

5장. 배열(Array)



대용량 자료 처리



배열(Array)

● 배열은 왜 써야 할까?

- 정수 20개를 이용한 프로그램을 할 때 20개의 정수 타입의 변수를 선언

int num1, int num2, int num3... num20;



비효율적이고 관리하기 어렵다.

- 배열을 선언하면 선언한 자료형과 배열 길이에 따라 메모리가 할당된다.

int num[20];



num[0] num[1] num[2]

num[19]

num - 배열 이름
[] - 인덱스 연산자

0 ~ n-1 개



배열 사용하기

● 배열 선언 및 자료 저장

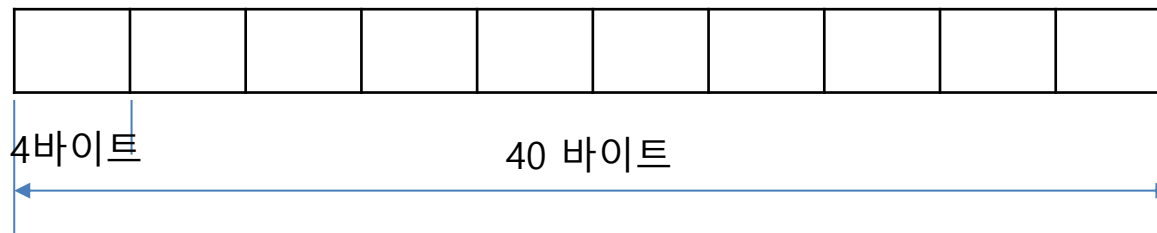
① 자료형[] 배열 이름 = **new** 자료형[개수]

② 자료형 배열 이름[] = **new** 자료형[개수]

```
int[ ] numbers = new int[10]
```

- 배열 길이(length) - 10개

numbers



- 배열의 길이 - `numbers.length`



배열(Array)

■ 배열의 선언과 사용

```
/* 변수를 사용하여 점수 저장 - 비효율적
int std1KorScore = 0;
int std1EngScore = 0;
int std1MathScore = 0;

int std2KorScore = 0;
int std2EngScore = 0;
int std2MathScore = 0;*/

//배열을 이용하여 점수 저장
int[] korScore = new int[3];    //3명의 국어점수 배열
korScore[0] = 90;
korScore[1] = 80;
korScore[2] = 70;

//int korScore = {90, 80, 70}; //배열 생성시 초기화

//배열의 크기(개수)
System.out.println(korScore.length + "개");

//1번 인덱스 값 조회
System.out.println(korScore[1]);
```



배열(Array)

■ 정수형 배열 생성 및 배열의 크기

```
// 배열 초기화 및 출력
int[] num = new int[] {1, 2, 3, 4};
//int[] num = {1, 2, 3, 4};

System.out.println("배열의 길이 : " + num.length);
System.out.printf("배열의 개수 : %d\n", num.length);

//배열 요소 접근 - 인덱싱
System.out.println(num[0]);
System.out.println(num[3]);

System.out.println();

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

```
배열의 길이 : 4
배열의 개수 : 4
1
4

1
2
3
4
```



배열(Array)

■ 정수형 배열의 연산

배열의 개수 : 4

3

6

9

12

4

3

```
int[] arr = new int[4];
int sum = 0; //총합

//저장
arr[0] = 3;
arr[1] = 6;
arr[2] = 9;
arr[3] = 12;

System.out.printf("배열의 개수 : %d\n", arr.length);

//연산
System.out.println(arr[0] + 1);
System.out.println(arr[2] - arr[1]);
System.out.println("=====");

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

//총합
for(int i=0; i<arr.length; i++) {
    sum += arr[i];
}
System.out.println("총합 : " + sum);
```



배열(Array)

■ 디버깅(Debugging)

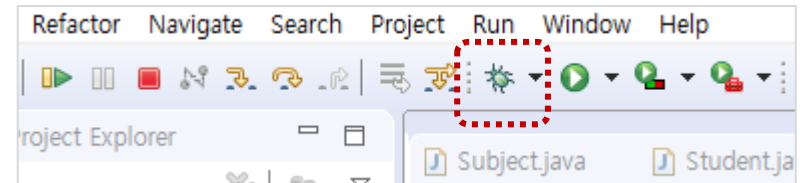
① 브레이크 포인트 설정

