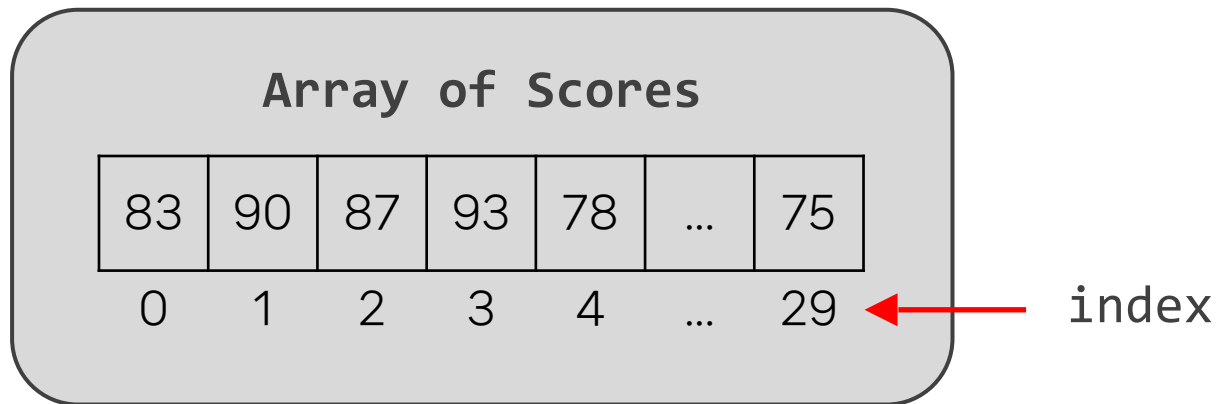


03_2 Arrays

Object-Oriented Programming

Definition of Array

- A data structure that lists **data of the same type** in a contiguous array, with **each element** having an **index**



```
score[0] = 83;  
score[1] = 90;  
...  
score[29] = 75;  
score[5] = score[2] + 3;
```

Declaration of Array

```
int[] iarray1;  
double[] darray1;  
String[] stArray1;
```

```
int iarray2[];  
double darray2[];  
String stArray2[];
```

- People generally prefer the former one.
- At this stage, memory has not yet been allocated for each array element.

Creating Array Objects

- Creating an array from a list of values
 - ex) `String[] capitals = {"Seoul", "ToKyo", "Beijing", "London"}`
 - ex) `int[] scores = {1, 2, 5, 13, -39}`
- Creating an array using the new operator
 - ex1) `int[] a1 = new int[5];`
 `a1[0] = 35;`
 `a1[3] = 70;`
 - ex2) `String[] str = new String[3];`
 `str[0] = "Seoul";`
 `str[1] = "Tokyo";`
 `...`

Default Initialization Values

- Each array element is automatically initialized as the **default initialization value** of that type.
- ex) `int[] a[3];` `// initialized as a[0] = a[1] = a[2] = 0`
- Default initialization values
 - byte, short, int, long, float, double: 0 (or 0.0f, 0.0)
 - char: `'\u0000'` (null character)
 - boolean: false
 - Reference Types (including String): null

