

# 23.12.15[JAVA]

🕒 작성 일시	@2023년 12월 15일 오전 9:42
🏷 태그	

generic docs

<https://docs.oracle.com/javase/tutorial/java/generics/index.html>

java 에러 발생 시

```
"C:\Program Files\Java\jdk-17\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2023.3
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: Index 8 out of bounds for length 7
    at javaExm12_14_03.main(javaExm12_14_03.java:16)
Process finished with exit code 1
```

click

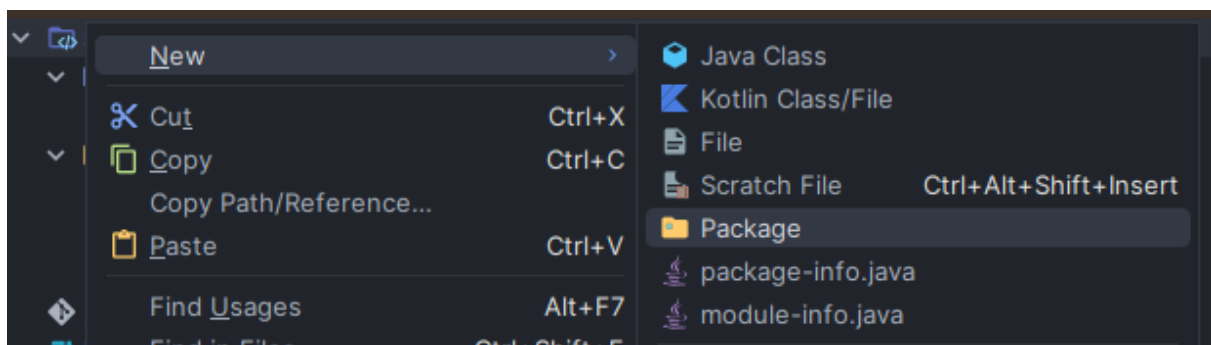
오류 발생 시 해당 오류 **document** 불러

```
public class ArrayIndexOutOfBoundsException extends IndexOutOfBoundsException {
    @java.io.Serial
    private static final long serialVersionUID = -5116101128118950844L;
```

java

오브젝트화 된 라이브러리가 많다.

디렉토리 만들기



## ArrayList

```
public class JavaExm12_15_01 {  
    public static void main(String[] args) {  
        ArrayList pitches = new ArrayList();  
        pitches.add(138);  
        pitches.add(129);  
        pitches.add(145);  
  
        System.out.println(pitches);           [138, 129, 145]  
        System.out.println(pitches.get(2));    145  
        System.out.println(pitches.size());    3  
        System.out.println(pitches.contains("145")); false  
        System.out.println(pitches.indexOf(129)); 1  
        System.out.println(pitches.remove(index: 1)); 129  
        System.out.println(pitches.size());    2  
        System.out.println(pitches);           [138, 145]  
    }  
}
```

get() 가져온다

size() 길이

contains() 조회(T/F)

indexOf() 조회(몇번째)

remove() 삭제

ArrayList(Arrays.asList()).1

```
String[] data = {"129", "145", "135"}; // 배열 만들기  
ArrayList<String> play = new ArrayList(Arrays.asList(data)); // data 의 오브젝트를 play 넣는다.  
System.out.println(play);
```

ArrayList(Arrays.asList()).2

```
ArrayList<String> play1 = new ArrayList(Arrays.asList("129", "135", "145"));  
System.out.println(play1);
```

## 배열 정렬

오름 `play.sort(Comparator.naturalOrder())` / 내림 `.reverseOrder()`

```
ArrayList<String> play = new ArrayList(Arrays.asList("129", "145", "135"));
play.sort(Comparator.naturalOrder()); // 오름차순 정렬
System.out.println(play);             [129, 135, 145]
play.sort(Comparator.reverseOrder()); // 내림차순 정렬
System.out.println(play);             [145, 135, 129]
```

## String.join

```
ArrayList<String> play = new ArrayList(Arrays.asList("129", "145", "135"));

String result = String.join(" ", play);

System.out.println(result);           129, 145, 135
```

## String.join[]

```
String[] play01 = new String[]{"129", "145", "135"};
String result01 = String.join(" ", play01);
System.out.println(result01);         129, 145, 135
```

`LinkedHashMap` / `TreeMap`

## LinkedHashMap/ Tree Map

```
//LinkedHashMap 입력 순서대로
Map<String, Integer> LinkedHashMap = new LinkedHashMap<>();
LinkedHashMap.put("sim",55);
LinkedHashMap.put("kim",15);
LinkedHashMap.put("zoo",35);

System.out.println(LinkedHashMap);
{sim=55, kim=15, zoo=35}
//TreeMap 오름차순
Map<String, Integer> TreeMap = new TreeMap<>();
TreeMap.put("sim",55);
TreeMap.put("kim",15);
TreeMap.put("zoo",35);

System.out.println(TreeMap);
{kim=15, sim=55, zoo=35}
```

HashSet[중복 값 제거, 오름 차순]

```
//HashSet 중복값 제거
HashSet<String> set = new HashSet<>(Arrays.asList("H","e","l","l","o"));
System.out.println(set);
```

retainAll[교집합]

