

● 다차원 배열: 2차원 이상의 배열. 타입[][] 변수이름; 으로 선언할 수 있다. 2차원 배열은 주로 테이블 형태를 데이터를 담는데 사용된다. (3차원 이상의 고차원 배열의 선언은 대괄호[] 개수를 차원의 수만큼 추가해 주면 된다.) 다차원 배열은 '배열의 배열'의 형태로 처리한다.

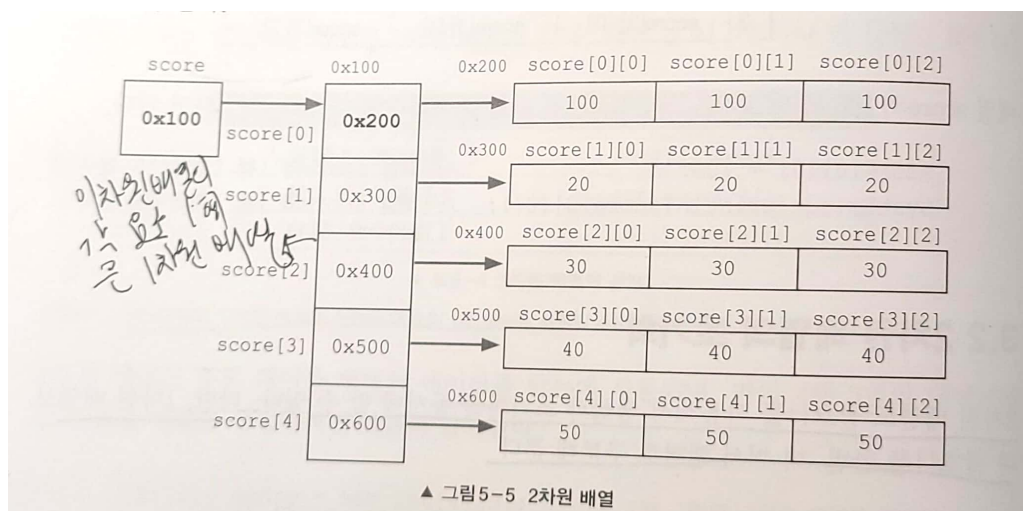
● 이차원 배열: 1차원 배열의 배열.

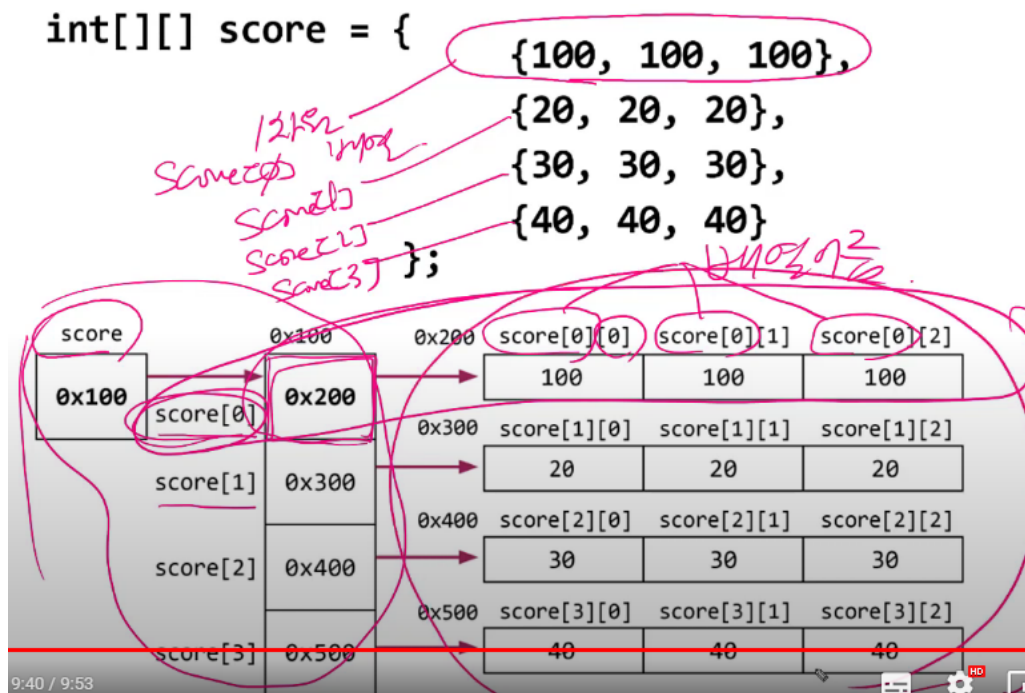
```
타입 [][] 변수이름 = new 타입[행의 길이][열의 길이];
// 모두 행의 길이 x 열의 길이 만큼의 값을 저장할 수 있는 공간이 마련된다.
```

- 배열의 생성하면, 배열의 각 요소에는 배열요소타입의 기본값이 자동적으로 저장된다.
- 2차원 배열의 '행index'의 범위는 '0~행의길이-1'이고, '열index'의 범위는 '0~열의길이-1'이다.
- 2차원 배열의 각 요소에 접근하는 방법은 '배열이름[행index][열index]'이다.
- 2차원 배열도 괄호를{}써서 생성과 초기화를 동시에 할 수 있는데, 괄호{}를 한번 더 써서 행별로 구분해 준다.

```
타입 [][] 변수이름 = {
    {요소1, 요소2, 요소3},
    {요소4, 요소5, 요소6}
};
```

