



Java 4강-배열

양 명 속

[now4ever7@gmail.com]



목차

- 배열 (Arrays)이란
- 배열 변수의 복사
- Enhanced for 문
- 로또 만들기 예제
- 반환타입이 배열인 메서드
- 배열을 매개변수로 받는 메서드
- 다차원배열
- 가변배열
- 명령줄 인수

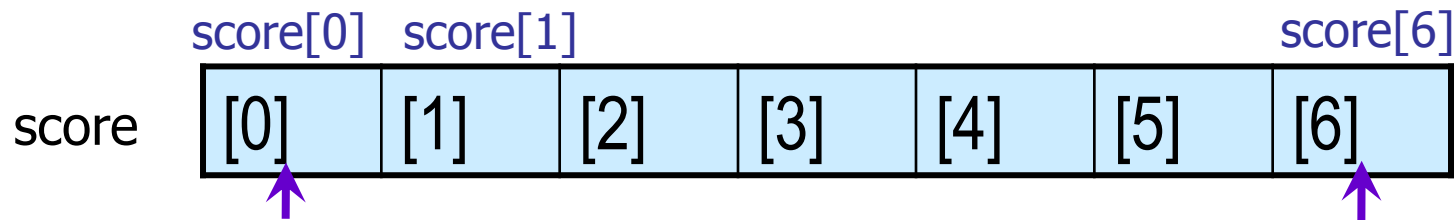
```
int a
```

변수

배열이란?

- 동일한 타입의 변수들을 하나의 이름으로 묶어 놓은 것
 - 같은 타입의 데이터를 여러 개 저장할 수 있는 저장소
 - 참조형 객체로 heap에 할당되어야 사용할 수 있음
 - 배열의 이름은 연속된 변수들을 참조하기 위한 참조값
-
- 여러 개의 변수들을 하나의 배열에 모아 두면 개별 요소를 첨자로 액세스할 수 있으므로 반복적인 처리가 효율적임
 - 배열은 요소들의 집합.
 - 배열의 각 요소들은 모두 동일한 데이터 타입을 가짐
 - 각 요소들은 정수형 **인덱스(index, 첨자)**로 접근 가능

```
int[] score = new int[7]
```



정수 Index 0

정수 Index 6



배열

- 배열의 선언은 둘 이상의 변수 선언에 편의를 제공함
- 배열로 선언된 변수들에는 반복문을 이용해서 동일한 코드 패턴을 적용할 수 있음

배열

- 변수 선언
`int a;`
`a=10;`

■ 배열의 선언

자료형 [] 배열이름;

예) `int[] a;` //생성된 배열을 다루는데 사용될 참조변수 선언

배열의 차원 수

`int a[];` //가능

■ 메모리 할당 (배열 생성)

`a = new int[3];` //3개의 int 값을 저장할 수 있는 공간 생성

//java의 배열은 참조형->new 키워드로
메모리에 인스턴스화 시킨 후 사용

■ 선언과 메모리 할당을 동시에

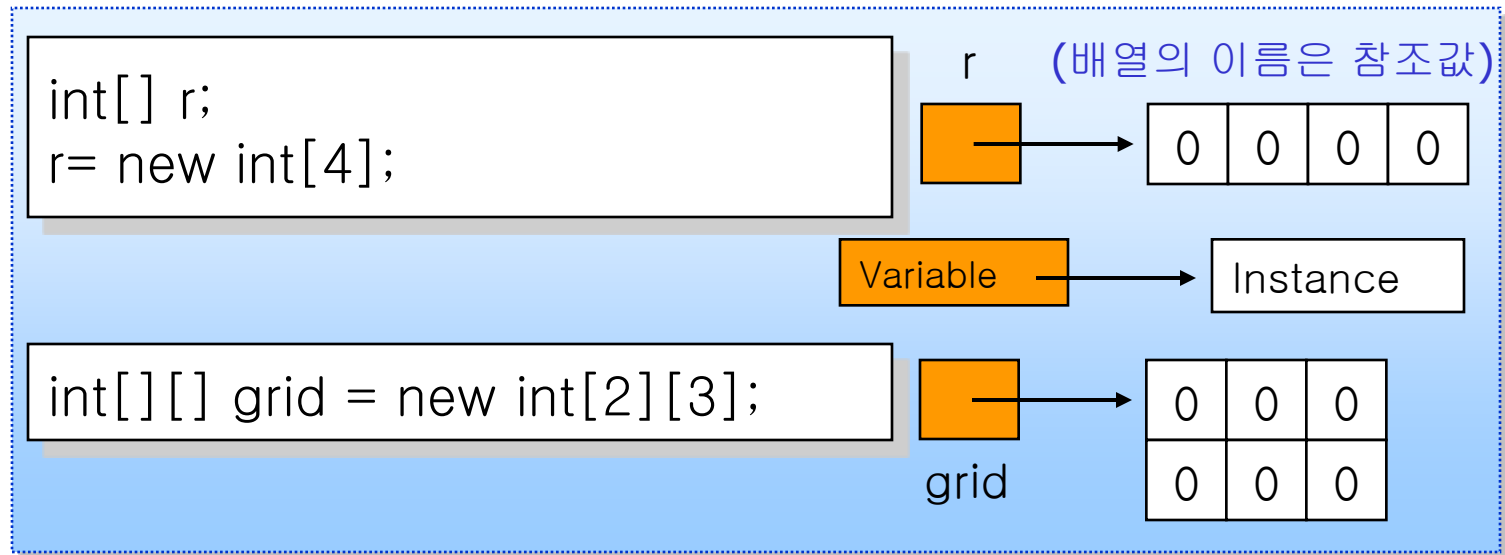
자료형[] 배열이름 = new 자료형[참자];

예) `int[] a = new int[3]`

참조형 - String, Scanner
`String str = new String("hello");`
`Scanner sc = new Scanner(System.in)`

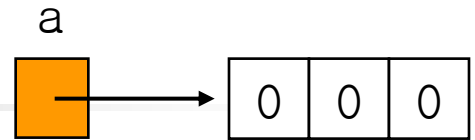
배열의 인스턴스 생성

- 배열을 선언한다고 배열요소가 만들어지는 건 아님
 - 배열의 요소를 생성하려면 new 연산자 사용
 - 배열 요소들은 기본값으로 0으로 초기화



