

22.11.24 - Array2 (2 Dimension)

おさらい

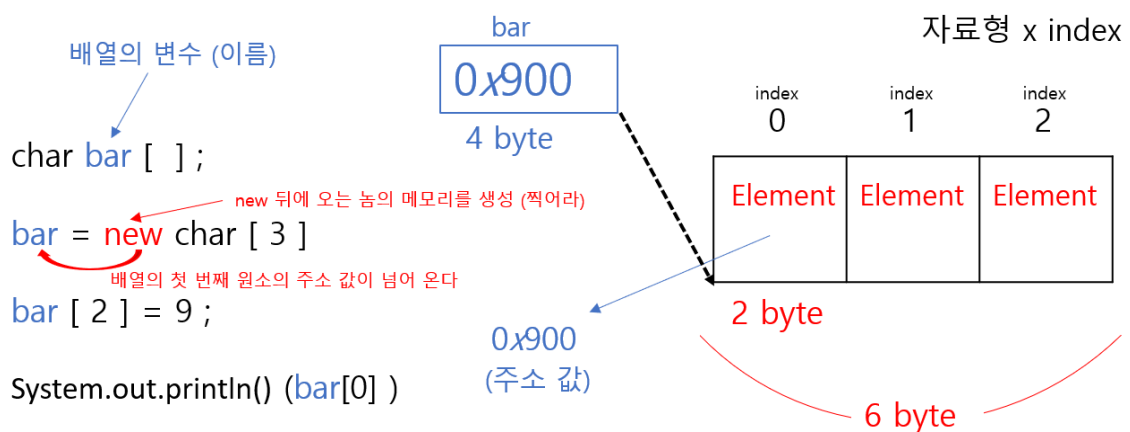
구조적 자료형

여러 자료를 묶어서 하나의 단위로 처리하는 자료형을

구조적 자료형(structured data type) 이라 하는데,

배열(동질형 자료의 모임) 과 레코드(이질형 자료의 모임) 로 구분할 수 있다.

c언어에 서는 레코드를 (구조체) 라고 한다



- 메모리 상에서 배열의 변수는 만들어진 배열의 첫 번째 주소를 담기 위해서 4byte로 정해진다 (주소의 크기는 jvm에 의존적)
- 메모리상에서 데이터가 있으면 어디에 있는지 구분하기 위해 **메모리 주소** 를 사용한다
- 각각의 데이터는 메모리 상의 주소 값을 가진다

변수의 동작 모드 (변수로 할 수 있는 것)

- **get** - 변수의 값을 읽어 오는 것
- **set** - 변수에 값을 저장하는 것

변수의 초기 값 = 변수가 만들어지면서 처음으로 넣는 값

배열의 초기 값 \Rightarrow 배열의 변수 선언 후 뒤에 new 찍을 필요 없이 리터럴 상수가 블레이스에 싸여져 있으면 jvm 에서 자동으로 배열을 생성한다

배열을 생성하면서 초기 값까지 넣는 방법

```
int bar[ ] = {10 , 7 , 3};
```

```
// 초기값 10

int foo[] = new int[3];

foo[0] = 10;
foo[1] = 7;
foo[2] = 3;

// 배열을 생성하면서 초기 값까지 넣을 수 있는 방법
int bar[ ] = {10 , 7 , 3};

System.out.println(bar[2]);
```

\Rightarrow 블레이스에 묶여 있으면 자동적으로 배열 생성 , 콤마개수 + 1 = 원소의 개수

\Rightarrow 리터럴 상수를 이용해서 배열을 생성하면서 동시에 초기 값까지 넣을 수 있다

```
// 방법 1      (C언어 스타일 )

// h e l l o
char bar [ ] = {'h','e','l','l','o'};

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

// 방법 2      (Java 스타일 )
// h e l l o
char bar[] = new char[]{ 'h', 'e', 'l', 'l', 'o' }; // 브라켓 안에 정수의 개수 x
//정수의 개수는 뒤에오는 리터럴 상수 개수와 매핑

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

※ 상수는 자료형 앞에 `final` 를 쓰고 상수 이름을 대문자로 쓴다 ※

배열의 원소 개수는 1개의 상수로 관리하면 편하다

```
// 1 ~ 50 사이 정수 중 난수로 10개를 선택하시오 => 개수가 10개인 INT 형 배열 만들기

final int THE_NUM_OF_ELEMENT = 10;    // 상수

int bar [ ] = new int[THE_NUM_OF_ELEMENT]; // 배열의 크기는 하나의 상수로 관리 한다 !

for (int i = 0; i < 10; i++) {

    bar[i]= (int)(Math.random()*50)+1;