```
6 // 배열의 연산
7 int[] arr = new int[4];
8
9 //저장
10 arr[0] = 3;
11 arr[1] = 6;
12 arr[2] = 9;
13 arr[3] = 12;
14
15 System.out.printf("배열의 개수 : %d\n", arr.length);
16
```

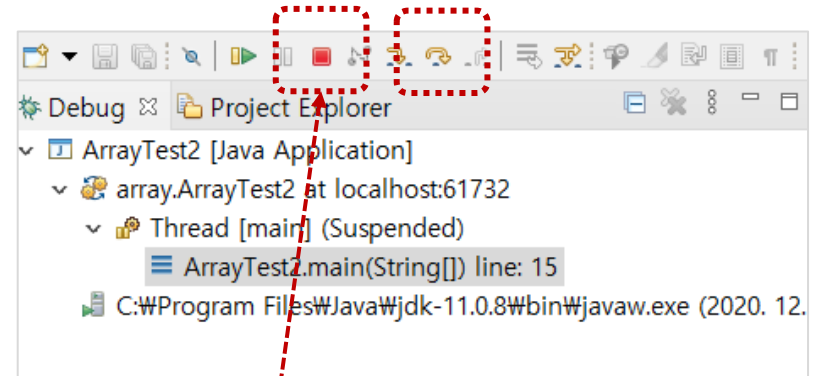
④ 결과 확인

(x)= Variables Breakpoints Expressions	
Name	Value
no method return value	
args	String[0] (id=19)
arr	(id=20)
[0]	3
[1]	6
[2]	9
[3]	12

② 디버그 실행



③ Step Over(단계)



⑤ 디버깅 종료



배열(Array)

■ 실수형 배열의 연산

```
double[] data = new double[5];
double total = 0.0;    //총합
double times = 0.0;    //곱한값

//저장
data[0] = 10.0;
data[1] = 20.0;
data[2] = 30.0;

//연산 및 조회
for(int i=0; i<data.length; i++) {
    total += data[i];
    times *= data[i];
    System.out.println(data[i]);
}
System.out.println();
System.out.println("총합 : " + total);
System.out.println("곱 : " + times);
```

```
10.0
20.0
30.0
0.0
0.0

총합 : 60.0
곱 : 0.0
```



배열(Array)

■ 배열의 유효한 요소값 출력하기

```
double[] data = new double[5];
int size = 0;

data[0] = 10.0;
size++;

data[1] = 20.0;
size++;

data[2] = 30.0;
size++;

//출력
for(int i=0; i<size; i++) {
    System.out.println(data[i]);
}
```

```
10.0
20.0
30.0
```



배열(Array)

■ 문자열형 배열 생성 및 조회(출력)

```
String[] cars = {"Morning", "Sonata", "Sportage", "K7"};
//배열의 값 찾기
System.out.println(cars[1]);

//배열의 값 변경하기
cars[2]="BMW";
System.out.println(cars[2]);

//배열의 값 출력
for(int i=0; i<cars.length; i++) {
    System.out.println(cars[i]);
}
System.out.println("-----");

//항상 for로 출력
for(String car : cars) {
    System.out.println(car);
}
```



배열(Array)

■ 문자형 배열 – 알파벳 저장하고 출력하기

```
char[] alphabets = new char[26];
char ch = 'A';
int i;

/*
//기본적으로 공백이 저장되어 초기화 됨.
alphabets[0] = ch;
System.out.println("0번째 요소 : " + alphabets[0]);
ch++;
System.out.println(ch);
*/

//배열에 알파벳 저장
for(i=0; i<alphabets.length; i++) {
    alphabets[i] = ch;
    ch++;
}

//출력
for(i=0; i<alphabets.length; i++) {
    System.out.println((int)alphabets[i] + ", " + alphabets[i]);
}
```

65,	A
66,	B
67,	C
68,	D
69,	E
70,	F
71,	G
72,	H
73,	I
74,	J
75,	K
76,	L
77,	M
78,	N
79,	O
80,	P
81,	Q
82,	R
83,	S
84,	T
85,	U
86,	V



배열 실습 예제

■ 실습 예제

배열 길이가 5인 정수 배열을 선언하고, 1~10중 짝수만을 배열에 저장한 후 그 합을 출력하세요.

👉 실행결과

총합 : 30

```
int[] arr = new int[5];
int sum = 0;
int i;

//저장
for(i=0; i<arr.length; i++) {
    arr[i] = (i+1)*2;
}

//조회
for(i=0; i<arr.length; i++) {
    System.out.println("arr["+i+"]"+"="+arr[i]);
}