배열

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



■ 배열의 초기화

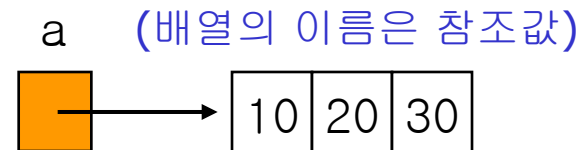
- 힙에 할당되는 배열 요소는 디폴트값으로 초기화됨

- `int[] a = new int[3]`

- `a[0]=10;`

- `a[1]=20;`

- `a[2]=30;`



- 선언과 동시에 초기화하면 배열의 길이를 따로 정해주지 않아도 됨
 - `int[] a = new int[]{10, 20, 30};`
 - `int[] a = {10, 20, 30};`



배열

- 선언과 할당을 따로 하는 경우
 - `int[] a;`
 - `a = new int[]{10, 20, 30};`
 - `a = new int[3]; a = {10, 20, 30};` //에러
- 배열 요소에 접근하기
 - 배열의 이름 뒤에 인덱스 번호 붙여서 사용
 - `int[] a = new int[2];`
 - `a[0] = 10;`
 - `a[1] = 20;`
 - `a[2] = 30;` //에러
 - 배열의 인덱스는 0부터 시작
 - 배열의 길이가 2 이면 인덱스번호는 0~1



배열

10
20
30

```
class ArrayTest1{  
    public static void main(String[] args) {  
        int[] arr = new int[3];  
        arr[0]=10;  
        arr[1]=20;  
        arr[2]=30;  
  
        //int[] arr = {10, 20, 30};  
  
        for(int i=0; i < 3; i++)  
        {  
            System.out.println(arr[i]);  
        }  
    }  
}
```

```
System.out.println(arr[0]);  
System.out.println(arr[1]);  
System.out.println(arr[2]);
```

예제

```
class ArrayTest2
```

```
{  
    public static void main(String[] args)  
    {  
        int[] item = {90,75,70,90,60,85,75,70};  
  
        System.out.println("##### 변경 전 #####");  
        for(int i=0;i<8;i++)  
        {  
            System.out.println(item[i]);  
        }  
  
        System.out.println("##### 변경 후 #####");  
        for(int i=0;i<item.length;i++)  
        {  
            item[i] += 5;  
            System.out.println(item[i]);  
        }  
    }  
}
```

```
----- 변경 전 -----  
90  
75  
70  
90  
60  
85  
75  
70  
----- 변경 후 -----  
95  
80  
75  
95  
65  
90  
80  
75
```

• 배열명.length : 배열의 길이



실습

```
farr[0]=12.5  
farr[1]=-24.3  
farr[2]=0.76
```

- 1. 요소 3개를 갖는 float타입의 배열을 선언하고, 12.5, -24.3, 0.76의 값을 넣고, 화면에 출력하기
- 2. 요소 4개를 갖는 double 배열을 선언하여
 - 임의의 값을 넣고
 - 이 배열 요소들의 총점과 평균을 계산하여 출력하기

```
총점: 344.2, 평균: 86.1
```

for 문안에서 배열 초기화하기

```
import java.util.*;
class ArrayTest4 {
    public static void main(String[] args) {
        int[] arr = new int[5]; //5, 10, 15, 20, 25
        for (int i=0;i<arr.length ;i++ ){
            arr[i]=5*(i+1);
        }

        for (int n : arr ){
            System.out.println(n);
        }

        //사용자로부터 double 값 4개 입력 받아서 배열에 넣기
        Scanner sc = new Scanner(System.in);
        System.out.println("실수를 4번 입력하세요");

        double[] darr=new double[4];
        for (int i=0;i<darr.length ;i++ ){
            darr[i] = sc.nextDouble();
        }

        //확장 for문 이용하여 출력하기
        for (double d : darr){
            System.out.println(d);
        }
    }
}
```

```
//arr[0]=5;
//arr[1]=10;
```

```
/*darr[0] = sc.nextDouble();
darr[1] = sc.nextDouble();
darr[2] = sc.nextDouble();
darr[3] = sc.nextDouble();*/
```



실습

```
원하는 배열의 갯수를 입력하세요 : 4  
arr[0] = 0  
arr[1] = 1  
arr[2] = 2  
arr[3] = 3
```

- [1] 사용자에게 횟수를 입력 받고
그 횟수만큼 배열의 요소를 만들어 값을 자동으로 입력하
고, 화면 출력하기
 - 값은 index값을 넣는다(for문 이용)
 - 예) 4를 입력하면 0,1,2,3 을 배열에 입력
3을 입력하면 0,1,2를 배열에 입력
- [2] 5개의 정수를 요소로 갖는 배열을 만들고
for문에서 1,3,5,7,9 의 값을 배열에 입력한 후, 확장 for문
을 이용해서 배열의 값을 출력한다

10 57 19 35 77 26
최대값 = 77

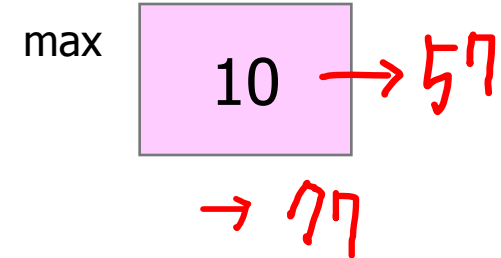
예제

배열에 저장된 최대값 구하기

```
class ArrayTest3{
    public static void main(String[] args) {
        //1. 최대값 구하기
        int[] arr={10, 57, 19, 35, 77, 26};
        int max = arr[0];

        for (int i=1;i<arr.length ;i++ ){
            if(arr[i]>max)
                max=arr[i];
        }

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





Math.round()

• 소수이하 둘째자리에서 반올림해서
소수 이하 1자리만 표시하기
예) 84.362 => 84.4

[1] 84.362*10 => 843.62
[2] Math.round(843.62) => 844
[3] 844/10.0 => 84.4

- static int round(float a)
- static long round(double a)
 - 소수점 첫째 자리에서 반올림한 정수 값을 반환
 - long l = Math.round(5.55); //6
 - long l2 = Math.round(5.11); //5
 - long l3 = Math.round(-5.55); //-6
 - long l4 = Math.round(-5.11); //-5
 - 소수점 셋째 자리에서 반올림하려면?
 - double d = 90.7552;
 - double d2 = Math.round(d*100)/100.0; //90.76



계산된 크기의 배열 선언

- 컴파일 시 배열 크기 지정
 - 컴파일시에 미리 배열의 크기를 지정해 놓는 것이 다른 방법보다 속도가 향상됨