}

for (int i = 0; i < 10; i++) {
    System.out.println(" " + bar[i]);
}
```

2차원 배열

배열에도 차원이 있다.

차원 (Dimension)

- 하나의 점이 있으면 그 점의 좌표를 나타내는 축의 개수
- ex) 3D (3 Dimension)

Scalar Vector Matrix Tensor

1

$$\begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

$$\begin{bmatrix} \begin{bmatrix} 1 & 2 \end{bmatrix} & \begin{bmatrix} 3 & 2 \end{bmatrix} \\ \begin{bmatrix} 1 & 7 \end{bmatrix} & \begin{bmatrix} 5 & 4 \end{bmatrix} \end{bmatrix}$$

1.1. 스칼라, 벡터, 행렬, 텐서.

- 스칼라는 하나의 숫자를 의미합니다.
- 벡터는 숫자(스칼라)의 배열입니다.

`int bar = 3`

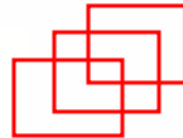
7	8	9
---	---	---

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix}$$

- 행렬은 2차원의 배열입니다.

$$\mathbf{A} = \begin{bmatrix} A_{1,1} & A_{1,2} & \dots & A_{1,n} \\ A_{2,1} & A_{2,2} & \dots & A_{2,n} \\ \dots & \dots & \dots & \dots \\ A_{m,1} & A_{m,2} & \dots & A_{m,n} \end{bmatrix}$$

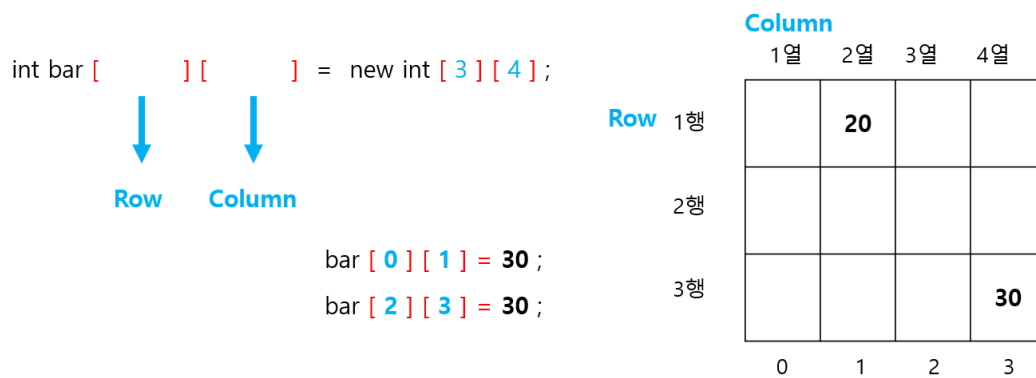
- 텐서는 2차원 이상의 배열입니다.



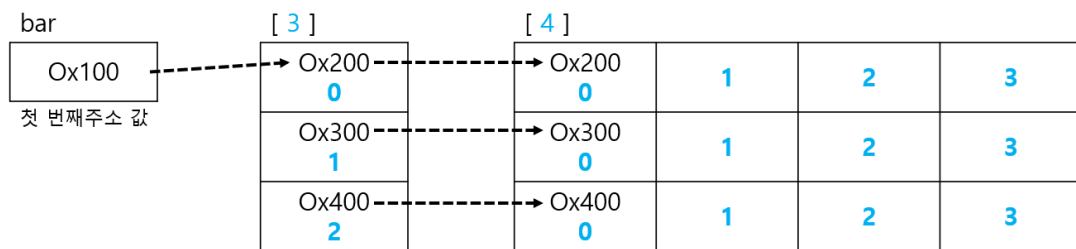
概念

1

3 x 4 Matrix



2



```
// 2차원 , 3차원 , 4차원 , N차원 Array

// 2 X 3
int bar [] [] = new int [] []{
    {1,2,3},
    {4,5,6}
};

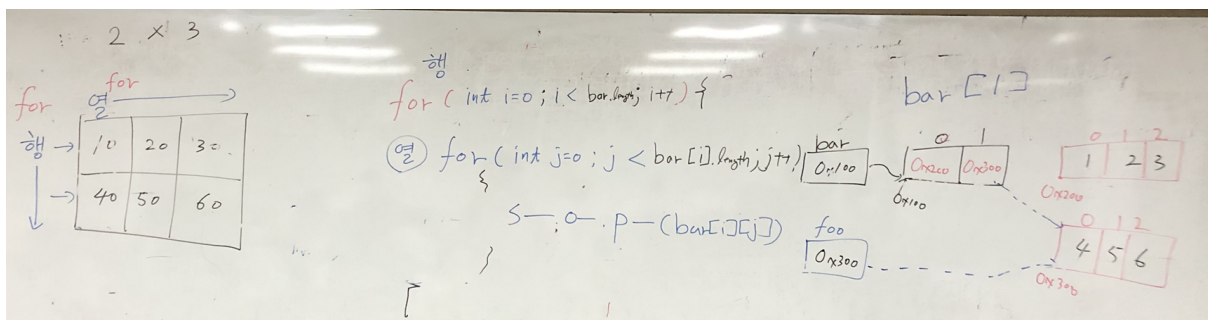
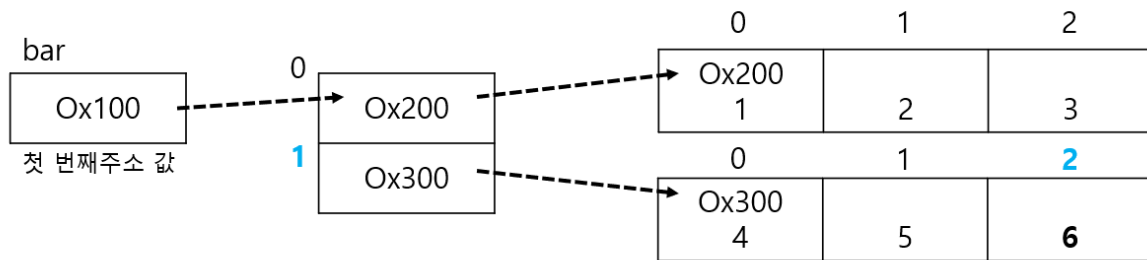
int foo[];

foo =bar[1];

System.out.println(foo[2]);
```

메모리 구조

bar [1] [2] = 6 ;



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

```
/*
  1 | 2 | 3
  -----
  4 | 5 | 6
*/

/* 행 */
for (int index = 0; index < bar.length; index++) {

    /* 열 */
    for (int j = 0; j < bar[index].length; j++) {
        System.out.print(bar[index][j]+ "\t");
    }
    System.out.println();
}
```