Java의 정석

제 5 장

배 열

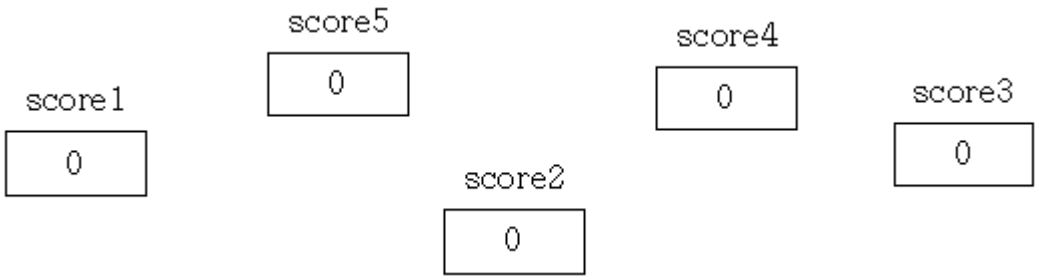
1. 배열(array)

1. 배열(array)

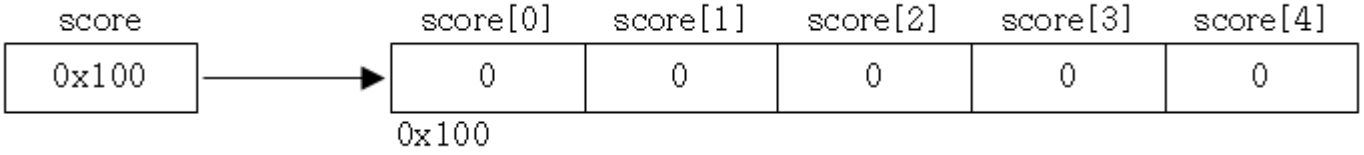
1.1 배열(array)이란?

- 같은 타입의 여러 변수를 하나의 묶음으로 다루는 것
- 많은 양의 값(데이터)을 다룰 때 유용하다.
- 배열의 각 요소는 서로 연속적이다.

```
int score1=0, score2=0, score3=0, score4=0, score5=0 ;
```




```
int[] score = new int[5]; // 5개의 int 값을 저장할 수 있는 배열을 생성한다.
```



1.2 배열의 선언과 생성(1)

- 타입 또는 변수이름 뒤에 대괄호[]를 붙여서 배열을 선언한다.

선언방법	선언 예
타입[] 변수이름;	`int[] score;` `String[] name;`
타입 변수이름[];	`int score[];` `String name[];`

[표5-1] 배열의 선언방법과 선언 예

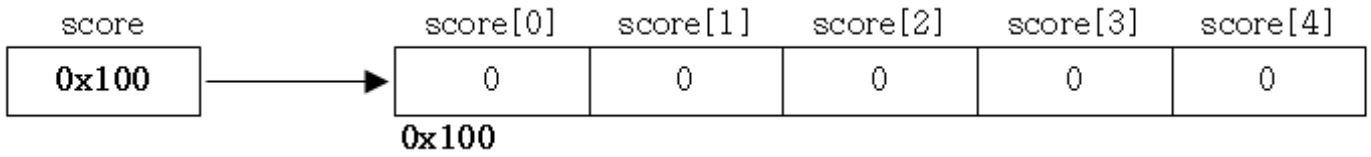
1.2 배열의 선언과 생성(2)

- 배열을 선언한다고 해서 값을 저장할 공간이 생성되는 것이 아니라 배열을 다루는데 필요한 변수가 생성된다.

```
int[] score;          // 배열을 선언한다.(생성된 배열을 다루는데
score = new int[5]; // 배열을 생성한다.(5개의 int값을 저장할
```

[참고] 위의 두 문장은int[] score = new int[5];와 같이 한 문장으로 줄여 쓸 수 있

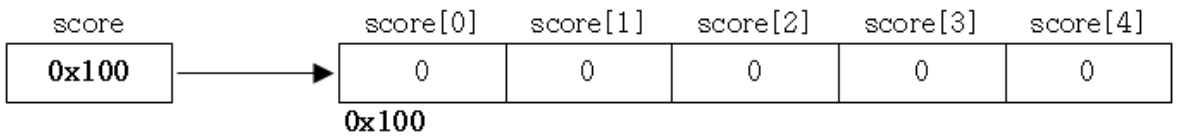
자료형	기본값
boolean	false
char	'\u0000'
byte	0
short	0
int	0
long	0L
float	0.0f
double	0.0d 또는 0.0
참조형 변수	null