```
long[] row = new long[4];
```

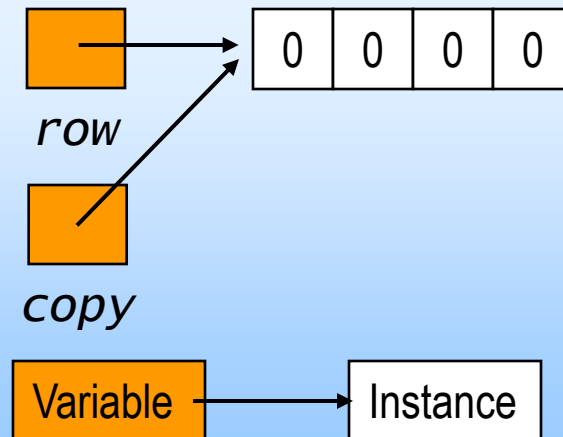
- 런타임 시 배열 크기 지정

```
Scanner sc = new Scanner(System.in);  
int size = sc.nextInt();  
long[ ] row = new long[size];
```


배열 변수의 복사

- 배열 변수의 복사는 배열 변수 자체만 복사
 - 배열의 인스턴스는 복사되지 않음
 - 2개의 배열 변수는 동일한 배열 인스턴스를 참조
 - 배열은 참조형이므로 **배열 복사**하게 되면 **같은 메모리 영역을 가리키게 됨**

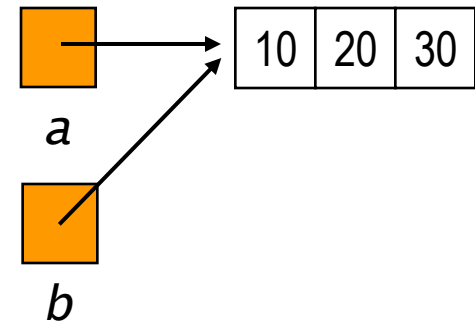
```
int[ ] row = new int[4];  
int[ ] copy = row;  
...  
row[0]++;  
System.out.println(copy[0]);
```



배열 복사

```
a[0] : 11  
b[0] : 11
```

```
class ArrayCopy  
{  
    public static void main(String[] args)  
    {  
        int[] a = new int[]{10,20,30};  
        int[] b = a;  
  
        a[0]++;  
  
        System.out.println("a[0] : "+ a[0]);  
        System.out.println("b[0] : "+ b[0]); //a[0]를 증가시키면 b[0]도 증  
가됨  
    }  
}
```



Enhanced for 문

```
int[] number= {10, 20, 30, 40};  
for (int i=0; i<number.length;i++)  
{  
    System.out.println(number[i]);  
}
```

- 일정한 개수로 구성된 특정 집합의 요소들을 반복 처리할 때 사용됨
- 배열의 모든 요소를 순회할 때가 가장 전형적임
- 배열 또는 컬렉션(해시, 리스트 등)에 데이터가 있는 만큼 반복

```
for (타입변수 : 배열)  
{  
    명령문;  
}
```

```
int[] number= {10, 20, 30, 40};  
  
for (int n : number)  
{  
    System.out.println(n);  
}
```

제어변수 **n** 은 읽기 전용
반복문 내에서 **n**의 값을 변경할 수 없음

10 20 30 40



예제 for

```
class ForeachTest
{
    public static void main(String[] args)
    {
        int[] num= {12, 43, 64, 56, 32};

        for (int n : num)
        {
            System.out.println(n);
        }
    } //main
} //class
```

```
12
43
64
56
32
```

```
String[] hobby= {"movie", "reading", "sports"};

for (String s : hobby)
{
    System.out.println(s);
}
```

```
movie
reading
sports
```



예제-컬렉션

```
import java.util.*;
```

```
ArrayList<Integer> list = new ArrayList<Integer>();  
list.add(10);  
list.add(25);  
list.add(32);
```

```
for(int n : list)  
{  
    System.out.println(n);  
}
```

```
10  
25  
32
```

```
import java.util.*;
```

```
ArrayList<Integer> list = new ArrayList<Integer>();  
for(int i = 0 ; i < 5; i++)  
{  
    list.add(i);  
}  
for(int n : list)  
{  
    System.out.println(n);  
}
```

```
0  
1  
2  
3  
4
```



Arrays.sort() 메서드

```
import java.util.*;
```

```
class ArrayMethodTest  
{
```

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

```
        int[] arr = {10, 5, 44, 55, 1000, 140, 3, 55};
```

```
        System.out.println( "Before sorting:" );  
        for (int i : arr) System.out.print(i+"Wt ");  
        System.out.println("Wn");
```

```
        Arrays.sort(arr); // 오름차순 정렬  
        System.out.println( "After sorting:" );  
        for (int i : arr) System.out.print(i+"Wt ");  
        System.out.println("Wn");
```

```
    }
```

```
}
```

```
Before sorting:
```

```
10      5      44      55      1000      140      3      55
```

```
After sorting:
```

```
3        5      10      44      55      55      140      1000
```



예제-로또 만들기

- 배열 이용, 길이를 6으로
- 1~45 중에서 6개 숫자 뽑기
- 중복되지 않도록

//랜덤값을 뽑는 메서드를 만듦