- 2차원 배열은 '배열의 배열'로 구성되어 있다. 1차원 배열들의 참조변수들을 하나의 배열로 또 묶은 것이다. 즉, 여러 개의 1차원 배열을 묶어서 또 하나의 배열로 만든 것이다. 2차원 배열의 각 요소(행)은 1차원 배열이다.
- (1) 배열의 이름인 참조변수 score을 위한 공간 생성 (2) 크기가 5인 일차원 배열 생성. 각 배열 score[0~4]은 실제 값을 담는 배열의 메모리 주소를 저장하는 참조변수이다. (3) 크기가 3인 int형 배열 생성. 각 배열의 메모리값 주소는 참조변수 score[0~4]에 저장된다.





- 배열.length;는 배열의 행의 길이와 같으며, 배열[i].length;의 길이는 열의 길이와 같다.

```
package ch5;

class ArrayEx18 {
    public static void main(String[] args) {
        int[][] score = {
            {100, 100, 100},
            {20, 20, 20},
            {30, 30, 30},
            {40, 40, 40}
        };

        int sum = 0;

        for(int i=0; i < score.length; i++) { // score 배열의 모든 요소 출력하기
            for(int j=0; j < score[i].length; j++) {
                System.out.printf("score[%d][%d]=%d\n", i, j, score[i][j]);
            }
        }

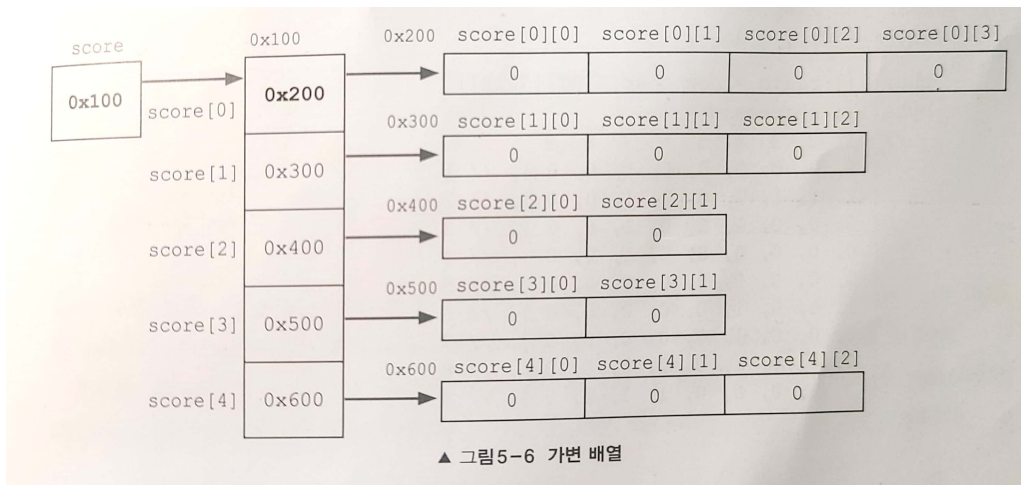
        for (int[] tmp : score) { // score의 각 요소(1차원 배열 주소)를 tmp에 저장
            for (int i : tmp) { // tmp는 1차원 배열을 가리키는 참조변수
                sum += i;
            }
        }

        // for (int[] tmp : score) { // 모든 요소를 출력하기. score의 각 요소(행)만큼 돌며
        //     for (int i : tmp) { // 각 요소가 참조하고 있는 배열의 길이만큼(열) 돈다
        //         System.out.println "[" + i + "]"; // 모든 요소 출력
        //     }
        // }

        System.out.println("sum="+sum);
    }
}
```

● 가변 배열: 전체 배열 차수 중 마지막 차수의 길이를 지정하지 않고, 추후에 각기 다른 길이의 배열을 생성함으로써 고정된 형태가 아닌 보다 유동적인 가변 배열을 구성할 수 있다. (행의 길이는 고정하고, 열의 길이는 각 행마다 다르게 설정할 수 있다).

```
int[][] score = new int[5][];           // 두 번째 차원의 길이는 지정하지 않는다.
score[0] = new int[4];
score[1] = new int[3];
score[2] = new int[2];
score[3] = new int[2];
score[4] = new int[3];
```



- 1) 참조변수 score을 위한 공간이 만들어진다. 2) 크기가 5인 일차원 배열이 생성된다. 이 배열은 배열의 참조변수를 담는데 사용된다. 참조 변수의 기본 값인 null로 모두 초기화된다. 3) 크기가 4인 int형 배열을 생성한 다음, 그 주소를 참조변수 score[0]에 저장한다. 4) 같은 과정을 score[4]까지 반복한다.
- 이 경우 score.length의 값은 여전히 5로 고정이지만, score[0].length의 값과 score[1].length의 값은 다르다.
- 가변 배열 역시 중괄호{}를 이용해서 생성과 초기화를 동시에 하는 것이 가능하다.

```
int[][] score = {
    {100, 100, 100, 100},
    {20, 20, 30},
    {40, 40}
};
```

Resource: 자바의 정석