

5강 JAVA



배열

메모리

❖ 메모리 구조

- 모든 자바 프로그램은 자바 가상 머신(JVM)을 통해서 실행된다
- 운영체제에 독립적이다

❖ 메소드(method) 영역

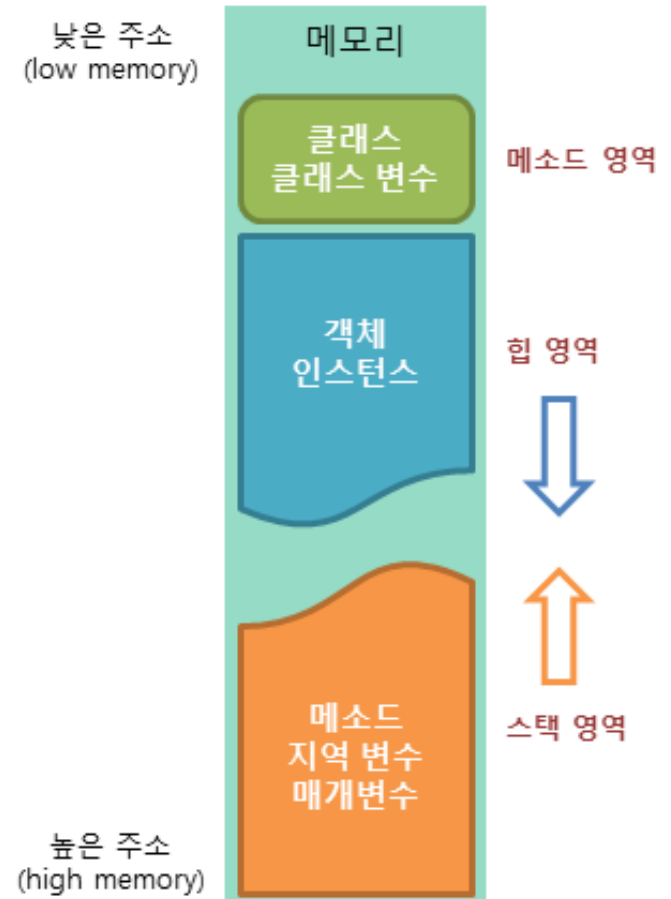
- 클래스의 정보 클래스 변수(static variable)저장
- JVM은 클래스가 사용될때 *.class 읽어들이어 해당 클래스에 대한 정보를 메소드 영역에 저장 한다.

❖ 힙(heap) 영역

- 인스턴스 변수가 저장되는 영역
- New키워드를 사용해서 만드는 변수 영역

❖ 스택(stack) 영역

- 메소드의 정보 또는 일반 변수들이 저장된다.



배열

❖ 배열이란?

- 같은 자료 형으로 연속된 메모리 공간에 할당하여 사용하는 것

❖ 배열의 쓰임

- 여러 개의 변수를 한 번에 선언 가능.
- 같은 형의 변수들을 연속적으로 쓸 수 있다.

❖ 배열 참조 변수 생성

DataType[] 배열이름;

- 일반 선언문과 동일하며 배열이름 뒤 배열의 요소 크기를 정한다.
- 배열 선언 시 [] 괄호가 1개일 때 1차원 배열 2개 일 때 2차원 배열이라 한다.
- 사용할 수 있는 배열의 요소의 첨자는 0부터 시작한다.
- 배열의 이름은 배열의 첫 번째 주소를 가지는 참조 변수이다.