```
int temp=(int)(Math.random()*45+1); //1~45 중에서 뽑는다
```

Math.random()

– Returns a double value with a positive sign,
greater than or equal to 0.0 and less than 1.0.

$0.0 \leq x < 1.0$

예제-로또 만들기

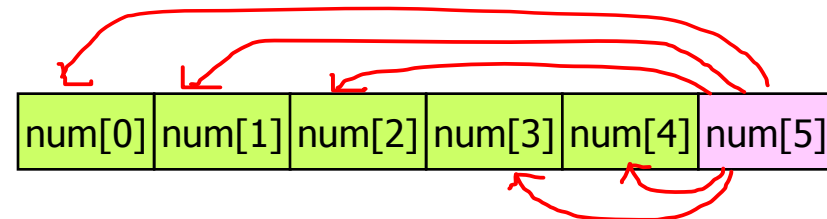
```
import java.util.*;
class LottoTest1
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        int[] num = new int[6];
        while(true)
        {
            for (int i=0;i<6;i++ )
            {
                //랜덤 뽑는다
                int temp=(int)(Math.random()*45+1);
                num[i]=temp;

                //중복데이터 검사
                for(int k=0; k < i; k++){
                    if(num[k]== num[i]){ i--;break;}
                }
            } //for
            //정렬
            Arrays.sort(num);
```

	k	i	temp
i=1, k=0			num[0]==num[1]
i=2, k=0			num[0]==num[2]
	k=1		num[1]==num[2]
....			
i=5, k=0			num[0]==num[5]
	k=1		num[1]==num[5]
	k=2		num[2]==num[5]
	k=3		num[3]==num[5]
	k=4		num[4]==num[5]

```

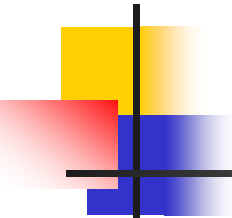
7      8      9      14      24      40
=====
계속 하시겠습니까?(Y/N)
y
9      13      21      24      37      40
=====
계속 하시겠습니까?(Y/N)
y
7      24      30      33      38      39
=====
계속 하시겠습니까?(Y/N)
n
계속하려면 아무 키나 누르십시오 . . .
```





```
//출력
for(int i=0;i<num.length;i++)
{
    System.out.print(num[i]+"Wt");
}
}
//for
System.out.println("Wn=====");

System.out.println("Wn계속 하시겠습니까?(Y/N)");
String s = sc.nextLine();
if (s.equalsIgnoreCase("N")) break;
```



예제-로또 만들기(반환형이 배열인 메서드로 변환)

```
import java.util.*;
class LottoTest2{
    public static void main(String[] args){
        Scanner sc = new Scanner(System.in);
        int[] num;
        while(true)
        {
            num = lotto();

            //정렬
            Arrays.sort(num);

            //출력
            for(int i=0;i<num.length;i++)
            {
                System.out.print(num[i]+"Wt");
            }
            System.out.print("\n");

            System.out.print("계속 하시겠습니까?(Y/N) : ");
            String quit = sc.nextLine();
            if (quit.toUpperCase().equals("N")) break;

        }
    }
}
```



예제-로또 만들기

```
public static int[] lotto()
{
    int[] num = new int[6];

    for (int i=0;i<6;i++ )
    {
        //랜덤 뽑는다
        int temp=(int)(Math.random()*45+1);
        num[i]=temp;

        //중복데이터 검사
        for(int j=0; j < i; j++){
            if(num[j]== temp){ i--; break;}
        }
    } //for
    return num;
} //lotto
}
```

배열을 매개변수로 받는 메서드

```
import java.util.*;
class ArrayArg
{
    //배열을 매개변수로 받는 메서드
    public static void findAvg(int[] score)
    {
        //국, 영, 수 점수를 배열로 받아오자
        //총점구하기
        int sum=0;
        for (int i=0;i<score.length ;i++ )
        {
            sum+=score[i];
        }

        //평균 구하기
        int avg = sum/score.length;

        System.out.println("총점:"+sum +", 평균:"+avg);

        score[0]+=5;
    }
}
```

```
국어, 영어, 수학 점수를 입력하세요
90
80
70
총점:240, 평균:80
subject[0] : 95
```

```

public static void main(String[] args)
{
    Scanner sc = new Scanner(System.in);
    System.out.println("국어, 영어, 수학 점수를 입력하세요");
    int[] subject = new int[3];
    for (int i=0;i<subject.length ;i++ )
    {
        subject[i]=sc.nextInt();
    }

    //메서드 호출
    findAvg(subject); //call by reference - 매개변수로 값이 넘어 가는 게 아니라
    //주소가 넘어감 => 같은 주소를 가지므로, 같은 곳을 가리키게 됨

    System.out.println("subject[0] : " + subject[0]);
    //=> 메서드에서 매개변수의 값을 변경하면
    //main (메서드를 호출한 곳)에서도 변수의 값이 변경됨

    //call by value - 값을 매개변수로 넘김
}
}

```

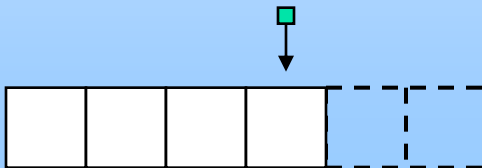
배열의 Rank(차원 수)

■ 배열의 종류

- n열의 요소로 구성된 1차원 배열
- n열의 요소가 m행으로 구성된 2차원 배열

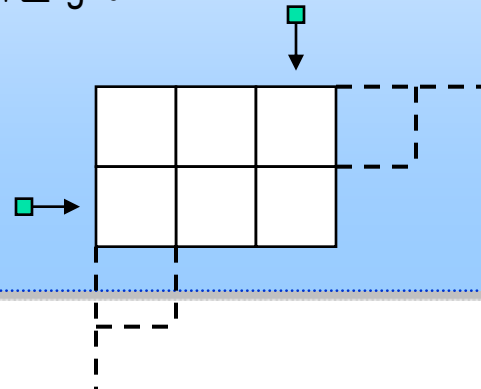
```
long[] row;
```

long형 데이터를 가지는 1차원 배열 row