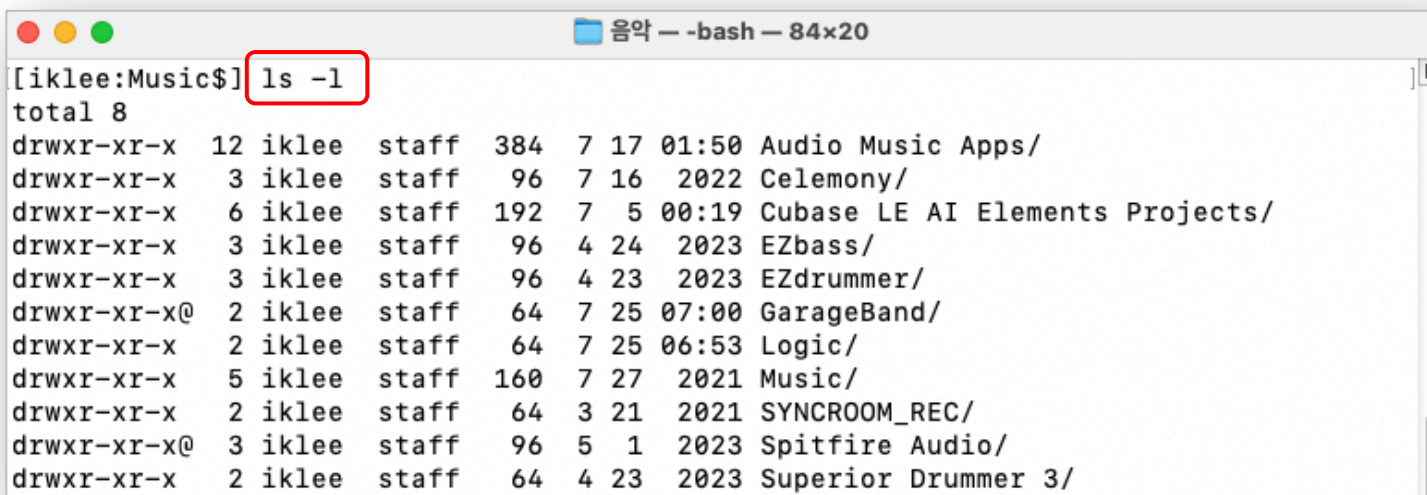
Length of an Array

- 'length' field of an array
- 'length' is a read-only field
- ex) `int[] intArray = {10, 20, 30};`
 `int size = intArray.length; // 3`

Command Line Arguments (1/4)

- An array of strings passed as a parameter to the main method.

```
public class CommandLineArguments {  
    public static void main(String[] args) {  
        System.out.println("args[0] = " + args[0]);  
        System.out.println("args[1] = " + args[1]);  
    }  
}
```



A terminal window titled "음악 - bash - 84x20" showing the output of the command `ls -l`. The command is highlighted with a red box. The output lists the contents of the current directory, including permissions, owner, group, size, date, and file name.

```
[iklee:Music$] ls -l  
total 8  
drwxr-xr-x 12 iklee staff 384 7 17 01:50 Audio Music Apps/  
drwxr-xr-x 3 iklee staff 96 7 16 2022 Celemony/  
drwxr-xr-x 6 iklee staff 192 7 5 00:19 Cubase LE AI Elements Projects/  
drwxr-xr-x 3 iklee staff 96 4 24 2023 EZbass/  
drwxr-xr-x 3 iklee staff 96 4 23 2023 EZdrummer/  
drwxr-xr-x@ 2 iklee staff 64 7 25 07:00 GarageBand/  
drwxr-xr-x 2 iklee staff 64 7 25 06:53 Logic/  
drwxr-xr-x 5 iklee staff 160 7 27 2021 Music/  
drwxr-xr-x 2 iklee staff 64 3 21 2021 SYNCROOM_REC/  
drwxr-xr-x@ 3 iklee staff 96 5 1 2023 Spitfire Audio/  
drwxr-xr-x 2 iklee staff 64 4 23 2023 Superior Drummer 3/
```

Command Line Arguments (2/4)