//합계
for(i=0; i<arr.length; i++) {
    sum += arr[i];
}
System.out.println("총합 : " + sum);
```



향상된 for문과 배열

■ 향상된 for문

```
for(자료형 변수 : 배열이름){  
    반복실행(변수)  
}
```

```
int[ ] numbers = {1, 2, 3, 4, 5};
```

```
for(int num : numbers) {
```

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

```
}
```

```
String[ ] cars = {"Morning", "Sonata", "Sportage", "K7"};
```

```
for(String car : cars) {
```

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

```
}
```



배열 복사하기

■ 배열 복사하기

역순으로 복사

```
TEN
=====
cow dog cat
```

```
// 배열의 역순 복사
char[] arr1 = {'N', 'E', 'T'};
char[] arr2 = new char[3];
int i;

for(i=0; i<arr1.length; i++) {
    arr2[2-i] = arr1[i];
}

for(i=0; i<arr2.length; i++) {
    System.out.print(arr2[i] + "");
}

System.out.println("\n=====");

String[] str1 = {"cat", "dog", "cow"};
String[] str2 = new String[3];

for(i=0; i<str1.length; i++) {
    str2[2-i] = str1[i];
}

for(i=0; i<str2.length; i++) {
    System.out.print(str2[i] + " ");
}
```



배열 복사하기

■ 배열 복사하기

1. 기존 배열과 자료형 및 배열 크기가 똑같은 배열을 새로 만들때.
2. 배열의 모든 요소에 자료가 꼭 차서 더 큰 배열을 만들때

```
int[] array1 = {10, 20, 30, 40, 50};  
int[] array2 = new int[5];  
int i;
```

```
for(i=0; i<array2.length; i++) {  
    array2[i] = array1[i];  
}
```

① 배열 복사

```
for(i=0; i<array2.length; i++) {  
    System.out.println(array2[i]);  
}
```

배열의 끝은 세미콜론 붙임



배열 복사하기

■ 배열 복사하기

```
//2. clone 메서드
int[] array3 = array2.clone();
for(int i : array3) {
    System.out.print(i + " ");
}

System.out.println("\n-----");
//3. System 클래스의 arraycopy() 메서드
System.arraycopy(array1, 0, array2, 0, 5);
for(int i=0; i<array2.length; i++) {
    System.out.print(array2[i] + " ");
}
```

10	20	90	40	50

10	20	90	40	50

10	20	90	40	50

배열의 인덱스

배열의 크기(개수)



최대값 찾기

- 최대값과 최대값
위치 찾기

```
max = 8  
max_idx = 3
```

```
//최대값 구하기  
int[] array = {1, 5, 3, 8, 2};  
int max = array[0]; //0번 인덱스값을 최대값으로 설정  
  
for(int i=1; i<array.length; i++) {  
    if(max < array[i]) {  
        max = array[i];  
    }  
}  
  
System.out.println("max = " + max);  
  
//최대값 위치 구하기  
int max_idx = 0; //인덱스 0을 최대값으로 설정  
for(int i=1; i<array.length; i++) {  
    if(array[max_idx] < array[i]) {  
        max_idx = i;  
    }  
}  
  
System.out.println("max_idx = " + max_idx);
```



배열을 이용한 성적 처리 프로그램

■ 성적 처리 프로그램

키보드로부터 학생 수와 각 학생들의
점수를 입력받아 최고 점수 및
평균 점수를 구하는 프로그램 만들기

```
=====
1. 학생수 | 2. 점수입력 | 3. 점수리스트 | 4. 분석 | 5. 종료
=====

선택>1
학생수>3
=====
1. 학생수 | 2. 점수입력 | 3. 점수리스트 | 4. 분석 | 5. 종료
=====

선택>2
scores[0]: 80
scores[1]: 90
scores[2]: 70
=====
1. 학생수 | 2. 점수입력 | 3. 점수리스트 | 4. 분석 | 5. 종료
=====

선택>3
scores[0]: 80
scores[1]: 90
scores[2]: 70
=====
1. 학생수 | 2. 점수입력 | 3. 점수리스트 | 4. 분석 | 5. 종료
=====

선택>4
평균 점수: 80.0
최고 점수: 90
=====
1. 학생수 | 2. 점수입력 | 3. 점수리스트 | 4. 분석 | 5. 종료
=====