```
int[][] grid;
```

int형 데이터를 가지는 2차원 배열 grid



다차원배열

■ 2차원 배열

■ 선언

- `int[][] a;`
- `int a[][];`
- `int[] a[];`

■ 메모리 할당

- `a = new int[2][3];` //2행 3열의 2차원 배열

■ 선언과 할당을 동시에

- `int[][] a = new int[2][3];`

■ 선언, 메모리할당, 초기화 모두 동시에

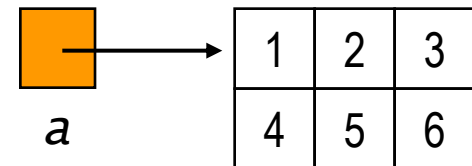
- `int[][] a = new int[][]{{1,2,3},{4,5,6}};`
- `int[][] a = {{1,2,3},{4,5,6}};`

■ cf) 동적배열

- `int[][] a = new int[2][];` //동적 배열

• 초기화

```
a[0][0] = 1;  
a[0][1] = 2;  
a[0][2] = 3;  
a[1][0] = 4;  
a[1][1] = 5;  
a[1][2] = 6;
```

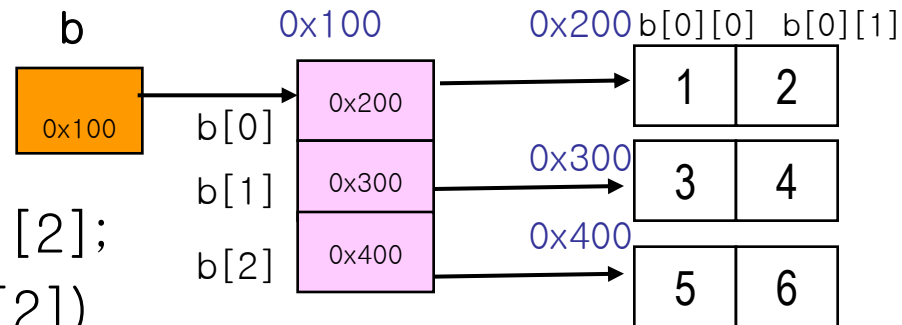


다차원배열

- 자바의 다차원 배열에서 각 영역은 모두 reference 를 표시하는 공간이고, 마지막 배열 공간만이 실제 데이터를 나타내는 공간임
 - 예) `byte[][] b = new byte[3][2];`
 - 레퍼런스형의 공간은 b 와 배열의 첫번째에 있는 [3]의 공간임

- **length 프로퍼티**

- 배열의 길이를 얻어옴
- 예) `byte[][] b = new byte[3][2];`
- `b.length => 3` (`b[0]`, `b[1]`, `b[2]`)
- `b[0].length => 2` (`b[0][0]`, `b[0][1]`)
- `b[1].length => 2` (`b[1][0]`, `b[1][1]`)

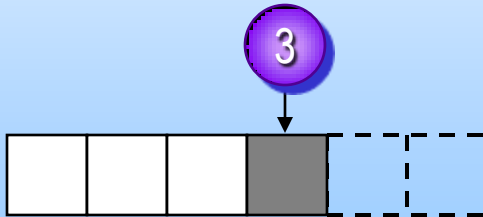


자바의 2차원 배열은 "배열의 배열"로 구성되어 있음

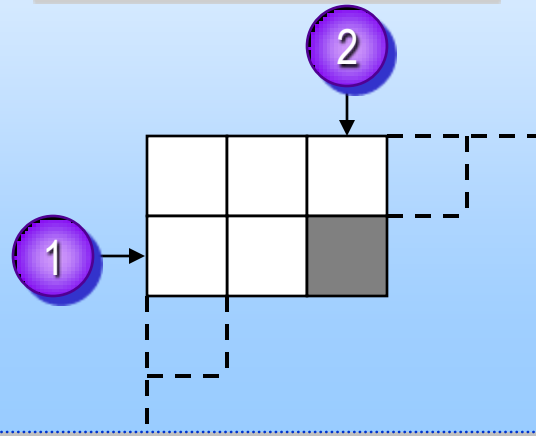
배열 요소에 접근하기

- 각 rank에 정수형 인덱스를 지정함
 - 인덱스는 0부터 시작.

```
long[] row;  
...  
row[3];
```

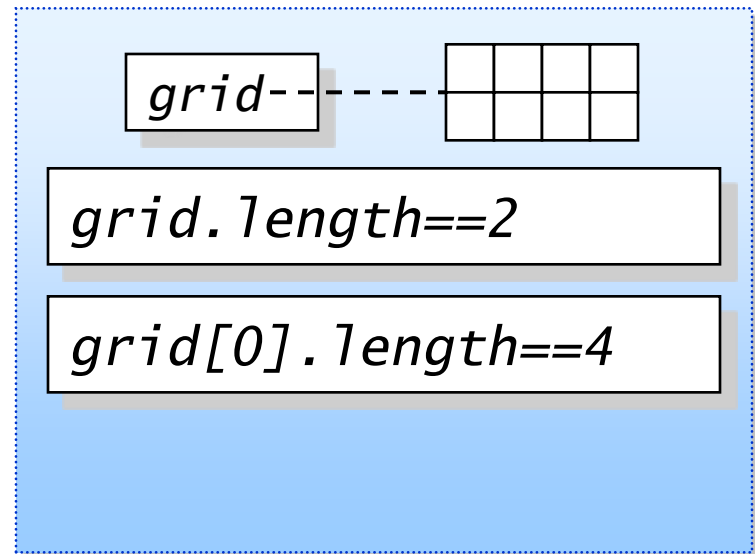
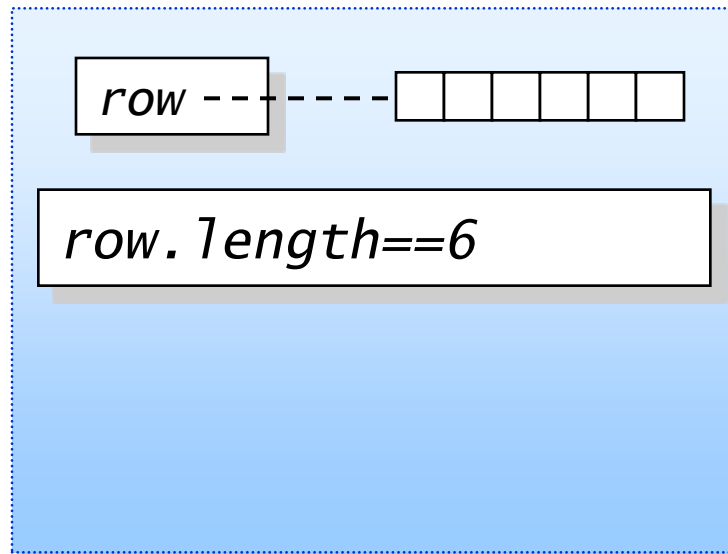


