SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

삼성 청년 SW 아카데미

Java

SAMSUNG SW ACADEMY FOR YOUTH

Programming Language

Java 기본 문법

- 배열
- 다차원 배열

SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

배열

배열(Array)

배열이란

int score5 = 84

점수가 늘어난다면???

int score1= 99

int score3 = 70

int score4 = 83

int score2= 98

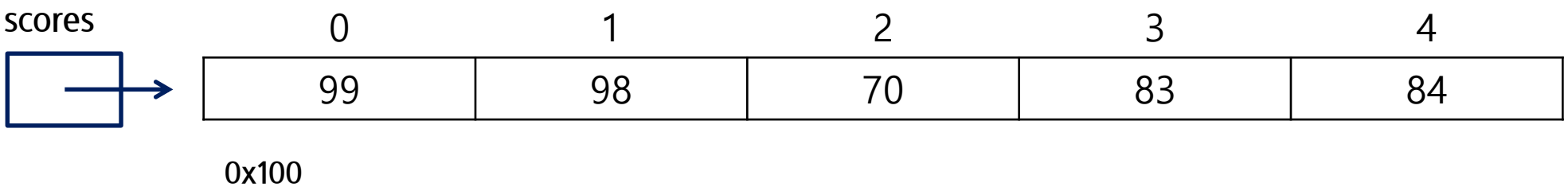
배열(Array)

배열

- 같은 종류의 데이터를 저장하기 위한 자료구조
- 크기가 고정되어 있음. (한번 생성된 배열은 크기를 바꿀 수 없음)
- 배열을 객체로 취급(참조형)
- 배열의 요소를 참조하려면 배열이름과 색인(index)이라고 하는 음이 아닌 정수 값을 조합하여 사용.
- index 번호를 가지고 각 요소에 접근
- index 번호는 0부터 시작
- 배열이름.length를 통해 배열의 길이 조회 가능
- 배열의 길이는 임의로 변경 불가함
- 길이 변경 필요시 새로운 배열을 생성 후 내용을 옮긴다.

배열(Array)

배열



배열(Array)

- 배열의 선언
  - 타입[ ] 변수
  - 타입 변수[ ]

타입	배열이름	선언
int	iArr	int[] iArr;
char	cArr	char[] cArr;
boolean	bArr	boolean[] bArr;
String	strArr	String[] strArr;
Date	dateArr	Date[] dateArr;

배열(Array)

배열의 생성과 초기화

- 자료형[] 배열이름 = new 자료형[길이]; //배열 생성(자료형의 초기값으로 초기화)
- 자료형[] 배열이름 = new 자료형[] {값1, 값2, 값3, 값4}; //배열 생성 및 값 초기화
- 자료형[] 배열이름 = { 값1, 값2, 값3, 값4 }; //선언과 동시에 초기화

자료형	기본값	비고
boolean	false	
char	'₩u0000'	공백문자
byte, short, int	0	
long	0L	
float	0.0f	
double	0.0	
참조형 변수	null	아무것도 참조하지 않음

배열(Array)

Confidential

배열의 메모리 생성과정

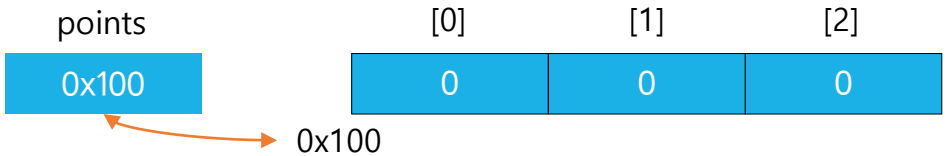
int [] points = new int[3];

배열 선언 : int [] points

배열 생성: new int[3];
메모리에 연속된 공간 차지 → 크기 변경 불가!
Type에 대한 default 초기화

int 타입의 데이터 3개를 담을 수 있는 메모리 공간 확보

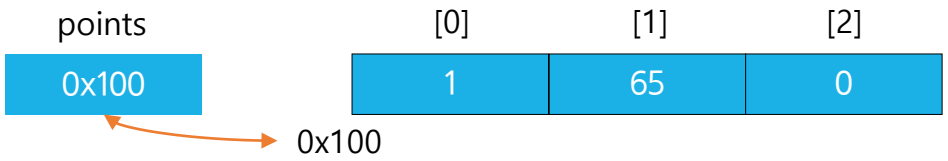
참조 값 할당 : points = new int[3];