1.3 배열의 초기화

- 생성된 배열에 처음으로 값을 저장하는 것

```
int[] score = new int[5]; // 크기가 5인 int형 배열을 생성한다.
score[0] = 100;           // 각 요소에 직접 값을 저장한다.
score[1] = 90;
score[2] = 80;
score[3] = 70;
score[4] = 60;
```




```
int[] score = { 100, 90, 80, 70, 60};          // 1번
int[] score = new int[]{ 100, 90, 80, 70, 60}; // 2번
```

```
int[] score;
score = { 100, 90, 80, 70, 60}; // 에러 발생!!!

int[] score;
score = new int[]{ 100, 90, 80, 70, 60}; // OK
```

```
int add(int[] arr) { /* 내용 생략 */}

int result = add({ 100, 90, 80, 70, 60});          // 에러 발생!!!
int result = add(new int[]{ 100, 90, 80, 70, 60}); // OK
```

1.4 배열의 활용

▶ 배열에 값을 저장하고 읽어오기

```
score[3] = 100;      // 배열 score의 4번째 요소에 100을 저장한다.
int value = score[3]; // 배열 score의 4번째 요소에 저장된 값을 읽어서 value에 저장.
```

▶'배열이름.length'는 배열의 크기를 알려준다.

```
int[] score = { 100, 90, 80, 70, 60, 50 };

for(int i=0; i < 6; i++) {
    System.out.println(socre[i]);
}
```



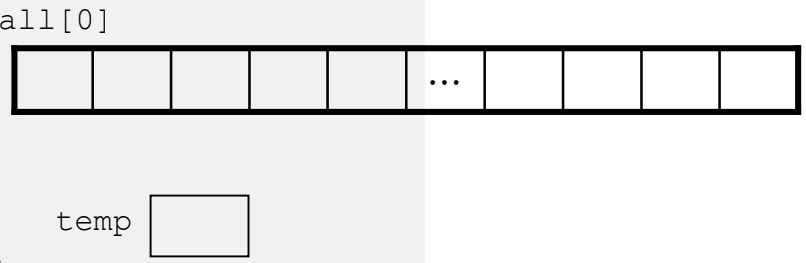
```
for(int i=0; i < score.length; i++) {
    System.out.println(socre[i]);
}
```

1.4 배열의 활용 – 예제1

[예제5-4]/ch5/ArrayEx4.java

```
class ArrayEx4 {
   public static void main(String[] args)
   {
        // 45개의 정수값을 저장하기 위한 배열 생성.
        int[] ball = new int[45];

        // 배열의 각 요소에 1~45의 값을 저장한다.
        for(int i=0; i < ball.length; i++)
            ball[i] = i+1;    // ball[0]에 1이 저장된다.

        int temp = 0;  // 두 값을 바꾸는데 사용할 임시변수
        int j = 0;     // 임의의 값을 얻어서 저장할 변수

        // 배열에 저장된 값이 잘 섞이도록 충분히 큰 반복횟수를 지정한다.
        // 배열의 첫 번째 요소와 임의의 요소에 저장된 값을 서로 바꿔서 값을 섞는다.
        for(int i=0; i < 100; i++) {
            j = (int)(Math.random() * 45); // 배열 범위(0~44)의 임의의 값을 얻는다.
            temp = ball[0];
            ball[0] = ball[j];
            ball[j] = temp;
        }
        // 배열 ball의 앞에서 부터 6개의 요소를 출력한다.
        for(int i=0; i < 6; i++)
            System.out.print(ball[i]+" ");
   }
}
```

ball[0]과 ball[j]의 값을 서로 바꾼다.