```
int[][] grid;  
...  
grid[1][2];
```



배열의 경계 체크하기

- 모든 배열의 경계를 체크 가능
 - 배열의 첨자가 배열 범위를 벗어나면 예외발생
 - `ArrayIndexOutOfBoundsException`
 - `length` 속성 사용





예제

//1.배열 선언

```
int[][] arr;
```

//2. 메모리 할당

```
arr = new int[2][3]; //2행3열
```

//3. 초기화

```
arr[0][0]=10;
```

```
arr[0][1]=20;
```

```
arr[0][2]=30;
```

```
arr[1][0]=15;
```

```
arr[1][1]=17;
```

```
arr[1][2]=14;
```

//[2] 배열선언과 메모리 할당을 동시에

```
int[][] arr3=new int[3][2]; //3행 2열
```

//초기화

```
arr3[0][0]=4;
```

```
arr3[0][1]=3;
```

```
arr3[1][0]=7;
```

```
arr3[1][1]=5;
```

```
arr3[2][0]=1;
```

```
arr3[2][1]=2;
```



예제

//[3] 배열선언, 메모리 할당, 초기화를 모두 동시에

```
int[][] arr4 = {{1, 2, 3, 4},{5, 6, 7, 8},{9, 10, 11, 12}}; //3행 4열
```

//배열의 값들 출력하기

```
for (int i=0;i<arr.length ;i++ ) //행
```

```
{
```

```
    for (int k=0;k<arr[i].length ;k++ ) //열
```

```
{
```

```
        System.out.println("arr["+i+"]["+k+"] : "+arr[i][k]);
```

```
    }//안쪽 for
```

```
}//바깥 for
```

```
System.out.println("행의 개수 : " + arr.length);
```

```
System.out.println("열의 개수 : " + arr[0].length);
```



예제

```
class ArrayTest4
```

```
{
```

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

```
    {
```

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

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

```
        for(int i=0; i < arr.length; i++) //2행
```

```
        {
```

```
            for(int j = 0; j < arr[i].length; j++) //3열
```

```
            {
```

```
                System.out.println "[" + i + ", " + j + "] = " + arr[i][j]);
```

```
            }
```

```
        }
```

```
        System.out.println("arr.length : " + arr.length);
```

```
        System.out.println("arr[0].length : " + arr[0].length);
```

```
    }
```

```
}
```

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

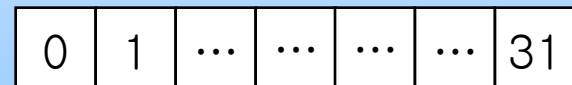
가변배열(Jagged Array)

- 배열의 배열(Jagged Arrays) 들쭉날쭉 배열, 동적 배열
 - 배열의 요소가 배열
 - 배열의 요소가 일정하지 않은 배열
 - 배열 요소마다 차원과 크기를 다르게 지정할 수 있음
 - 동적으로 메모리를 할당할 수 있는 방식
 - 예) 달력의 12개월 각각의 날짜수가 다름,
사람 이름을 가나다순 배열로 정의

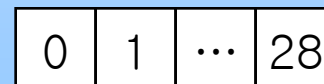
```
int[][] daysInMonth;  
daysInMonth = new int[12][];  
daysInMonth[0] = new int[31];  
daysInMonth[1] = new int[28];  
...  
daysInMonth[11] = new int[31];
```

각 행마다 다른 크기의 배열을
생성하는 것이 가능

daysInMonth[0]



daysInMonth[1]



가변배열(Jagged Array)

- 가변배열 선언

```
자료형[][] 배열명 = new 자료형[크기][];  
예) int[][] a = new int[3][];
```

- 배열의 길이 정의

- `a[0] = new int[4];`
- `a[1] = new int[2];`
- `a[2] = new int[3];`

- 초기화

- `a[0] = new int[]{4,2,8,9};`
- `a[1] = new int[]{3,7};`
- `a[2] = new int[]{1,8,5};`

- 선언과 동시에 초기화

```
int[][] a = {  
    new int[]{4,2,8,9},  
    new int[]{3,7},  
    new int[]{1,8,5}  
};
```

```
int[][] arr = {  
    {4,2,8,9},  
    {3,7},  
    {1,8,5}  
};
```

예제 1

1	2
3	4
5	6

arr2

1	2	3	4		
5	6				
7	8	9	10	11	12

jArr

2
10
2

```
class JaggedArrayTest
{
    public static void main(String[] args)
    {
        int[][] arr2 = new int[][]{ { 1, 2 }, { 3, 4 }, { 5, 6 } }; //2차원배열
        int[][] jArr = new int[3][]; //가변배열
        jArr[0] = new int[] { 1, 2, 3, 4 };
        jArr[1] = new int[] { 5, 6 };
        jArr[2] = new int[] { 7, 8, 9, 10, 11, 12 };

        System.out.println(jArr[0][1]); //2
        System.out.println(jArr[2][3]); //10
        System.out.println(arr2[0][1]); //2
    }
}
```


예제2

```
class JaggedArrayTest2{
    public static void main(String[] args)
    {
        int[][] arr = new int[3][];

        arr[0] = new int[] {4, 2, 8, 9};
        arr[1] = new int[] {2, 3};
        arr[2] = new int[] {5, 2, 7};

        for (int i=0; i < arr.length; i++)
        {
            System.out.print("(" + i + ")번째 배열의 요소 : ");

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

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

```
<0>번째 배열의 요소 : 4 2 8 9
<1>번째 배열의 요소 : 2 3
<2>번째 배열의 요소 : 5 2 7
```