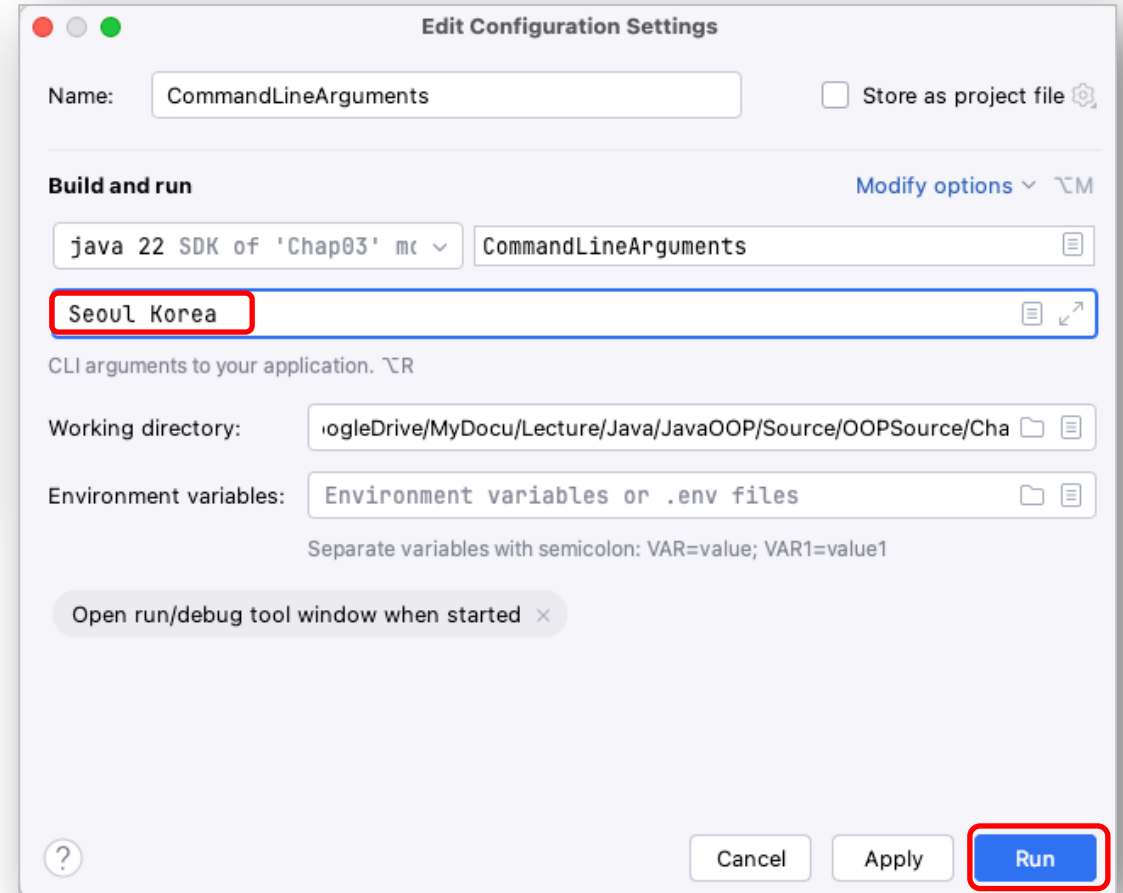
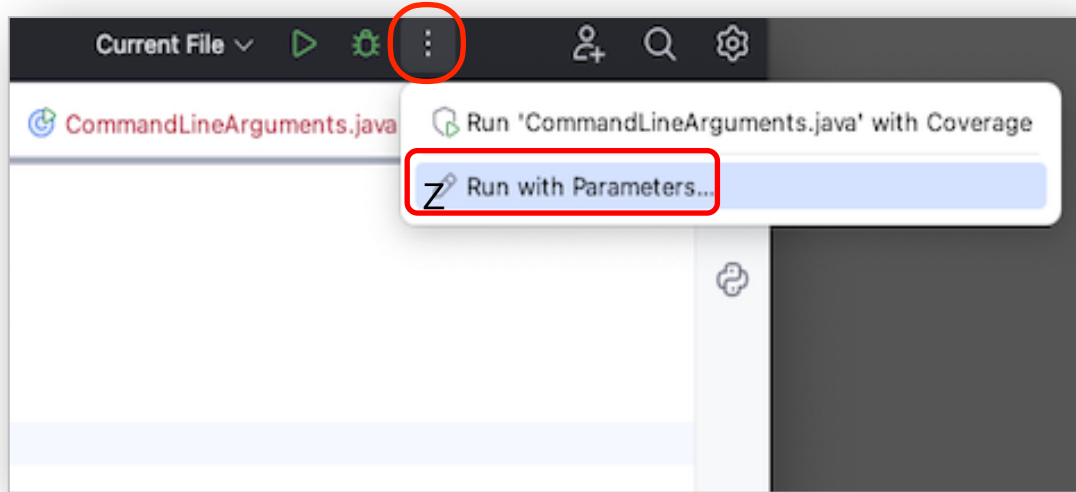
- An array of strings passed as a parameter to the main method.

```
public class CommandLineArguments {  
    public static void main(String[] args) {  
        System.out.println("args[0] = " + args[0]);  
        System.out.println("args[1] = " + args[1]);  
    }  
}
```

