

day18-java

☰ 태그	
📅 날짜	@2022년 10월 25일

Array

같은 타입의 여러 변수를 하나의 묶음으로 다루는 것

만약 배열이 없다면, int a, b, c, d, e, f.... 등 많은 변수를 사용해야 한다.

- 기본 형태

```
int[] score = new int[5];
```

5개의 int 타입의 데이터를 저장할 수 있게 된다.

score ———> score[0], score[1], score[2], score[3], score[4]

위의 그림과 같이 배열은 각 저장 공간이 연속적으로 배치되어 있다는 특징을 볼 수 있다.
변수 score는 배열을 다루는데 필요한 변수명일 뿐 값을 저장하고 있다는 의미는 아닌 것이다.

1. 배열의 선언과 생성

```
타입[] 변수명; int[] score; String[] name;
```

```
타입 변수명[]; int score[]; String name[];
```

배열을 선언할 때는 원하는 타입 그리고 변수를 의미하는 [] 대괄호를 붙이면 된다.

참고로 대괄호는 타입 뒤나 변수명 뒤도 문제 없지만 타입 뒤에 붙이는 것을 선호한다.

```
타입[] 변수이름;           //배열을 선언하고
변수이름 = new 타입[갯수]; // 배열을 생성한다.

타입[] 변수이름 = new 타입[갯수]
```

1. 타입[] 변수이름; // 배열을 선언하면 이름만 만들어지고 저장할 수 있는 공간은 아직 만들어지지 않는다.
2. 변수이름 = new타입[갯수]; 연산자 new에 의해서 메모리의 빈공간에 지정한 수의 지정된 데이터 타입을 저장할수 있는 공간이 생긴다.

그리고 int 로 타입을 지정할 경우에는 자동으로 default 값인 0으로 초기화된다.

마찬가지로 String 으로 타입을 지정할 경우에는 default null 값으로 초기화된다.

- 기본적인 사용 방법

```
score[2]=100;           //배열에 넣는 기본
int value = score[0];   //배열에서 값을 꺼내는 기본
```

	0	0	100	0	0
score————>	score[0]	score[1]	score[2]	score[3]	score[4]
			value		

배열의 또 다른 장점은 index로 상수 대신 변수나 수식도 사용할 수 있다는 장점이 있다.

```
for(int i = 0; i < 5; i++) {
    score[i] = i *10;
}
```

배열을 다룰때 아주 중요한 점은 index값을 벗어나면 예외 발생된다는 점이다.

가장 많이 볼 예러 : `ArrayIndexOutOfBoundsException`

`score[0] : 50`

`score[1] : 60`

`score[2] : 70`

`score[3] : 80`

`score[4] : 90`

- 배열의 길이 (`arr.length`)

배열의 크기 갯수 길이는 `배열명.length`

```
package day18_1025;

public class array_test {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        int[] arr = new int[5];

        arr[0] = 50;
        arr[1] = 60;
        arr[2] = 70;
        arr[3] = 80;
        arr[4] = 90;

        for(int i = 0; i < arr.length; i++) {
            System.out.println("arr[" + i + "] = " + arr[i]);
        }
    }
}
```

배열의 길이를 바꾸는 경우 for문에도 바꾸어야 하기 때문에 예외가 발생할 경우가 생긴다.

따라서 배열의 길이를 직접 for문에 넣어주지 않고 length를 사용하여 넣어주면 오류 발생 여지가 줄어든다.

배열의 가장 큰 특징은 한번 생성하면 그 크기를 바꿀 수 없다.

하지만 강제로 바꾸어야 할 필요성이 생긴다.

- 배열의 길이를 변경하는 우회방법 2가지

1. 더 큰 배열을 새로 만든다. → 2. 기존 배열의 데이터를 옮긴다.

하지만 이러한 작업들은 또 다른 비용을 발생시키기 때문에 배열을 만들때 부터 넉넉하게 약 2배정도로 설정 하는것이 바람직.

- 배열을 만들고 사용하는 가장 간단한 방법

```
int[] score = new int[] {50,60,70,80,90};  
// []안에 갯수를 넣으면 안된다.  
  
//더 간단한 방법  
int[] arr= {10,20,30,40,50};  
  
//  
import java.util.Arrays;  
System.out.println(Arrays.toString(score));
```

```
27
28     System.out.println(Arrays.toString(score));
29     System.out.println(score);
30 }
31
32 }
33
```

Console Problems Debug Shell Call Hierarchy

<terminated> array_test [Java Application] C:\Program Files\Java\jdk-17.0.4\bin\javaw.exe (2022. 10. 25)

[50, 60, 70, 80, 90]
[I@28864e92

배열의 복사

5개 짜리 배열

```
int[] arr = new int[5];
```

갯수가 부족하다..

```
int[] tmp = new int[arr.length*2];
```

기존 arr 2배의 크기로 임시 배열을 생성한다.

for문으로 기존 배열의 데이터를 tmp배열로 넣는다.

```
arr = tmp; // 기존 참조 배열 변수 arr이 새로운 배열을 가리키도록 만든다.
```