명령줄 인수(Command-Line Arguments)

- 커맨드라인을 통해 입력 받기
 - 프로그램을 실행할 때 클래스이름 뒤에 공백문자로 구분하여 여러 개의 문자열을 프로그램에 전달 할 수 있음
 - 예) d:>java MainTest **hong 123**
 - 커맨드라인을 통해 입력된 두 문자열은 String배열에 담겨서 MainTest 클래스의 main메서드의 매개변수(args)에 전달됨
 - main 메서드 내에서 args[0], args[1]과 같은 방식으로 전달받은 문자열에 접근할 수 있음

```
class MainTest {  
    public static void main(String[] args) {  
        System.out.println("매개변수의 개수 :" + args.length);  
        for (int i = 0; i < args.length; i++) {  
            System.out.println(args[i]);  
        }  
    }  
}
```

```
매개변수의 개수 :2  
hong  
123
```

실습

복권판매점 번호 1 => 복권판매점배열[0] += 판매량
복권판매점 번호 2 => 복권판매점배열[1] += 판매량
....
복권판매점 번호 8 => 복권판매점배열[7] += 판매량

- 복권 판매점별 판매량 구하기
 - 8개의 복권 판매점에서 복권 티켓을 판매
 - 8개 복권 판매점별로 판매량의 합계를 구하기 위해, 판매량이 각각 독립적으로 누적되어야 함

```
복권판매점 번호<1~8>와 복권 판매량을 입력하세요.<끝낼때는 q>
3
25
그룹번호 : 3, 판매량:25
복권판매점 번호<1~8>와 복권 판매량을 입력하세요.<끝낼때는 q>
5
30
그룹번호 : 5, 판매량:30
복권판매점 번호<1~8>와 복권 판매량을 입력하세요.<끝낼때는 q>
3
10
그룹번호 : 3, 판매량:35
복권판매점 번호<1~8>와 복권 판매량을 입력하세요.<끝낼때는 q>
5
20
그룹번호 : 5, 판매량:50
복권판매점 번호<1~8>와 복권 판매량을 입력하세요.<끝낼때는 q>
q
계속하려면 아무 키나 누르십시오 . . .
```



과제 1

```
임의의 문자열을 입력하세요  
JAVA  
str.charAt(0):J=>74  
str.charAt(1):A=>65  
str.charAt(2):U=>86  
str.charAt(3):A=>65
```

- 1. 사용자로부터 문자열을 입력 받고, 각 문자열에 해당하는 코드값을 출력하기
 - String 클래스의 charAt, length 메서드를 이용 - API문서 찾기
 - public char **charAt**(int index)
 - 문자열 중에서 index 번째 위치에 있는 문자를 반환함
 - index의 값은 배열처럼 0부터 시작함
 - public int **length**()

10	30	40	20	50
----	----	----	----	----

과제2

- 5개의 숫자를 사용자로부터 입력 받아서 배열에 넣고, 내림차순으로 정렬한 후 출력하기
- 예) 10,30,40,20,50 => 50,40,30,20,10

```
5개의 숫자를 입력하세요
5
17
69
43
55
===내림차순으로 정렬===
69      55      43      17      5
```

2번째 배열(num[1])의 값부터 앞의 값들과 차례로 비교하여 더 큰 값을 앞의 배열로 이동한다 (swap)

```
10,30,40,20,50
1) 30,10,40,20,50
2) 40,10,30
   40,30,10,20,50
3) 40,30,20,10,50
4) 50,30,20,10,40
   50,40,20,10,30
   50,40,30,10,20
   50,40,30,20,10
```

실습

```
season[0]=봄  
season[1]=여름  
season[2]=가을  
season[3]=겨울
```

- 1. 봄, 여름, 가을, 겨울을 요소로 갖는 배열 선언하기
 - 배열 선언 후 값을 초기화하고, for문을 이용하여 화면 출력하기
 - 배열선언과 초기화를 따로 할 것
- 2. 3행 2열의 double 타입 2차원 배열을 선언하고, 값을 초기화한 후, 중첩 for문을 이용하여 화면 출력하기

10.1	10.2
10.3	10.4
10.5	10.6

```
dArr[0][0]=10.1  
dArr[0][1]=10.2  
dArr[1][0]=10.3  
dArr[1][1]=10.4  
dArr[2][0]=10.5  
dArr[2][1]=10.6
```

- 3. 2차원 배열을 1차원 배열로 바꾸기
 - 위의 2차원 배열을 1차원 배열로 바꿔서 값을 넣어준 후, 화면 출력하기

[row][column] => [column의 크기*row+column]

10.1	10.2	10.3	10.4	10.5	10.6
------	------	------	------	------	------

```
10.1 10.2 10.3 10.4 10.5 10.6
```

실습

$[row][column] \Rightarrow [column의 크기 * row + column]$

column

row

	0	1
0	10.1	10.2
1	10.3	10.4
2	10.5	10.6

3행 2열
column의 크기 : 2

$2*0+0,$ $2*0+1,$ $2*1+0,$ $2*1+1,$ $2*2+0,$ $2*2+1$

0	1	2	3	4	5
10.1	10.2	10.3	10.4	10.5	10.6

0, 0 0, 1 1, 0 1, 1 2, 0 2, 1

실습

- 길이가 5인 double 형 배열을 선언하여 0보다 큰 실수 5개를 입력 받아서 저장하고, 이렇게 저장된 값을 은행 고객의 잔고라고 가정하자.
 - (5명의 고객 잔고를 배열에 저장한 것)
 - 사용자로부터 이자율을 입력 받고, 배열에 저장된 모든 값을 이자율 만큼 씩 증가시키고, 증가된 결과를 출력한다
 - 예) 이자율이 3.3이 입력되면, 이율이 3.3% 라는 뜻으로 해석되어 배열에 저장된 모든 값을 3.3%씩 증가시킨다