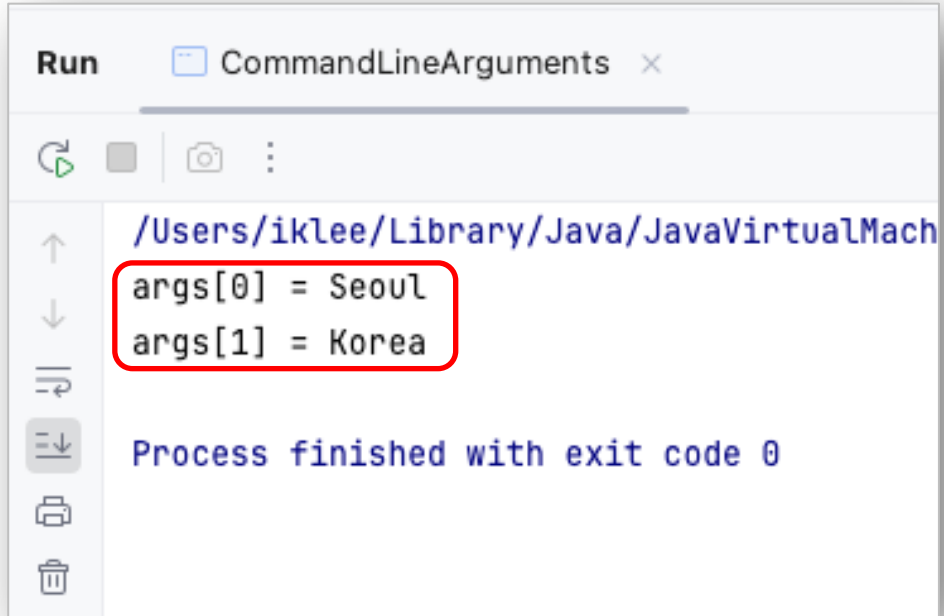
- If we try to run the program, exception occurs:

Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: Index 0 out of bounds for length 0
at CommandLineArguments.main(CommandLineArguments.java:3)

Command Line Arguments (2/3)



Command Line Arguments (3/3)

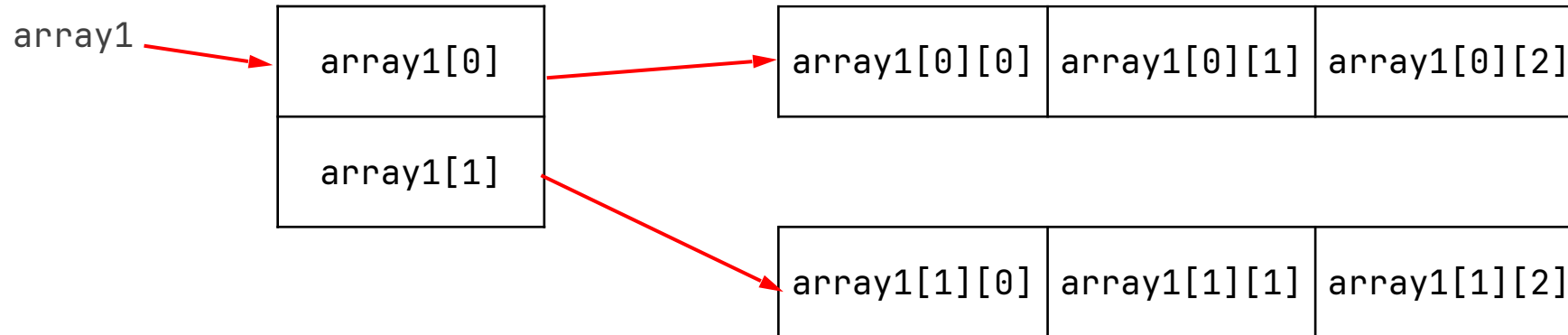


The screenshot shows the 'Run' console of an IDE. The title bar indicates the file 'CommandLineArguments'. The console output shows the file path '/Users/iklee/Library/Java/JavaVirtualMach' followed by two lines of code: 'args[0] = Seoul' and 'args[1] = Korea', which are enclosed in a red rectangular box. Below this, the message 'Process finished with exit code 0' is displayed. The left sidebar contains standard IDE icons for running, debugging, and viewing output.

```
/Users/iklee/Library/Java/JavaVirtualMach  
args[0] = Seoul  
args[1] = Korea  
  
Process finished with exit code 0
```

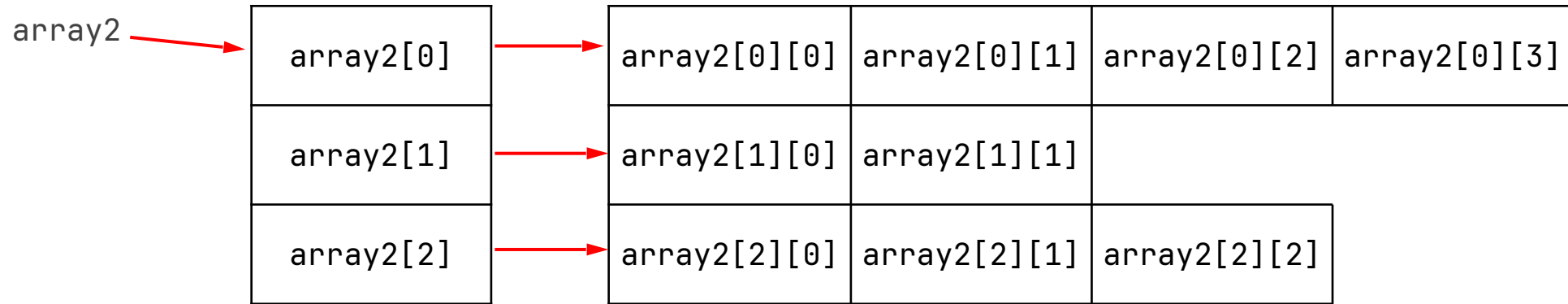
Multidimensional Array (1/2)

```
public class MultiDArray1 {  
    public static void main(String[] argv) {  
        int[][] array1 = new int[2][3];  
  
        for (int i = 0; i < array1.length; i++) // fill the array1  
            for (int j = 0; j < array1[i].length; j++) {  
                array1[i][j] = i + j;  
            }  
    }  
}
```