예제

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

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

```
char[] str = new char[5];
```

```
char[] str = new char[]{'a','b','c'};
```

```
String[] str = new String[5];
```

```
String[] str = new String[]{"111","222","333"};
```

배열 예제

```
public class ArrayEx01 {  
    public static void main(String[] args) {  
        int[] arr = new int[5];  
        arr[0] = 100;  
        arr[1] = 200;  
        arr[2] = 300;  
        arr[3] = 400;  
        arr[4] = 500;  
        System.out.println("arr[0] : " + arr[0]);  
        System.out.println("arr[1] : " + arr[1]);  
        System.out.println("arr[2] : " + arr[2]);  
        System.out.println("arr[3] : " + arr[3]);  
        System.out.println("arr[4] : " + arr[4]);  
    }  
}
```

배열 예제

```
public class ArrayEx02 {  
    public static void main(String[] args) {  
        int[] arr = new int[5];  
        for(int i = 0; i < arr.length; i++){  
            arr[i] = 100*(i+1);  
        }  
  
        System.out.println("arr[0] : " + arr[0]);  
        System.out.println("arr[1] : " + arr[1]);  
        System.out.println("arr[2] : " + arr[2]);  
        System.out.println("arr[3] : " + arr[3]);  
        System.out.println("arr[4] : " + arr[4]);  
    }  
}
```

배열 예제

```
public class ArrayEx03 {  
    public static void main(String[] args) {  
        int[] arr = new int[5];  
        for(int i = 0; i < arr.length; i++) {  
            arr[i] = 100*(i+1);  
        }  
        for(int i = 0; i < arr.length; i++) {  
            System.out.println("arr["+i+"] : " + arr[i]);  
        }  
    }  
}
```


배열 예제

```
public class ArrayEx04 {  
    public static void main(String[] args) {  
        int[] arr = new int[] {10,20,30};  
  
        for(int i = 0; i < arr.length; i++) {  
            System.out.println("arr["+i+"] : " + arr[i]);  
        }  
    }  
}
```

배열 예제

```
public class ArrayEx05 {  
    public static void main(String[] args) {  
        Scanner input = new Scanner(System.in);  
        String name[] = new String[3];  
  
        for(int i = 0; i < name.length; i++) {  
            System.out.print(i+1+"번째 이름 입력 : ");  
            name[i] = input.next();  
        }  
        for(int i = 0; i < name.length; i++)  
            System.out.println(i+1+"번째 값 : "+ name[i]);  
    }  
}
```

선택정렬

❖ 정의

- 첫째 자리에 원하는 값을 위치하는 것으로 오름차순과 내림차순에 따라 값이 변할 수 있다. 오름 차순을 기준으로 하였을 경우 앞에 있는 값과 그 이후의 값들을 비교하여 가장 작은 값을 그 위치에 놓는다. 다음 위치에 있는 값과 그 이후의 값들을 비교하여 그 중 가장 작은 값을 놓는다. 정렬이 끝날 때까지 이를 반복하면 전체적으로 값이 정렬이 이루어 진다.

❖ 오름차순

- 수치가 점점 올라가는 수
- 예) 1, 2, 3, 4 ,5
- 예) 가, 나, 다, 라, 마

❖ 내림차순

- 수치가 점점 내려가는 수
- 예) 5, 4, 3, 2, 1
- 예) 마, 라, 다, 나, 가

선택정렬 알고리즘

❖ 오름차순

정렬전	4	8	2	7	6	비교
1차	4	8				$4 > 8$
2차	4		2			$4 > 2$
3차	2		4			swap
4차	2			7		$2 > 7$
5차	2				6	$2 > 6$
	2	8	4	7	6	

선택정렬 알고리즘

❖ 오름차순

정렬전	2	8	4	7	6	비교
1차		8	4			$8 > 4$
2차		4	8			swap
3차		4		7		$4 > 7$
4차		4			6	$4 > 6$
		4	8	7	6	

선택정렬 알고리즘

❖ 오름차순

정렬전	2	4	8	7	6	비교
1차			8	7		$8 > 7$
2차			7	8		swap
3차			7		6	$7 > 6$
4차			6		7	swap
			6	8	7	

