

Multi - Dimentional Array

Multi - Dimentional Array - 2차원 배열

- `DataType[][] name = new DataType[세로][가로];`
- 구조는 세로, 가로 2차원의 형태를 가지게 된다.

```
int[][] arr = new int[3][2];
```

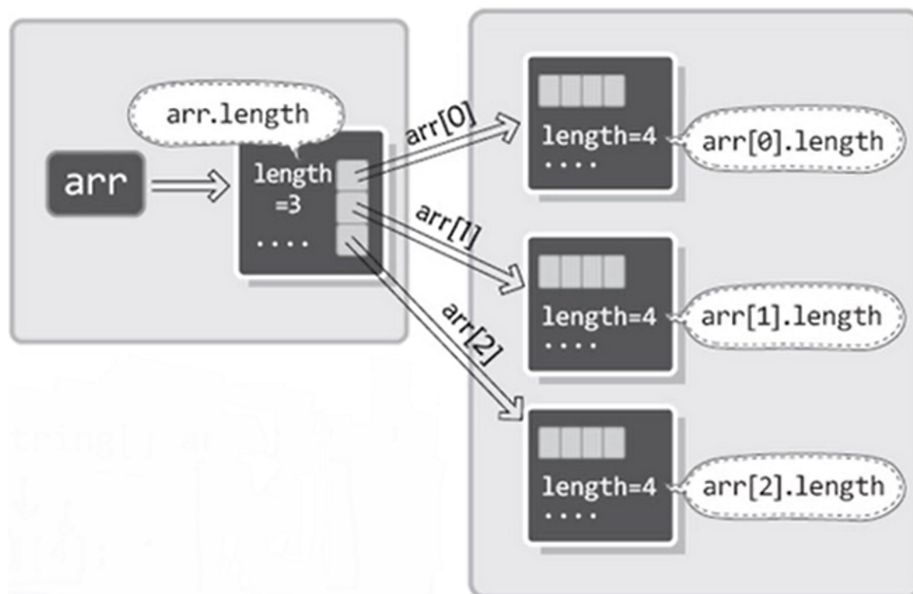
| | |
|-------------|-------------|
| index[0][0] | index[0][1] |
| index[1][0] | index[1][1] |
| Index[2][0] | index[2][1] |

2차원 배열의 메모리 구조

- `int[][] arr = new int[3][4];`
길이가 4인 1차원 배열 3개를 묶어놓은 것!
- i. 길이가 3인 int형 Array Instance arr을 만든다
- ii. 그 후에 다음과 같이 길이가 4인 Array Instance 3개를 만든다.

```
arr[0] = new int[4];  
arr[1] = new int[4];  
arr[2] = new int[4];
```

- iii. 길이가 4인 각각의 Array Instance의 참조변수는 arr의 요소 하나하나가 된다.



```
class MultiDimensionalArray {  
    public static void main(String[] args) {  
        int[][] arr = new int[3][4];  
  
        for(int i=0; i<arr.length; i++) {  
            for(int j=0; j<arr[i].length; j++)  
                arr[i][j] = i + j;  
        }  
  
        for(int i=0; i<arr.length; i++) {
```

```

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

        System.out.println("");
    }
}

```

2차원 배열의 선언 및 초기화

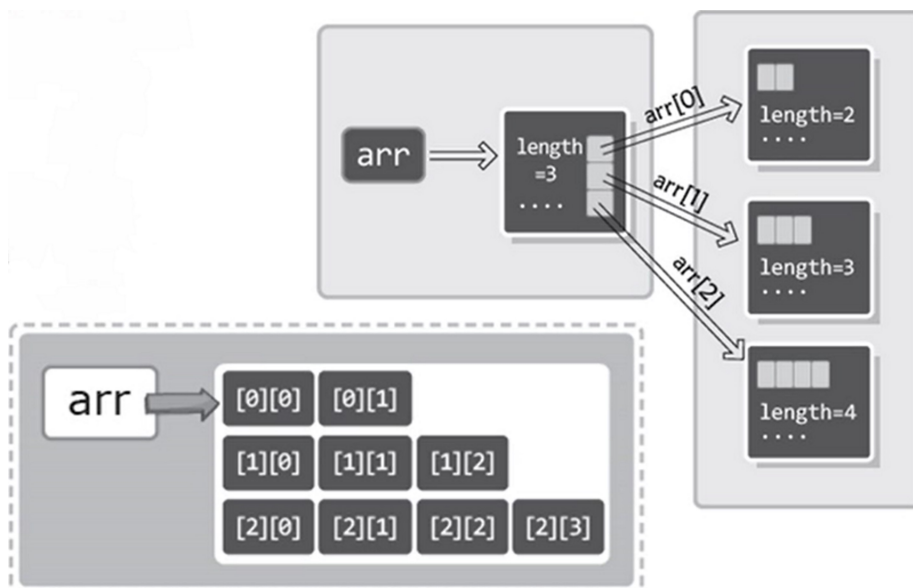
```

i. int[][] = {
    {1, 2, 3, 4},
    {5, 6, 7, 8},
    {9, 10, 11, 12}
}; // 세로[3], 가로[4]

ii. int[][] = new int[3][4]{
    {1, 2, 3, 4},
    {5, 6, 7, 8},
    {9, 10, 11, 12}
}; // 세로[3], 가로[4]

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

```



2차원 이상의 다차원 배열의 선언 및 초기화

- 특정 차원의 배열을 요소로 하고 묶어서 감싸는 배열을 선언하면 그 배열은 한차원 더 높은 배열이 된다.
- 1차원 배열의 배열 : 2차원 배열
- 2차원 배열의 배열 : 3차원 배열
- 3차원 배열의 배열 : 4차원 배열.....

Ex1) 3차원 배열

```

int[][][] threeDimArr =
{
    {
        {0,0},
        {0,0}
    },

```

```

        {
            {0,0},
            {0,0}
        }
    }
}

```

Ex2) 4차원 배열

```

int[][][][] fourDimArr =
{
    {
        {
            {0,0},
            {0,0}
        },
        {
            {0,0},
            {0,0}
        }
    },
    {
        {
            {0,0},
            {0,0}
        },
        {
            {0,0},
            {0,0}
        }
    }
}

```

for - each

```

package org.opentutorials.javatutorials.array;

public class ForeachDemo {
    public static void main(String[] args) {
        String[] members = { "최진혁", "최유빈", "한이람" };

        for (String e : members) {
            System.out.println(e + "이 상담을 받았습니다");
        }
    }
}

```

for(DataType Variable : Array Instance의 참조변수)

위의 구문은 배열 members의 요소 하나하나의 값을 변수 e에 담아서 중괄호 구간 안으로 전달해준다.
반복문의 종료조건이나 종료조건을 위해서 기준값을 증가시키는 등의 반복적인 작업을 내부적으로 감춘 것이라고 할 수 있다.
자바 5.0부터 도입된 기능이다.
