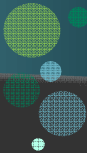
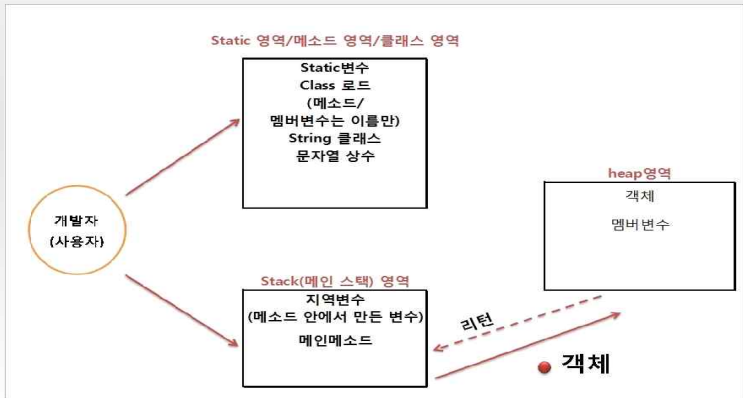


제6장 배열(Array)



1. 자바의 메모리 구조

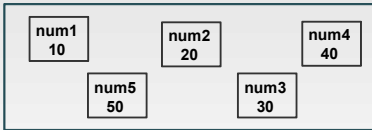


2. 배열(array)이란?

- 똑같은 데이터 타입의 여러 변수를 하나로 묶어 처리하는 자료구조
- 많은 데이터 양을 다룰 때, 유용하며 컬렉션 프레임워크의 기초가 된다.
- 배열의 각 요소들은 메모리에서 연속적으로 배치된다.

* 변수의 형태

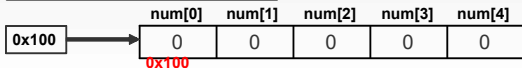
```
int num1 = 10;  
int num2 = 20;  
int num3 = 30;  
int num4 = 40;  
int num5 = 50;
```



좌측은 5개의 변수가 Stack에 메모리에 배치된 형태이다.

* 배열의 형태

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



좌측은 Stack에 참조변수 arr(배열명)이 heap(사용자 메모리 영역)에 new라는 연산자로 인해 5개의 4바이트 형태의 방이 연속적으로 메모리에 배치된 형태이다.

3. 배열의 선언과 생성

- 타입 또는 변수 이름 뒤에 대괄호[]를 붙여서 선언한다.

선언방법	선언 예
타입[] 변수이름;	<code>int[] score;</code> <code>String[] name;</code>
타입 변수이름[];	<code>int score[];</code> <code>String name[];</code>

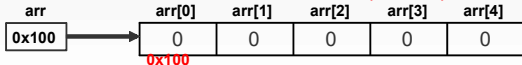
- 타입 정보 옆에 있는 것으로 쓰는 편이 좋다. 이유는 타입정보의 일 부이기 때문이다.

- 위와 같이 배열을 선언만 했다고, 저장할 수 있는 공간이 생성이 되는 것이 아니다.
- 배열을 다루는데 필요한 변수(무슨 변수?)가 생성된다.

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

- 첫 번째 코드를 실행하면 단지 `int[]` 타입의 참조변수 타입(4바이트)를 stack에 할당한다. 배열이 생성된 것이 아니다.

- 비로소, 두 번째 코드에서 `new` 연산자로 heap에 4바이트를 할당하는 방이 연속적으로 5개 생성된다.(20바이트)



좌측에서 중요한 것은 참조변수 `arr` 은 메모리의 주소를 저장하고 있다는 것에 주목을 해야 한다.

4. 배열의 초기화

- new 연산자로 heap영역에 배열을 생성했을 때, 처음으로 값을 저장하는 것
- 여기서, 중요한 것은 arr1, arr2는 배열의 주소 즉, 메모리의 주소 값을 지니고 있다.
- 즉, 배열명은 곧 주소인 것이다.(중요함)

```
//4*3=12바이트 생성, 0으로 초기화
```

```
int[] arr1 = new int[3];
```

```
//4*3=12바이트 생성 후, 바로 100, 20, 30으로 초기화함.
```

```
int[] arr2 = new int[] {100,20,30};
```

```
//참조변수만 선언해놓고, 방도 만들지 않았는데 값을  
//대입하고 있어서 초기화 문법 에러 발생
```

```
int[] arr3 = null;
```

```
arr3 = { 100, 50, 30};
```

5. 배열 사용하기

- 배열에 값을 저장하고, 또 읽어 들이기 방법

```
int[] arr1 = new int[3];  
//0번째 인덱스(참자) 배열방에 100 저장, 이 때 arr1[0]는  
//변수하고 동일함.  
arr1[0] = 100;  
  
//배열 0번째 방의 값을 읽어와서 value에 저장함.  
int value = arr1[0];
```

- ‘배열이름.length’ 는 배열의 크기를 알려주어, 반복문에서 효율적으로 사용 가능하다.

```
int[] arr = new int[] { 100, 200, 300 };  
  
//arr.length는 배열의 크기를 알려준다. 아래코드에서는  
//3이 된다. 하여 3번 루핑하면서 배열의 값을 출력해준다.  
for(int i=0; i<arr.length; i++) {  
    System.out.println("arr[" + i + "] 의 값 : " + arr[i]);  
}
```

감사합니다.