선택정렬 알고리즘

❖ 오름차순

정렬전	2	4	6	8	7	비교
1차				8	7	$8 > 7$
2차				7	8	swap
완료	2	4	6	7	8	

선택정렬 예제

```
public class ArrayEx06 {  
    public static void main(String[] args) {  
        int arr[] = new int[] {4,8,2,7,6};  
        int i, j, tmp;  
        for(i = 0; i < arr.length-1; i++) {  
            for(j = i + 1; j < arr.length; j++){  
                if(arr[i] > arr[j]) //if(arr[i] < arr[j])  
                {  
                    tmp = arr[i];  
                    arr[i] = arr[j];  
                    arr[j] = tmp;  
                }  
            }  
        }  
        System.out.print("정렬 후 : " );  
        for(i = 0; i < arr.length; i++) {  
            System.out.print(arr[i] + " ");  
        }  
    }  
}
```


Random 예제

```
import java.util.Random;
public class ArrayEx07 {
    public static void main(String[] args) {
        Random ran = new Random(); // 객체 생성

        for(int i = 0; i < 5; i++)
            System.out.println(ran.nextInt(45)+1);
    }
}
```

다차원 배열

다차원 배열

❖ 정의

- 2차원 이상의 배열을 의미
- 배열을 입체적으로 표현한 모양

❖ 형식

- `type 배열명[][] = new type[][];`
- `type [][] 배열명 = new type[][];`

❖ 2차원 배열

(0,0)	(0,1)	(0,2)	(0,3)	(0,4)	(0,5)
(1,0)	(1,1)	(1,2)	(1,3)	(1,4)	(1,5)
(2,0)	(2,1)	(2,2)	(2,3)	(2,4)	(2,5)
(3,0)	(3,1)	(3,2)	(3,3)	(3,4)	(3,5)
(4,0)	(4,1)	(4,2)	(4,3)	(4,4)	(4,5)

다차원 배열 예제

```
public class ArrayEx08 {  
    public static void main(String[] args) {  
        int arr[][] = new int[][] {  
            {10,20,30},  
            {40,50,60},  
            {70,80,90}  
        };  
        System.out.print(arr[0][0] + " ");  
        System.out.print(arr[0][1] + " ");  
        System.out.print(arr[0][2] + " ");  
        System.out.println();  
        System.out.print(arr[1][0]+ " ");  
        System.out.print(arr[1][1]+ " ");  
        System.out.print(arr[1][2]+ " ");  
        System.out.println();  
        System.out.print(arr[2][0]+ " ");  
        System.out.print(arr[2][1]+ " ");  
        System.out.print(arr[2][2]+ " ");  
    }  
}
```