## retainAll

```
HashSet<Integer> s1 = new HashSet<>(Arrays.asList(1,2,3,4,5,6,7));
HashSet<Integer> s2 = new HashSet<>(Arrays.asList(2,4,6,8,10));

// retainAll 교집합하는 자료 출력
HashSet<Integer> interSection = new HashSet<>(s1);
interSection.retainAll(s2);
System.out.println(interSection); [2, 4, 6]
```

addAll[합집합, 중복 제거]

## addAll

```
HashSet<Integer> union = new HashSet<>(s1);  
union.addAll(s2);  
System.out.println(union);  
[1, 2, 3, 4, 5, 6, 7]
```

removeAll[차집합]

## removeAll

```
HashSet<Integer> subtract = new HashSet<>(s1);  
subtract.removeAll(s2);  
System.out.println(subtract); [1, 3, 5, 7]
```

substring[범위 설정 출력]

## substring

```
String data= "880407-2345711";  
String b = data.substring(0,6);  
String n = data.substring(7,13);  
  
System.out.println("출생연도 : " + b);  
System.out.println("뒷번 : " + n);  
출생연도 : 880407  
뒷번 : 234571
```

and(&&)/or(||)연산



## if

```
// if && ||
int monney = 2000;
boolean hasCard = true;
if (monney >= 3000 && hasCard){
    System.out.println("버스를 이용하자");
}else{
    System.out.println("걸자");
}
```

걸자

```
//if contains
ArrayList<String> po =new ArrayList<String>();
po.add("go");
po.add("_");
po.add("home");
if (po.contains("go")){
    System.out.println("버스를 이용하자");
}else{
    System.out.println("걸자");
}
```

버스를 이용하자

Switch

```
//switch
int month = 6;
String monthString = "";
//switch 예는 byte short char int enum string 만 사용가능
switch (month) {
    case 1:
        monthString = "jan";
        break;
    case 2:
        monthString = "feb";
        break;
    case 3:
        monthString = "mar";
        break;
    case 4:
        monthString = "apr";
        break;
    default:
        monthString = "없네요";
        break;
}
System.out.println(monthString);
```

while문

```
int treeTen = 0;
while (treeTen<10){
    treeTen++;
    System.out.println("나무를"+ treeTen + "번 찍었어요");
    if (treeTen == 10){
        System.out.println("나무가 죽었네요!");
    }
}
```

나무를1번 찍었어요  
 나무를2번 찍었어요  
 나무를3번 찍었어요  
 나무를4번 찍었어요  
 나무를5번 찍었어요  
 나무를6번 찍었어요  
 나무를7번 찍었어요  
 나무를8번 찍었어요  
 나무를9번 찍었어요  
 나무를10번 찍었어요  
 나무가 죽었네요!



## for문

```
int[] mark={90, 55,60,75,85,43};
for (int i = 0; i < mark.length; i++) {
    if (mark[i] >= 60 ) {
        System.out.println((i+1)+"번은 합격");
    }else {
        System.out.println((i+1)+"번은 다음 생에 도전해");
    }
}
```

1번은 합격  
2번은 다음 생에 도전해  
3번은 합격  
4번은 합격  
5번은 합격  
6번은 다음 생에 도전해

## 2중 for문

```
public class JavaExm12_15_07 {
    public static void main(String[] args) {

        int sum ;
        for(int i=1; i<=9; i++){
            for(int k=2; k<=9; k++){
                System.out.print(k+"*"+i+"="+i*k+" ");
            }
            System.out.println(" ");
        }
    }
}
```

2\*1=2 3\*1=3 4\*1=4 5\*1=5 6\*1=6 7\*1=7 8\*1=8 9\*1=9  
2\*2=4 3\*2=6 4\*2=8 5\*2=10 6\*2=12 7\*2=14 8\*2=16 9\*2=18  
2\*3=6 3\*3=9 4\*3=12 5\*3=15 6\*3=18 7\*3=21 8\*3=24 9\*3=27  
2\*4=8 3\*4=12 4\*4=16 5\*4=20 6\*4=24 7\*4=28 8\*4=32 9\*4=36  
2\*5=10 3\*5=15 4\*5=20 5\*5=25 6\*5=30 7\*5=35 8\*5=40 9\*5=45  
2\*6=12 3\*6=18 4\*6=24 5\*6=30 6\*6=36 7\*6=42 8\*6=48 9\*6=54  
2\*7=14 3\*7=21 4\*7=28 5\*7=35 6\*7=42 7\*7=49 8\*7=56 9\*7=63  
2\*8=16 3\*8=24 4\*8=32 5\*8=40 6\*8=48 7\*8=56 8\*8=64 9\*8=72  
2\*9=18 3\*9=27 4\*9=36 5\*9=45 6\*9=54 7\*9=63 8\*9=72 9\*9=81

## while문 +[if]

```
//1000까지 수 중에 3의 배수의 합
int i=0;
int sum=1;
while(sum<1000){
    sum++;
    if( 0 == sum % 3 ){
        i= i+sum;
    }
}
System.out.println(i);    166833
```