선택>5
프로그램 종료
```



배열을 이용한 성적 처리 프로그램

```
public class ScoreMain {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int[] scores = null; //학생수 대로 개수 생성(즉, 점수의 개수)
        boolean run = true;

        while(run) {
            System.out.println("=====");
            System.out.println(" 1.학생수 | 2.점수입력 | 3.점수리스트 | 4.분석 | 5.종료");
            System.out.println("=====");
            try {
                System.out.print("선택>");
                int selNum = sc.nextInt();
                if(selNum == 1) {
                    //학생수
                    System.out.print("학생수>");
                    int studentsCount = sc.nextInt();
                    scores = new int[studentsCount]; //배열의 크기 생성
                }
                else if(selNum == 2) {
                    //점수 입력
                    for(int i=0; i<scores.length; i++) {
                        System.out.print("scores[" + i + "]: ");
                        scores[i] = sc.nextInt();
                    }
                }
                else if(selNum == 3) {
                    //점수 리스트
                    for(int i=0; i<scores.length; i++) {
                        System.out.println("scores[" + i + "]: " + scores[i]);
                    }
                }
            }
        }
    }
}
```



배열을 이용한 성적 처리 프로그램

```
else if(selNum == 4) {
    //계산
    int sum = 0;
    double avg = 0.0;
    //평균 점수
    for(int i=0; i<scores.length; i++) {
        sum += scores[i];
    }
    avg = (double)sum / scores.length;
    System.out.println("평균 점수: " + avg);

    //최고 점수
    int max = 0;
    for(int i=0; i<scores.length; i++) {
        if(max < scores[i])
            max = scores[i];
    }
    System.out.println("최고 점수: " + max);
}else if(selNum == 5) {
    //종료
    run = false;
}
else {
    System.out.println("메뉴를 잘못 눌렀습니다.");
}
}catch(NullPointerException e) {
    System.out.println("1. 먼저 학생수를 선택해 주세요!");
}
}
System.out.println("프로그램 종료");
sc.close();
}
```



순위(rank)

■ 순위 정하기

1. rank 배열에 초기값을 1로 저장한다.

2. 중첩 for문을 사용하여 행 단위로 값을 비교하여 뒤수가 앞수보다 크면 count값을 1증가시킨다.

```
int[] score = {70, 90, 60, 50, 80};
int[] rank = new int[5];
int i, j;

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

/*
초기값 count=1 rank={1, 1, 1, 1, 1}
i=0 일때    70<70, 70<90, 70<60, 70<50, 70<80 -> count=3
i=1 일때    90<70, 90<90, 90<60, 90<50, 90<80 -> count=1
i=2 일때    60<70, 60<90, 60<60, 60<50, 60<80 -> count=4
*/

for(i=0; i<rank.length; i++) {
    System.out.print(rank[i] + " ");
}
```



정렬(Sort)

■ 배열 요소의 정렬

정렬(sort)은 자료를 크기 순서로 맞춰 일렬로 나열하는 것이다. 사전은 단어를 가나다순 혹은 알파벳순으로 나열한 정렬의 좋은 예이다. 버블 정렬, 선택정렬 등의 방법이 있다.

버블정렬

서로 인접한 두 원소를 검사하여 정렬하는 알고리즘

크기를 비교하여 서로 교환한다.

```
//오름차순 정렬 - 버블 정렬 방식
int[] arr = {3, 6, 9, 2, 5, 4};
int i, j, temp;

for(i=0; i<arr.length; i++) {
    for(j=0; j<arr.length-1; j++) {
        if(arr[j] > arr[j+1]) {
            temp = arr[j];
            arr[j] = arr[j+1];
            arr[j+1] = temp;
        }
    }
}

/*
1행.. 3, 6, 2, 5, 4, 9
2행.. 3, 2, 5, 4, 6, 9
3행.. 2, 3, 4, 5, 6, 9
*/

for(int a : arr)
    System.out.print(a + " ");
```



정렬(Sort)

✓ 알파벳 정렬하기 - 버블정렬

```
// 오름차순 정렬 - 버블정렬 방식
char[] arr = {'S', 'B', 'M', 'K', 'C', 'Z', 'A'};
int i, j;
char temp; //교환용 임시 변수
for(i=0; i<arr.length; i++) { // 행의 길이
    for(j=0; j<arr.length-1; j++) { // 열의 길이
        if(arr[j] > arr[j+1]) { //크기 비교후 교환
            temp = arr[j];
            arr[j] = arr[j+1];
            arr[j+1] = temp;
        }
    }
}