배열의 주소를 변수에 할당하여 참조케 함

요소에 값 할당 :

```
points[0] = 1;
points[1] = 'A';
```



배열(Array)

- for-each
  - 가독성이 개선된 반복문으로, 배열 및 Collections 에서 사용
  - index 대신 직접 요소(elements)에 접근하는 변수를 제공
  - naturally ready only (copied value)

```
int intArray [] = { 1, 3, 5, 7, 9 };

for( int x : intArray ){
   System.out.println( x );
}
```

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

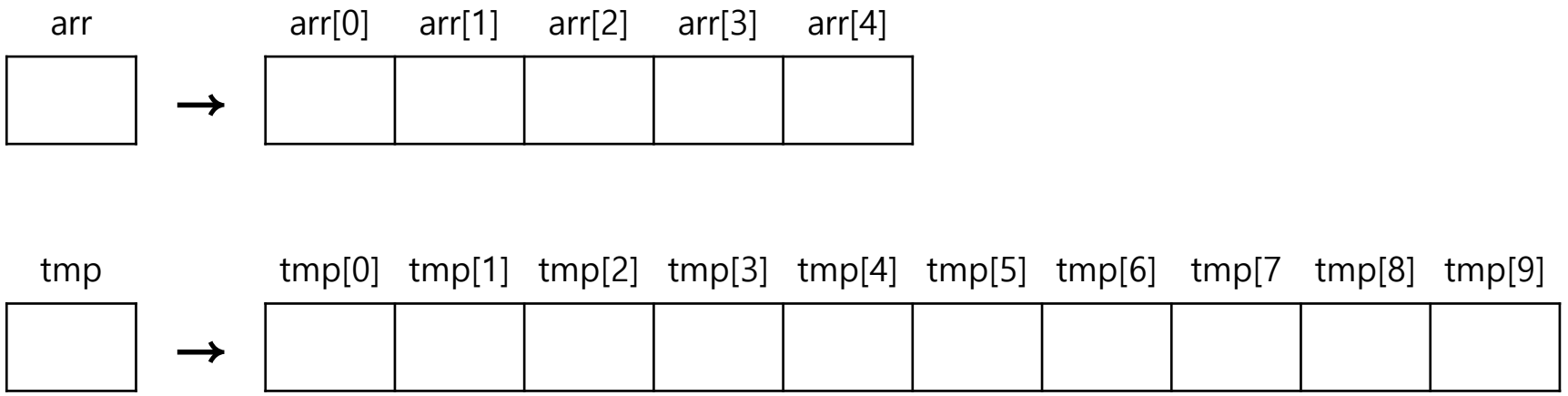
배열(Array)

- 배열의 출력
  - 반복문을 통해서 출력
  - Arrays.toString(배열) : 배열 안의 요소를 [값1, 값2, …] 형태로 출력

배열(Array)

배열의 복사

- 배열은 생성하면 길이를 변경할 수 없기 때문에 더 많은 저장공간이 필요하다면 큰 배열을 생성하고 이전 배열의 값을 복사 해야함.



- 새로운 배열 = Arrays.copyOf(복사하고_싶은_배열, 새로운_배열의 크기)

배열 실습 문제

- 최대값, 최소값 찾기

```
int[] intArray = { 3, 27, 13, 8, 235, 7, 22, 9, 435, 31, 54 };


int min = 1000;
int max = 0;
for(int num: intArray) {
    if(num>max) {
        max = num;
    }
    if(num<min) {
        min = num;
    }
}
System.out.printf("min: %d, max: %d%n", min, max);
```

배열 실습 문제

- 최대값, 최소값 찾기

```
int[] intArray = { 3, 27, 13, 8, 235, 7, 22, 9, 435, 31, 54 };

int min = Integer.MAX_VALUE;
int max = Integer.MIN_VALUE;

for (int num : intArray) {
    min = Math.min(min, num);
    max = Math.max(max, num);
}

System.out.printf("min: %d, max: %d%n", min, max);
```