1.4 배열의 활용 – 예제2

[예제5-7]/ch5/ArrayEx7.java

```
class ArrayEx7
{
   public static void main(String[] args)
   {
        char[] hex = { 'C', 'A', 'F', 'E'};

        String[] binary = {   "0000", "0001", "0010", "0011"
                            , "0100", "0101", "0110", "0111"
                            , "1000", "1001", "1010", "1011"
                            , "1100", "1101", "1110", "1111" };

        String result="";

        for (int i=0; i < hex.length ; i++ ) {
            if(hex[i] >='0' && hex[i] <='9') {
                result +=binary[hex[i]-'0'];    // '8'-'0'의 결과는 8이다.
            } else { // A~F이면
                result +=binary[hex[i]-'A'+10]; // 'C'-'A'의 결과는 2
            }
        }

        System.out.println("hex:"+ new String(hex));
        System.out.println("binary:"+result);
   }
}
```

문자	코드
…	…
0	48
1	49
2	50
…	…
A	65
B	66
C	67
…	…

1.5 다차원 배열의 선언과 생성

-'[]'의 개수가 차원의 수를 의미한다.

선언방법	선언예
타입[][] 변수이름;	int[][] score;
타입 변수이름[][];	int score[][];
타입[] 변수이름[];	int[] score[];

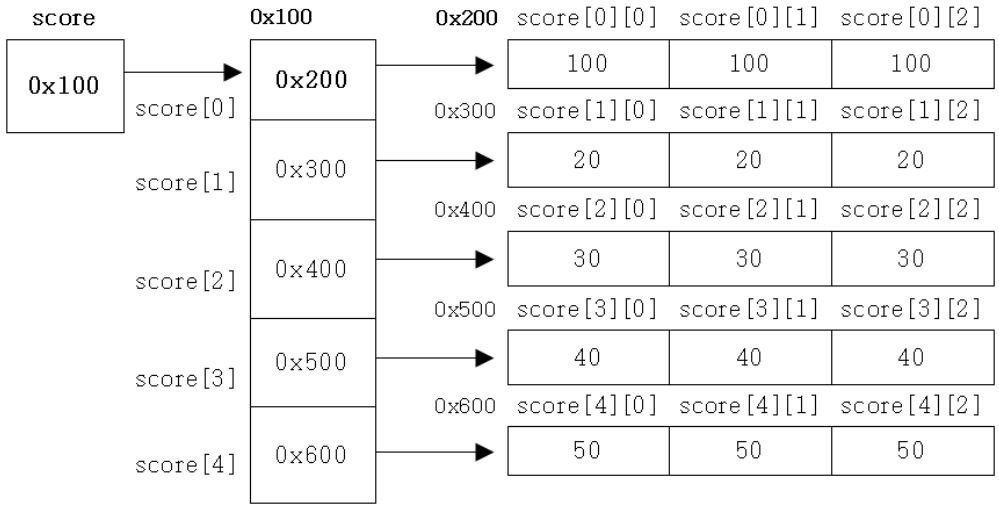
[표5-3] 2차원 배열의 선언

```
int[][] score = {
                    {100,100,100},
                    { 20, 20, 20},
                    { 30, 30, 30},
                    { 40, 40, 40},
                    { 50, 50, 50},
                };
```

```
int[][] score = new int[5][3];      // 5행 3열의 2차원 배열을 생성한다.
```

	국어	영어	수학
1	100	100	100
2	20	20	20
3	30	30	30
4	40	40	40
5	50	50	50

```
for (int i=0; i < score.length; i++) {
    for (int j=0; j < score[i].length; j++) {
        score[i][j] = 10;
    }
}
```



[그림5-2] 2차원 배열

1.6 가변배열

- 다차원 배열에서 마지막 차수의 크기를 지정하지 않고 각각 다르게 지정.