//출력
for(char ch : arr)
    System.out.print(ch + " ");

System.out.println();
//코드값 출력
for(int n : arr)
    System.out.print(n + " ");
```

A	B	C	K	M	S	Z
65	66	67	75	77	83	90



정렬(Sort)

✓ 선택 정렬 방식

```
package algorithm;
public class SelectionSort {
    public static void main(String[] args) {
        // 오름차순 정렬 - 선택 정렬 방식
        int[] arr = {3, 6, 9, 2, 5, 4};
        int i, j;
        int temp;    //교환을 위한 임시 변수
        int min_idx;

        for(i=0; i<arr.length-1; i++) {
            min_idx = i;
            for(j=i+1; j<arr.length; j++) {
                if(arr[j] < arr[min_idx])
                    min_idx = j;    //j=i+1
            }

            temp = arr[i];
            arr[i] = arr[min_idx]; //min_idx=j이므로 i+1과 같다.
            arr[min_idx] = temp;
        }
    }
}
```

선택정렬

위치를 이용하여 정렬하는 방법으로 첫위치(0번인덱스)를 최소로 정하고, 비교하여 자리바꿈을 한다.

그 다음행은 1번인덱스가 최소값 자리로 정하여 자리바꿈한다.



정렬(Sort)

✓ 선택 정렬 방식 - 검증 및 출력

```
/*
 * 1행 - 2, 6, 9, 3, 5, 4 -> 0번째 값이 최소값
 * 2행 - 2, 3, 9, 6, 5, 4 -> 1번째 값이 최소값
 * 3행 - 2, 3, 4, 9, 6, 5 -> 2번째 값이 최소값
 * 4행 - 2, 3, 4, 5, 9, 6 -> 3번째 값이 최소값
 * 5행 - 2, 3, 4, 5, 6, 9 -> 4번째 값이 최소값
 */

//출력
for(int a : arr) {
    System.out.print(a + " ");
}
```

2 3 4 5 6 9



다차원 배열

■ 2차원 이상의 배열

1. 지도, 게임 등 평면이나 공간을 구현할 때 많이 사용됨.
2. 이차원 배열의 선언과 구조

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

arr[0][0]	arr[0][1]	arr[0][2]
arr[1][0]	arr[1][1]	arr[1][2]

3. 선언과 초기화

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

arr[0][0]	arr[0][1]	arr[0][2]
1	2	3
4	5	6
arr[1][0]	arr[1][1]	arr[1][2]



다차원 배열

■ 2차원 배열 생성, 초기화, 저장하기 1

```
//2행 3열의 정수형 이차원 배열 생성
```

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

```
int i, j;
```

```
System.out.printf("배열의 길이(행) : %d개\n", a.length);
```

```
System.out.printf("배열의 길이(열) : %d개\n", a[0].length);
```

```
System.out.println("배열의 초기값 출력");
```

```
for(i=0; i<a.length; i++) {
```

```
    for(j=0; j<a[i].length; j++) {
```

```
        System.out.print(a[i][j] + " ");
```

```
    }
```

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

```
}
```

배열의 길이(행) : 2개

배열의 길이(열) : 3개

배열의 초기값 출력

0 0 0

0 0 0



다차원 배열

■ 2차원 배열 생성, 초기화, 저장 및 출력

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

System.out.println("배열의 유효한 값 출력");
for(i=0; i<a.length; i++) {
    for(j=0; j<a[i].length; j++) {
        System.out.print(a[i][j] + " ");
    }
    System.out.println();
}
```

배열의 유효한 값 출력

1	2	3
4	5	6



다차원 배열

- 실습 예제 - 아파트 세대 구현하기

1, 1	1, 2	1, 3
2, 1	2, 2	2, 3
3, 1		
4, 1	4, 2	4, 3
5, 1	5, 2	5, 3

```
int[][] household = new int[5][3];  
household[2] = new int[1];
```

```
System.out.printf("1층 %d세대\n", household[0].length);  
System.out.printf("2층 %d세대\n", household[1].length);  
System.out.printf("3층 %d세대\n", household[2].length);  
System.out.printf("4층 %d세대\n", household[3].length);  
System.out.printf("5층 %d세대\n", household[4].length);
```