배열 실습 문제

- 빈도수 구하기

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

int[] used = new int[10];

for(int num:intArray) {
    used[num]++;
}

System.out.println(Arrays.toString(used));
```

SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

다차원 배열

다차원 배열(Multidimensional Array)

- 다차원 배열이란?
  - 2차원 이상의 배열을 의미
  - 배열 요소로 또 다른 배열을 가지는 배열
  - 2차원 배열은 배열 요소로 1차원 배열의 참조를 가지는 배열
  - 3차원 배열은 배열 요소로 2차원 배열의 참조를 가지는 배열

다차원 배열(Multidimensional Array)

- 2차원 배열 선언
  - int[][] iArr
  - int iArr[][]
  - int[] iArr[]

- 2차원 배열 생성
  - 배열의 이름 = new 배열유형[1차원 배열개수][1차원 배열의 크기];
  - 배열의 이름 = new 배열유형[1차원 배열개수][];

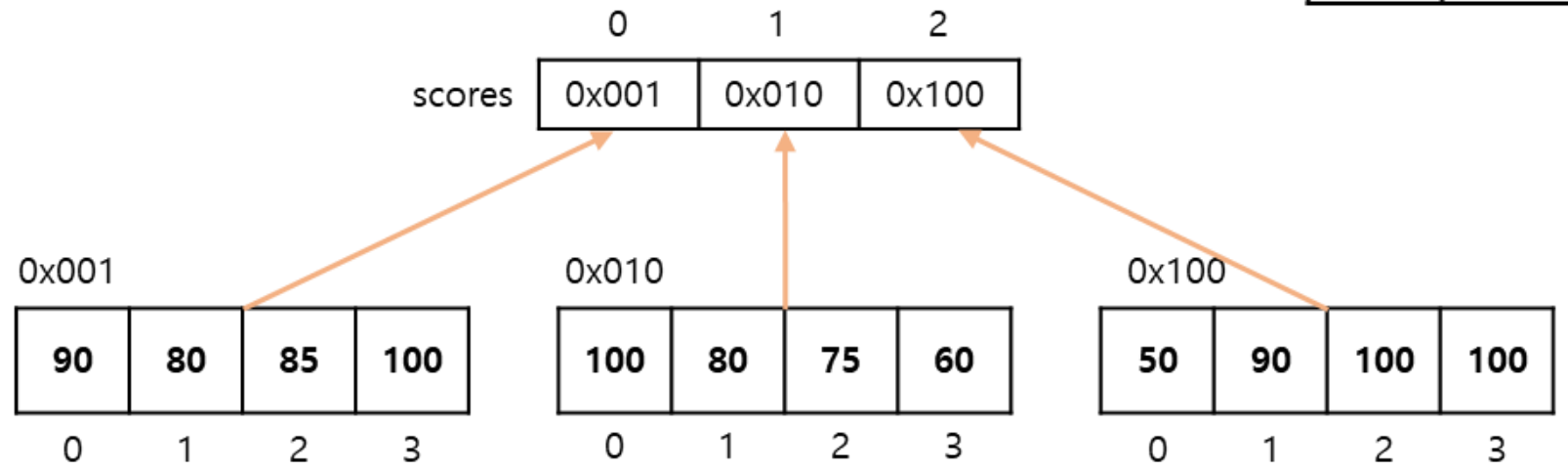
다차원 배열(Multidimensional Array)

2차원 배열

```
int[][] scores = {{90,80,85,100},
                  {100,80,75,60},
                  {50,90,100,100}};
```

scores

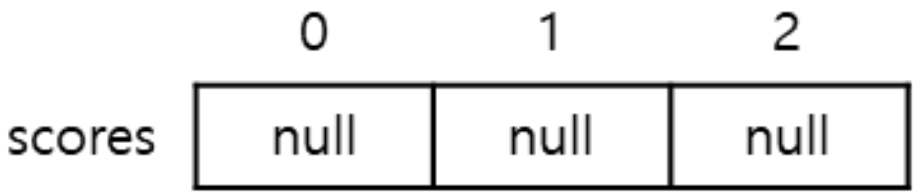
	0	1	2	3
0	90	80	85	100
1	100	80	75	60
2	50	90	100	100



다차원 배열(Multidimensional Array)

2차원 배열

```
int[][] scores = new int[3][];
```



다차원 배열(Multidimensional Array)

2차원 배열 메모리