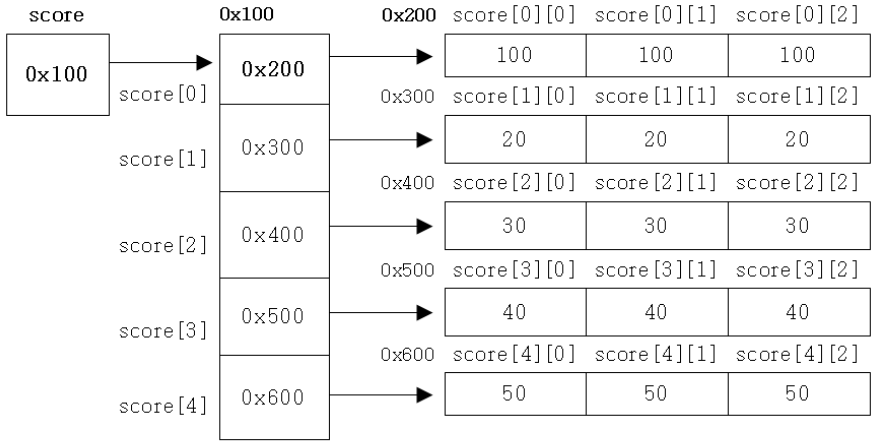
```
int[][] score = new int[5][3];      // 5행 3열의 2차원 배열을 생성한다.
```

```
int[][] score = new int[5][];
score[0] = new int[3];
score[1] = new int[3];
score[2] = new int[3];
score[3] = new int[3];
score[4] = new int[3];
```

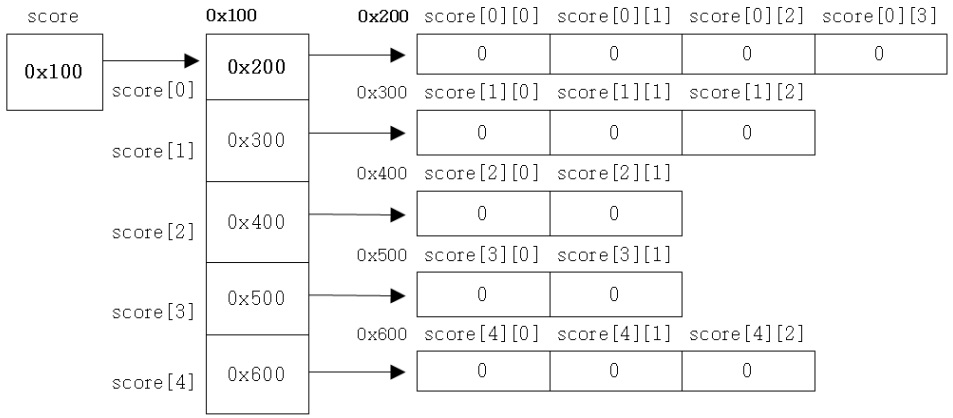
```
int[][] score =
    {
        {100,100,100},
        { 20, 20, 20},
        { 30, 30, 30},
        { 40, 40, 40},
        { 50, 50, 50},
    };
```

```
int[][] score = new int[5][];
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];
```

```
int[][] score =
    {
        {100,100,100,100},
        { 20, 20, 20},
        { 30, 30},
        { 40, 40},
        { 50, 50, 50},
    };
```