1층	3	세대
2층	3	세대
3층	1	세대
4층	3	세대
5층	3	세대



다차원 배열

■ 2차원 배열의 연산

```
numbers[1][2] = 7  
1 2 3 4 5 6 7  
합계: 28  
개수: 7  
평균: 4.0
```

```
int[][] numbers = { {1, 2, 3, 4}, {5, 6, 7} };  
int i, j;  
int sum = 0;  
int count = 0;  
double avg = 0.0;  
  
//인덱싱  
int x = numbers[1][2];  
System.out.println("numbers[1][2] = " + x);  
  
//전체 조회  
for(i=0; i<numbers.length; i++) {  
    for(j=0; j<numbers[i].length; j++) {  
        System.out.print(numbers[i][j] + " ");  
    }  
}  
System.out.println();  
  
//합계 및 평균  
for(i=0; i<numbers.length; i++) {  
    for(j=0; j<numbers[i].length; j++) {  
        sum += numbers[i][j];  
        count++;  
    }  
}  
avg = (double)sum / count;  
System.out.println("합계: " + sum);  
System.out.println("개수: " + count);  
System.out.println("평균: " + avg);
```



다차원 배열

- 실습 예제

학생 5명의 국어와 수학 점수를 이용하여 합계와 평균을 구하세요.

Score2DArray.java

국어	수학	
91	70	
80	50	
75	62	
93	85	
80	70	
국어	합계:	419
수학	합계:	337
국어	평균:	83.8
수학	평균:	67.4



다차원 배열

```
int[][] score = {
    {91, 70},
    {80, 50},
    {75, 62},
    {93, 85},
    {80, 70}
};
int[] total = {0, 0};
double[] avg = {0.0, 0.0};
int i, j;

//점수 출력
System.out.println("국어 수학");
for(i=0; i<score.length; i++) {
    for(j=0; j<score[i].length; j++) {
        System.out.print(score[i][j] + " ");
    }
    System.out.println();
}
```

```
//합계 및 평균
for(i=0; i<score.length; i++) {
    total[0] += score[i][0]; //국어 합계
    total[1] += score[i][1]; //수학 합계
}
avg[0] = (double)total[0] / score.length;
avg[1] = (double)total[1] / score.length;

System.out.println("국어 합계: " + total[0]);
System.out.println("수학 합계: " + total[1]);
System.out.println("국어 평균: " + avg[0]);
System.out.println("수학 평균: " + avg[1]);
```



다차원 배열

■ 실습 예제

아래의 실행 결과대로 성적을 계산하는 프로그램을 작성하세요.

```
이름을 입력하세요: 김산
국어점수를 입력하세요: 100
수학점수를 입력하세요: 90
영어점수를 입력하세요: 85
이름을 입력하세요: 이강
국어점수를 입력하세요: 85
수학점수를 입력하세요: 76
영어점수를 입력하세요: 95
이름을 입력하세요: 정들
국어점수를 입력하세요: 60
수학점수를 입력하세요: 60
영어점수를 입력하세요: 60
김산 100 90 85 275 91.67
이강 85 76 95 256 85.33
정들 60 60 60 180 60.00
```

```
// 3명의 이름,국어점수, 수학점수, 영어점수를 입력받아
// 출력하는 프로그램을 만들어보세요. 합계, 평균(소수점)
// 김산 100 90 85 275
// 이강 85 76 95 256
// 정들 60 60 60 180
```

```
Scanner sc = new Scanner(System.in);
String[] title = {"국어", "수학", "영어"}; //제목
String[] name = new String[3]; //이름 저장
int[][] score = new int[3][4]; //점수 저장
double[] avg = new double[3]; //평균 저장
int sum = 0;
```



다차원 배열

```
for(int i=0; i<score.length; i++) {  
    System.out.print("이름을 입력하세요: ");  
    name[i] = sc.nextLine(); //김산 enter  
  
    for(int j=0; j<score.length; j++) {  
        System.out.print(title[j] + "점수를 입력하세요: ");  
        score[i][j] = sc.nextInt(); //90 enter  
        sum += score[i][j];  
    }  
    score[i][3] = sum; //각 행별 4번 열에 합계 저장  
    avg[i] = (double)sum / score.length;  
    sum = 0; //초기화  
  
    sc.nextLine();  
} //입력
```

```
for(int i=0; i<score.length; i++) {  
    System.out.print(name[i] + " ");  
    for(int j=0; j<score[i].length; j++) {  
        System.out.print(score[i][j]+" ");  
    }  
    System.out.printf("%.2f", avg[i]);  
    System.out.println();  
} //출력  
  
sc.close();
```