```
int a = 10;

int [] arr = new int [4];

int [][] arr2 = new int[2][];
arr2[0] = new int [3];
arr2[1] = new int [3];
arr2[1][1] = 100;
```

다차원 배열(Multidimensional Array)

- 2차원 배열 탐색
- 모든 2차원 배열의 원소 중 3의 배수의 개수와 그들의 합을 출력

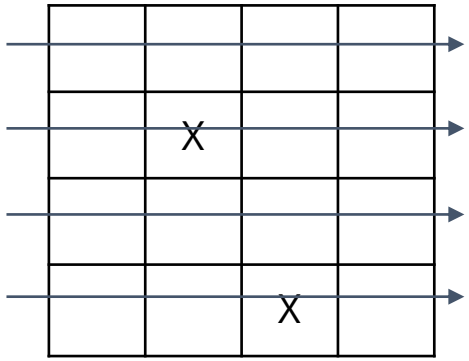
2	3	1	4	7
8	13	3	33	1
7	4	5	80	12
17	9	11	5	4
4	5	91	27	7

```
public static void main(String[] args) {

    int[][] grid = {
            {2, 3, 1, 4, 7},{8, 13, 3, 33, 1},
            {7, 4, 5, 80, 12},{17, 9, 11, 5, 4},
            {4, 5, 91, 27, 7}
    };

    int count = 0;
    int sum = 0;
    for(int [] row: grid) {
        for(int num:row) {
            if(num%3==0) {
                count++;
                sum+=num;
            }
        }
    }
    System.out.printf("개수: %d, 총합: %d%n", count, sum);
}
```

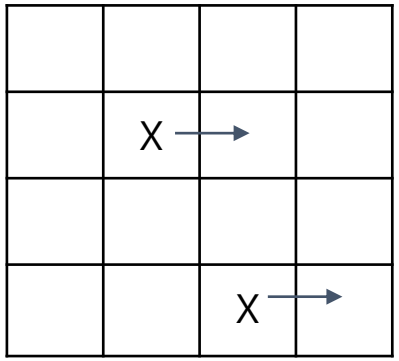
다차원 배열(Multidimensional Array)`

Confidential

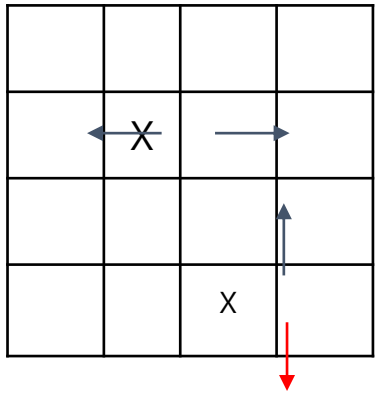
2차원 배열 탐색



전체 중 X를 만나면



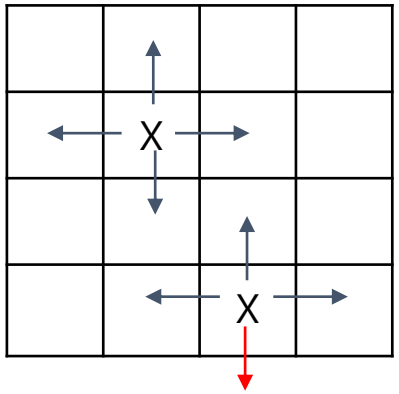
X가 움직이면서



X 주변 탐색

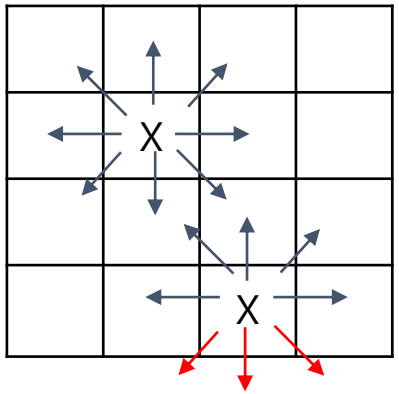
좌/우

상/하



X 주변 탐색

4방



X 주변 탐색

8방

특정 좌표로부터 주변을 탐색하는 경우, 배열의 범위를 벗어나지 않기 위한 코드 필요

SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

다음 방송에서 만나요!

삼성 청년 SW 아카데미