, 200)
    };

    Map<Boolean, Map<Boolean, Long>> failedStuBySex =
    /*
        (1) 알맞은 코드를 넣으시오.
    */

    long failedMaleStuNum = failedStuBySex.get(true).get(true);
    long failedFemaleStuNum = failedStuBySex.get(false).get(true);

    System.out.println("불합격[남자]:"+ failedMaleStuNum + "명");
    System.out.println("불합격[여자]:"+ failedFemaleStuNum + "명");
}
}

```

[실행결과]

불합격[남자]:2명

불합격[여자]:4명

```

// 14-8
Map<Boolean, Map<Boolean, Long>> failedStudBySex =
Stream.of(stuArr)
    .collect(
        partitioningBy(
            Student::isMale,
            partitioningBy(
                s -> s.getScore() < 150, counting()
            )
        )
    );

```

[14-9] 다음은 각 반별 총점을 학년 별로 나누어 출력하는 프로그램이다. (1)에 알맞은 코드를 넣어 완성하시오.

[연습문제]/ch14/Exercise14_9.java

```
import java.util.*;
import java.util.function.*;
import java.util.stream.*;
import static java.util.stream.Collectors.*;
import static java.util.Comparator.*;

class Student {
    String name;
    boolean isMale; // 성별
    int hak;        // 학년
    int ban;        // 반
    int score;

    Student(String name, boolean isMale, int hak, int ban, int score) {
        this.name = name;
        this.isMale = isMale;
        this.hak = hak;
        this.ban = ban;
        this.score = score;
    }

    String getName() { return name; }
    boolean isMale() { return isMale; }
    int getHak() { return hak; }
    int getBan() { return ban; }
    int getScore() { return score; }

    public String toString() {
        return String.format("[%s, %s, %d학년 %d반, %3d점]", name, isMale ?
                                "남":"여", hak, ban, score);
    }

    enum Level {
        HIGH, MID, LOW
    }
}

class Exercise14_9 {
    public static void main(String[] args) {
        Student[] stuArr = {
            new Student("나자바", true, 1, 1, 300),
            new Student("김지미", false, 1, 1, 250),
            new Student("김자바", true, 1, 1, 200),
            new Student("이지미", false, 1, 2, 150),
            new Student("남자바", true, 1, 2, 100),
            new Student("안지미", false, 1, 2, 50),
            new Student("황지미", false, 1, 3, 100),
            new Student("강지미", false, 1, 3, 150),
            new Student("이자바", true, 1, 3, 200),

            new Student("나자바", true, 2, 1, 300),
            new Student("김지미", false, 2, 1, 250),
```

```

        new Student("김자바", true, 2, 1, 200),
        new Student("이지미", false, 2, 2, 150),
        new Student("남자바", true, 2, 2, 100),
        new Student("안지미", false, 2, 2, 50),
        new Student("황지미", false, 2, 3, 100),
        new Student("강지미", false, 2, 3, 150),
        new Student("이자바", true, 2, 3, 200)
    };

    Map<Integer, Map<Integer, Long>> totalScoreByHakAndBan =
    /*
        (1) 알맞은 코드를 넣으시오.
    */

    for(Object e : totalScoreByHakAndBan.entrySet()) {
        System.out.println(e);
    }

    } // main의 끝
}

```

[실행결과]

```

1={1=750, 2=300, 3=450}
2={1=750, 2=300, 3=450}

```

```

// 14-9
Map<Integer, Map<Integer, Integer>> totalScoreByHakAndBan =
    Stream.of(stuArr)
        .collect(
            groupingBy(
                Student::getHak,
                groupingBy(
                    Student::getBan,
                    summingInt(Student::getScore)
                )
            )
        );

```