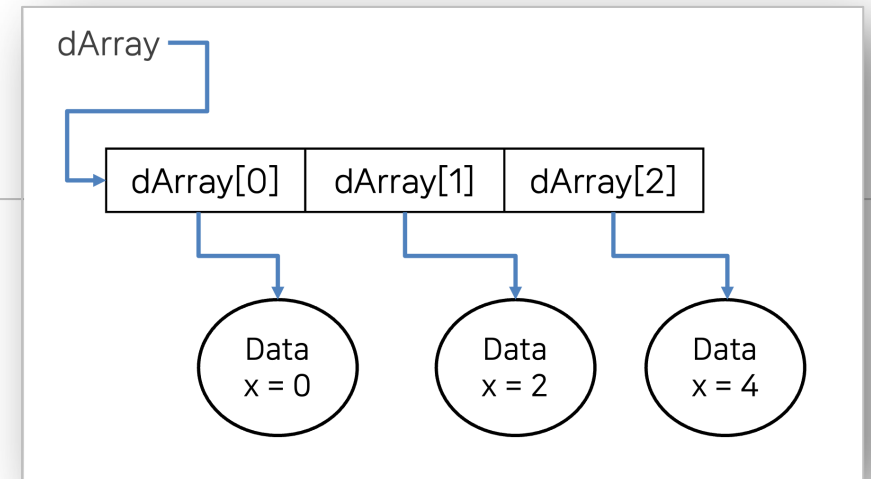
Multidimensional Array (2/2)

```
int[][] array2 = {{1, 2, 3, 4}, {5, 6}, {7, 8, 9}};
```



Array of Objects

```
class Data {  
    int x;  
}  
  
public class ArrayOfObjects {  
    public static void main(String[] argc) {  
        Data[] dArray = new Data[3]; // array of class object 'Data'  
        for (int i = 0; i < dArray.length; i++) {  
            dArray[i] = new Data(); // each object dArray[i] should be created  
            dArray[i].x = i * 2; // 0, 2, 4  
        }  
        for (int i = 0; i < dArray.length; i++) {  
            System.out.print(dArray[i].x + " "); // 0 2 4  
        }  
    }  
}
```



Enumeration Type (1/2)

- A type that stores one of the enumeration constants

```
enum Day {  
    SUNDAY,  
    MONDAY,  
    TUESDAY,  
    WEDNESDAY,  
    THURSDAY,  
    FRIDAY,  
    SATURDAY  
}
```

```
public class DayEnumDemo {  
    public static void main(String[] args) {  
        Day today = Day.WEDNESDAY;  
        System.out.println("Today is: " + today);  
        // Today is: WEDNESDAY  
        switch (today) {  
            case SATURDAY:  
            case SUNDAY:  
                System.out.println(today + " is a weekend.");  
                break;  
            default:  
                System.out.println(today + " is a weekday.");  
                break;  
        }  
        // WEDNESDAY is a weekday.  
    }  
}
```

Enumeration Type (2/2)

```
System.out.println("All days of the week:");  
for (Day day : Day.values()) {  
    String name = day.name();  
    int order = day.ordinal();  
    System.out.println(order + ") " + day + " " + name + " ");  
}  
Day theDay = Day.valueOf("FRIDAY");  
}
```

All days of the week:
0) SUNDAY SUNDAY
1) MONDAY MONDAY
2) TUESDAY TUESDAY
3) WEDNESDAY WEDNESDAY
4) THURSDAY THURSDAY
5) FRIDAY FRIDAY
6) SATURDAY SATURDAY