다차원 배열 예제

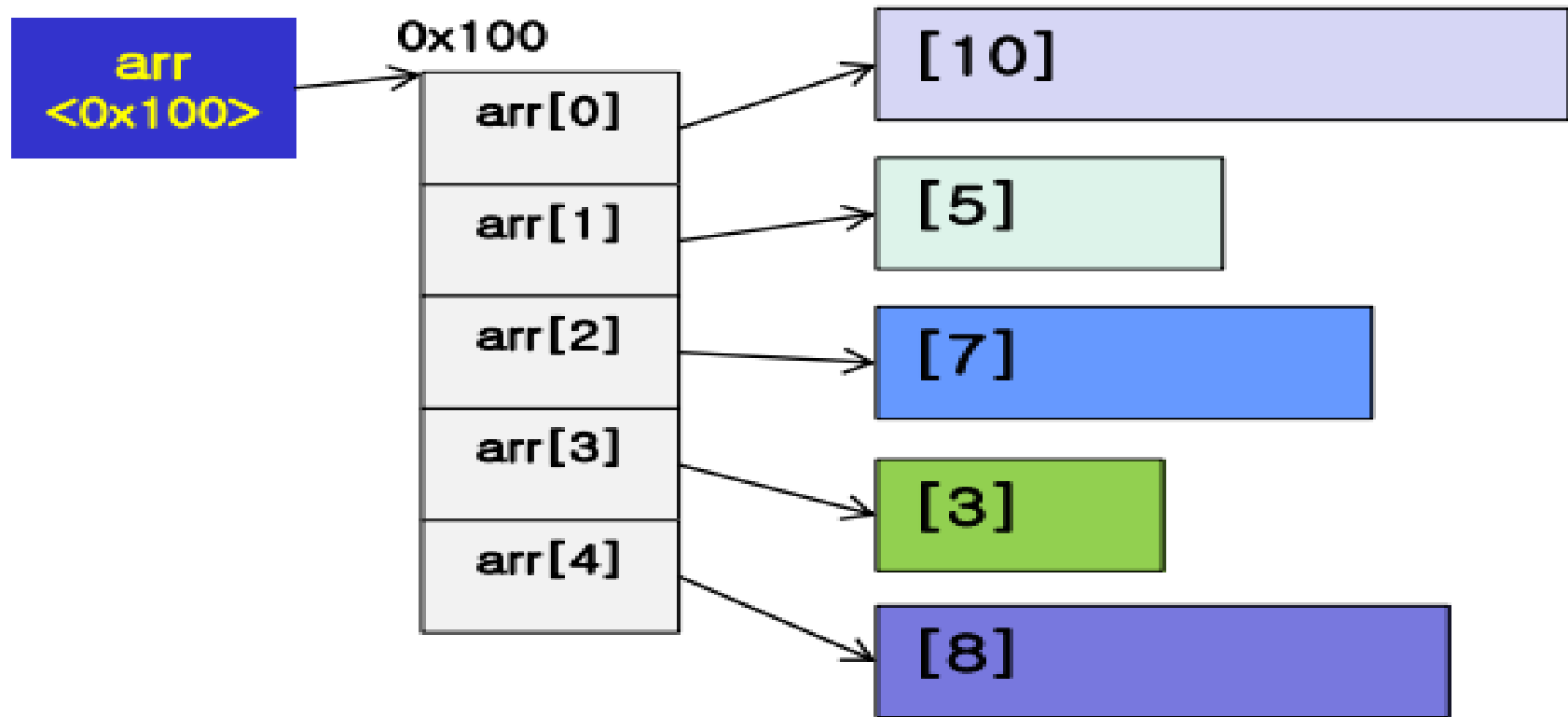
```
public class ArrayEx09 {  
    public static void main(String[] args) {  
        int arr[][] = new int[][] {  
            {10,20,30},  
            {40,50,60},  
            {70,80,90}  
        };  
        for(int i = 0; i < arr.length; i++) {  
            for(int j = 0; j < arr[i].length; j++) {  
                System.out.print(arr[i][j]+" ");  
            }  
            System.out.println();  
        }  
    }  
}
```

다차원 배열 예제

```
public static void main(String[] args) {  
    int arr[][] = new int[5][5];  
    int data = 1;  
    for(int i = 0; i < arr.length; i++) {  
        for(int j = 0; j < arr[i].length; j++) {  
            arr[i][j] = data++;  
        }  
    }  
    for(int i = 0; i < arr.length; i++) {  
        for(int j = 0; j < arr[i].length; j++) {  
            System.out.print(arr[i][j] + "\t");  
        }  
        System.out.println();  
    }  
}
```

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

가변길이 배열



<예제 배열 그림>

가변 배열 예제

```
public class ArrayEx11 {  
    public static void main(String[] args) {  
        Scanner input = new Scanner(System.in);  
        System.out.print("몇 행의 배열을 만드시겠습니까? : ");  
        int length = input.nextInt();  
        int[][] arr = new int[length][];  
        System.out.println("각 행에 배열을 만드세요!");  
        for (int i = 0; i < arr.length; i++) {  
            System.out.print(i + 1 + "번째 행의 배열 : ");  
            int n = input.nextInt();  
            arr[i] = new int[n];  
        }  
        for (int i = 0; i < arr.length; i++) {  
            for (int j = 0; j < arr[i].length; j++) {  
                arr[i][j] = j;  
            }  
        }  
        for (int i = 0; i < arr.length; i++) {  
            for (int j = 0; j < arr[i].length; j++) {  
                System.out.print(arr[i][j] + " ");  
            }  
            System.out.println();  
        }  
    }  
}
```