몇개의 배열을 생성할지 입력 받고 각 배열에 값을 입력받아서 배열을 만들어라.

```

array_test2.java

package day18_1025;
import java.util.Arrays;
import java.util.Scanner;
public class array_test2 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        System.out.print("몇개짜리 배열을 생성할까요? :");
        Scanner sc = new Scanner(System.in);
        int a = sc.nextInt();
        int i = 0;
        int[] arr = new int[a];

        for (i=0; i<a; i++) {
            System.out.print("arr["+i+"]= ");
            int b = sc.nextInt();

            arr[i] = b;

        }System.out.println(Arrays.toString(arr));
    }
}

```

```

몇개짜리 배열을 생성할까요? :5
arr[0]= 10
arr[1]= 20
arr[2]= 30
arr[3]= 40
arr[4]= 50
|[10, 20, 30, 40, 50]

```

※ 배열을 복사하는 쉬운 방법 = System.arraycopy();

```

System.arraycopy(배열명, 시작, 새배열명, 시작, 배열명.length);

//System.arraycopy
//(복사할 배열, 복사할 배열의 시작점, 복사될 배열, 복사될 배열의 시작점, 몇개를 복사할건지);

```

array.java

```

package day18_1025;

import java.util.Arrays;

public class array {

    public static void main(String[] args) {

        int[] arr = new int[] {10,20,30,40,50};
        int[] tmp = new int[arr.length*2];

        System.arraycopy(arr,0 ,tmp,0 ,5);
        System.out.println(Arrays.toString(tmp));
    }
}

```

multi-dimensional array

2차원 배열 - 가로 세로의 형태 배열

생성방법

`int[][] arr;`

`int arr [][];`

`int[] arr[];`

`int[][] score = new int[4][3]; //4행 3열`

	0	1	2	3
0	<code>int[0][0]</code>	<code>int[0][1]</code>	<code>int[0][2]</code>	<code>int[0][3]</code>
1	<code>int[1][0]</code>	<code>int[1][1]</code>	<code>int[1][2]</code>	<code>int[1][3]</code>
2	<code>int[2][0]</code>	<code>int[2][1]</code>	<code>int[2][2]</code>	<code>int[2][3]</code>
3	<code>int[3][0]</code>	<code>int[3][1]</code>	<code>int[3][2]</code>	<code>int[3][3]</code>

```
score[0][0] = 100;
System.out.println(score[0][0]);
```

```
int[][] arr = { {1,2,3}, {4,5,6}};

=

int[][] arr = {
    {1,2,3},
    {4,5,6}
};
```

```
array_test3.java
package day18_1025;

import java.util.Arrays;

public class array_test3 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        int[][] arr = { {1,2,3}, {4,5,6}};

        for(int i = 0; i<arr.length; i++) {
            for(int j = 0; j <=arr.length; j++) {
                System.out.println("arr["+i+"]["+j+"]=" + arr[i][j]);
            }

        }System.out.println(Arrays.deepToString(arr));
    }

}
```



```
arr[0][0]=1
arr[0][1]=2
arr[0][2]=3
arr[1][0]=4
arr[1][1]=5
arr[1][2]=6
[[1, 2, 3], [4, 5, 6]]
```

가변 배열

만약 5 * 3, 즉 5행 3열 2차원 배열을 생성할때.

```
int[][] score = new int[5][3];
```

위에서와 같이 배열을 생성할 때 갯수를 지정했다.

그런데

```
int[][] score = new int[5][ ]; // 두번째 즉 열을 지정하지 않았다.
```

```
package day18_1025;

public class array_test4 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        int[][] score = {
            {1,2,3,4},
            {1,2,3},
            {1,2},
            {1,2,3,4,5,6}
        };

    }

}

// 에러가 나지 않는다.
```

사람 수:5
점수를 입력하세요.
1번의 점수:72
2번의 점수:54
3번의 점수:68
4번의 점수:32
5번의 점수:92
합계는 318점입니다.
평균은 63.6점입니다.
최고점은 92점입니다.

```
array_quiz.java

package day18_1025;

import java.util.Arrays;
import java.util.Scanner;

public class array_quiz {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        System.out.print("사람 수 :");
        Scanner sc = new Scanner(System.in);
        int a = sc.nextInt();
        int[] arr = new int[a];
        int sum = 0;
        int max = 0;
        System.out.println("점수를 입력하세요.");

        for(int i = 0; i < a; i++) {
            System.out.print((i+1)+"번의 점수:");
            arr[i] = sc.nextInt();
            sum += arr[i];
            if(arr[i] > max) {
                max = arr[i];
            }
        }
    }
}
```

```

    }
}
System.out.println(Arrays.toString(arr));
System.out.println("합계는" + sum + "입니다.");
System.out.println("평균은" + (double)(sum/a) + "입니다.");
System.out.println("최고점은" + max + "입니다.");

}

}

```

```

사람 수 :8
점수를 입력하세요.
1번의 점수:11
2번의 점수:12
3번의 점수:13
4번의 점수:14
5번의 점수:15
6번의 점수:16
7번의 점수:17
8번의 점수:18
[11, 12, 13, 14, 15, 16, 17, 18]
합계는116입니다.
평균은14.0입니다.
최고점은18입니다.

```