잔고를 입력하세요

100000

잔고를 입력하세요

200000

잔고를 입력하세요

300000

잔고를 입력하세요

400000

잔고를 입력하세요

500000

이자율을 입력하세요

2.5

이자가 반영된 잔고 조회 : **102500**

이자가 반영된 잔고 조회 : **205000**

이자가 반영된 잔고 조회 : **307500**

이자가 반영된 잔고 조회 : **410000**

이자가 반영된 잔고 조회 : **512500**

Math.round() 이용하여 반올림한다

실습

- 가로, 세로의 길이가 3인 단위 행렬을 만들고 출력하기
 - 단위 행렬 - 대각선의 위치에 놓인 요소들이 1 이고, 나머지 요소들이 0으로 이루어진 행렬

1	0	0
0	1	0
0	0	1

1	0	1
0	1	0
1	0	1

$i=0, j=2$

$i=1, j=1$

$i=2, j=0$



실습

- 4행 5열의 2차원 배열을 선언하고, 임의의 값으로 초기화한 후, 배열에 담긴 모든 값의 총합과 평균을 구하는 프로그램을 작성하시오.

```
10 10 10 10 10
20 20 20 20 20
30 30 30 30 30
45 45 45 45 45
```

```
총합 : 525
평균 : 26.25
```

실습

- 다음은 거스름돈을 몇 개의 동전으로 지불할 수 있는지를 계산하는 문제이다. 변수 money의 금액을 동전으로 바꾸었을 때 각각 몇 개의 동전이 필요한지 계산해서 출력하라. 단, 가능한 한 적은 수의 동전으로 거슬러 주어야 한다. (1)에 알맞은 코드를 넣어서 프로그램을 완성하시오.
- [Hint] 나눗셈 연산자와 나머지 연산자를 사용해야 한다.

```
class Exchange
{
    public static void main(String[] args)
    {
        // 큰 금액의 동전을 우선적으로 거슬러 줘야 한다.
        int[] coinUnit = {500, 100, 50, 10};

        int money = 3870;
        System.out.println("money="+money+"\n");

        for(int i=0;i<coinUnit.length;i++) {
            (1)
        }
    }
}
```

money=3870

500원: 7
100원: 3
50원: 1
10원: 2



실습

- 다음은 알파벳과 숫자를 아래에 주어진 암호표로 암호화하는 프로그램이다.
- (1)에 알맞은 코드를 넣어서 완성하시오.

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t
`	~	!	@	#	\$	%	^	&	*	()	-	_	+	=		[]	{

u	v	w	x	y	z
}	;	:	,	.	/

0	1	2	3	4	5	6	7	8	9
q	w	e	r	t	y	u	i	o	p



실습

```
src:abc123  
result: '~!wer
```

```
class Encrypt {  
    public static void main(String[] args) {  
        char[] abcCode =  
            { '\'', '~', '!', '@', '#', '$', '%', '^', '&', '*',  
              '(', ')', '-', '_', '+', '=', '|', '[', ']', '{',  
              '}', ':', ';', ',', '.', '/', ' ' };  
            // 0 1 2 3 4 5 6 7 8 9  
        char[] numCode = {'q', 'w', 'e', 'r', 't', 'y', 'u', 'i', 'o', 'p'};  
  
        String src = "abc123";  
        String result = "";  
  
        // 문자열 src의 문자를 charAt()으로 하나씩 읽어서 변환 후 result에 저장  
        for(int i=0; i < src.length(); i++) {  
            char ch = src.charAt(i);  
            (1)  
        }  
  
        System.out.println("src:"+src);  
        System.out.println("result:"+result);  
    }  
}
```



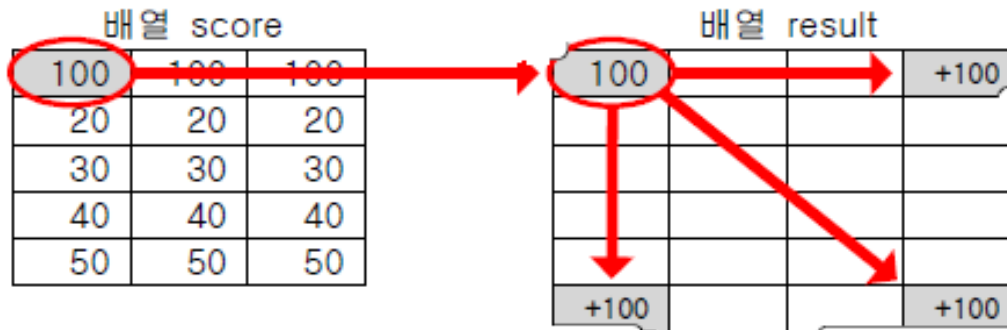
실습

- 총 열명의 수강생 점수를 입력 받아서 그 중에서 최고 점수를 얻은 두 명에게만 A학점을 준다고 할 때, A 학점의 기준이 되는 점수를 계산하여 출력하는 프로그램 작성하기

```
점수를 입력하세요
90
점수를 입력하세요
80
점수를 입력하세요
50
점수를 입력하세요
60
점수를 입력하세요
70
점수를 입력하세요
88
점수를 입력하세요
95
점수를 입력하세요
100
점수를 입력하세요
77
점수를 입력하세요
89
A학점은 95점 이상이 되어야 합니다.
```

실습

- 주어진 2차원 배열의 데이터보다 가로와 세로로 1이 더 큰 배열을 생성해서 배열의 행과 열의 마지막 요소에 각 열과 행의 총합을 저장하고 출력하는 프로그램이다. (1)에 알맞은 코드를 넣어서 완성하시오.



```

class ArraySum2{
    public static void main(String[] args){
        int[][] score = {
            {100, 100, 100}
            , {20, 20, 20}
            , {30, 30, 30}
            , {40, 40, 40}
            , {50, 50, 50}
        };
        //5행 3열 => 6행 4열
        int[][] result = new int[score.length+1][score[0].length+1];

        for(int i=0; i < score.length;i++) {
            for(int j=0; j < score[i].length;j++) {
                (1)
            }
        }//for

        for(int i=0; i < result.length;i++) {
            for(int j=0; j < result[i].length;j++) {
                System.out.print(result[i][j]+"Wt");
            }
            System.out.println();
        }//for
    } // main
}

```

100	100	100	300
20	20	20	60
30	30	30	90
40	40	40	120
50	50	50	150
240	240	240	720