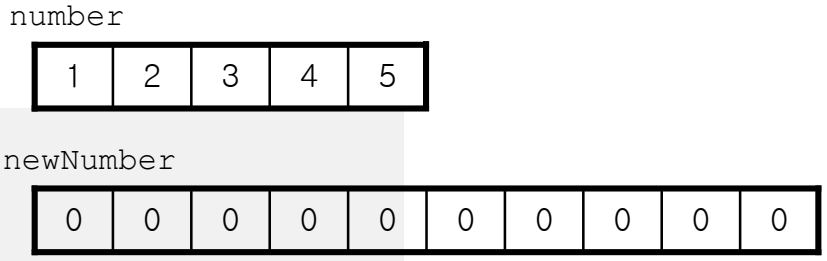
[그림5-2] 2차원 배열



[그림5-3] 가변배열

1.7 배열의 복사

▶ for문을 이용한 배열의 복사



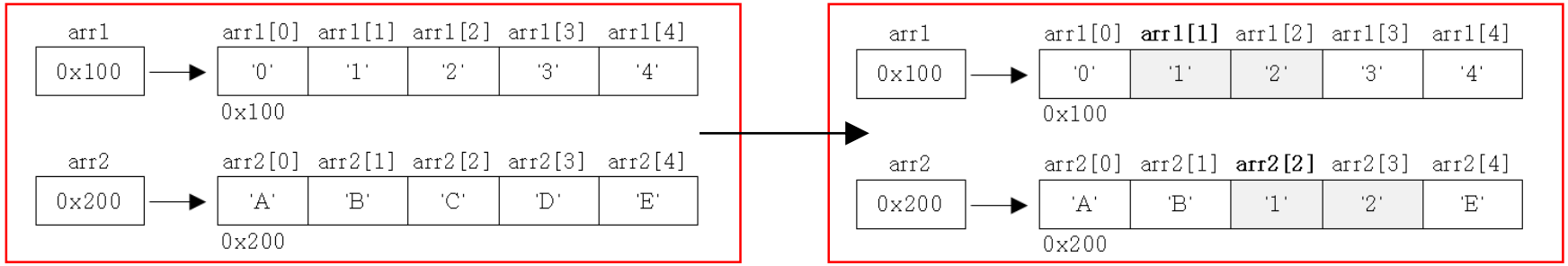

```
int[] number = {1,2,3,4,5};
int[] newNumber = new int[10];

for(int i=0; i<number.length;i++) {
    newNumber[i] = number[i];// 배열 number의 값을 newNumber에 저장한다.
}
```

▶ System.arraycopy()를 이용한 배열의 복사

```
System.arraycopy(arr1, 0, arr2, 0, arr1.length);
```

arr1[0]에서 arr2[0]으로 arr1.length개의 데이터를 복사

```
System.arraycopy(arr1, 1, arr2, 2, 2);
```



1.8 사용자 입력받기 - 커맨드라인

▶ 커맨드라인에서 입력된 값들은 문자열 배열에 담겨 main메서드에 전달된다.

[예제5-13]/ch5/ArrayEx13.java

```
class ArrayEx13
{
   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 + "] = \""+ args[i] + "\"");
        }

   }
}
```

[실행결과]

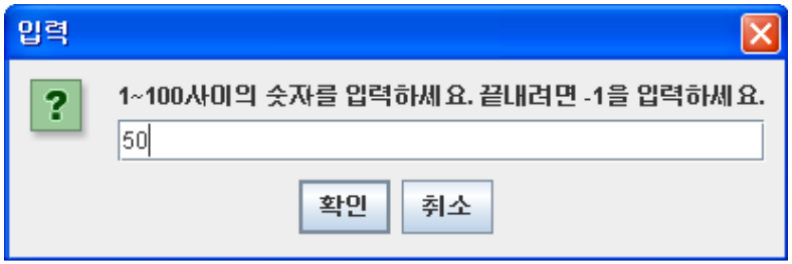
```
C:\jdk1.5\work>java ArrayEx13 abc 123 "Hello world"
매개변수의 개수:3
args[0] = "abc"
args[1] = "123"
args[2] = "Hello world"
```

1.9 사용자 입력받기 – 입력창(InputDialog)

- Swing패키지의 JOptionPane.showInputDialog()를 사용

[예제5-16]/ch5/ArrayEx16.java

```
import javax.swing.*;           // JOptionPane클래스를 사

class ArrayEx16 {
   public static void main(String[] args)
   {
        // 1~100사이의 임의의값을 얻어서 answer에 저장한다.
        int answer = (int)(Math.random() * 100) + 1;
        int input = 0;          // 사용자입력을 저장할 공간
        String temp = "";       // 사용자입력을 저장할 임시공간
        int count = 0;          // 시도횟수를 세기위한 변수

        do {
            count++;
            temp = JOptionPane.showInputDialog("1~100사이의 숫자를 입력하세요."
                                    + " 끝내려면 -1을 입력하세요.");

            // 사용자가 취소버튼을 누르거나 -1을 입력하면 do-while문을 벗어난다.
            if(temp==null || temp.equals("-1")) break;

            System.out.println("입력값 : "+temp);

            // 사용자입력을 문자열로 받아오기 때문에 int로 변환해 주어야한다.
            input = Integer.parseInt(temp);
```



[그림5-4] JOptionPane.showInputDialog()에 의해서 생성된 입력창