

배열 (Array)

▶ 1차원 배열

1차원 배열이란?

타입이 같은 둘 이상의 데이터를 저장할 수 있는 1차원 구조의 메모리 공간

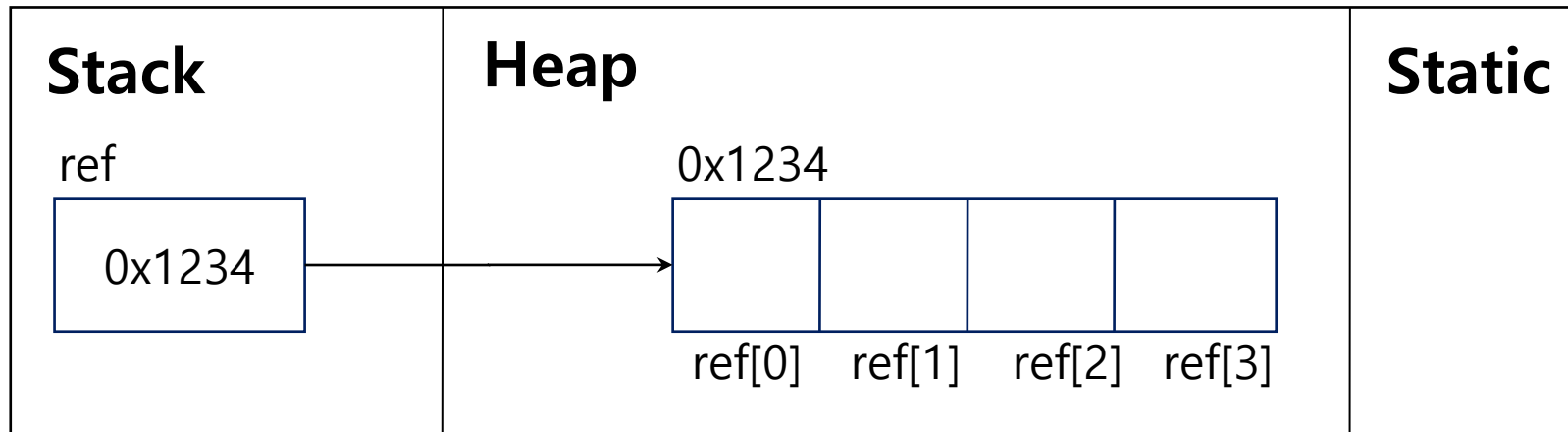
1차원 배열의 선언 방법

```
int[] ref = new int[5];    // 길이가 5인 int형 1차원 배열의 생성문
```

▶ 배열 저장구조

배열은 참조 변수로 Stack영역에 할당되며 배열 공간의 주소를 저장
배열 공간의 주소를 이용해 인덱스를 참조하는 방식으로 값 처리

```
int[] ref = new int[4];
```



▶ 배열선언문에 대한 세세한 결과

```
int[] ar = new int[3];           // 조금 더 선호하는 방법
```

```
int ar[] = new int[3];           참조변수의 이름뒤에 배열표시도 가능!
```

```
public static void main(String[] args) {  
    int[] ref;  
    ref = new int[5];  
    ....  
}
```

배열의 참조변수와 인스턴스의 선언도 분리 가능!

▶ 배열의 활용: 값의 저장과 참조

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

```
ar[0] = 7;           // 값의 저장: 첫 번째 요소
```

```
ar[1] = 8;           // 값의 저장: 두 번째 요소
```

```
ar[2] = 9;           // 값의 저장: 세 번째 요소
```

```
int num = ar[0] + ar[1] + ar[2];    // 값의 참조
```

▶ 배열 초기화

기본 자료형 배열은 모든 요소 **0**으로 초기화

```
int[] ar = new int[10];
```

String과 같은 객체배열은 모든 요소 **null**로 초기화

```
String[] ar = new String[10];
```

▶ 배열 초기화

배열 생성

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

배열 생성 및 초기화1

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

배열 생성 및 초기화2

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

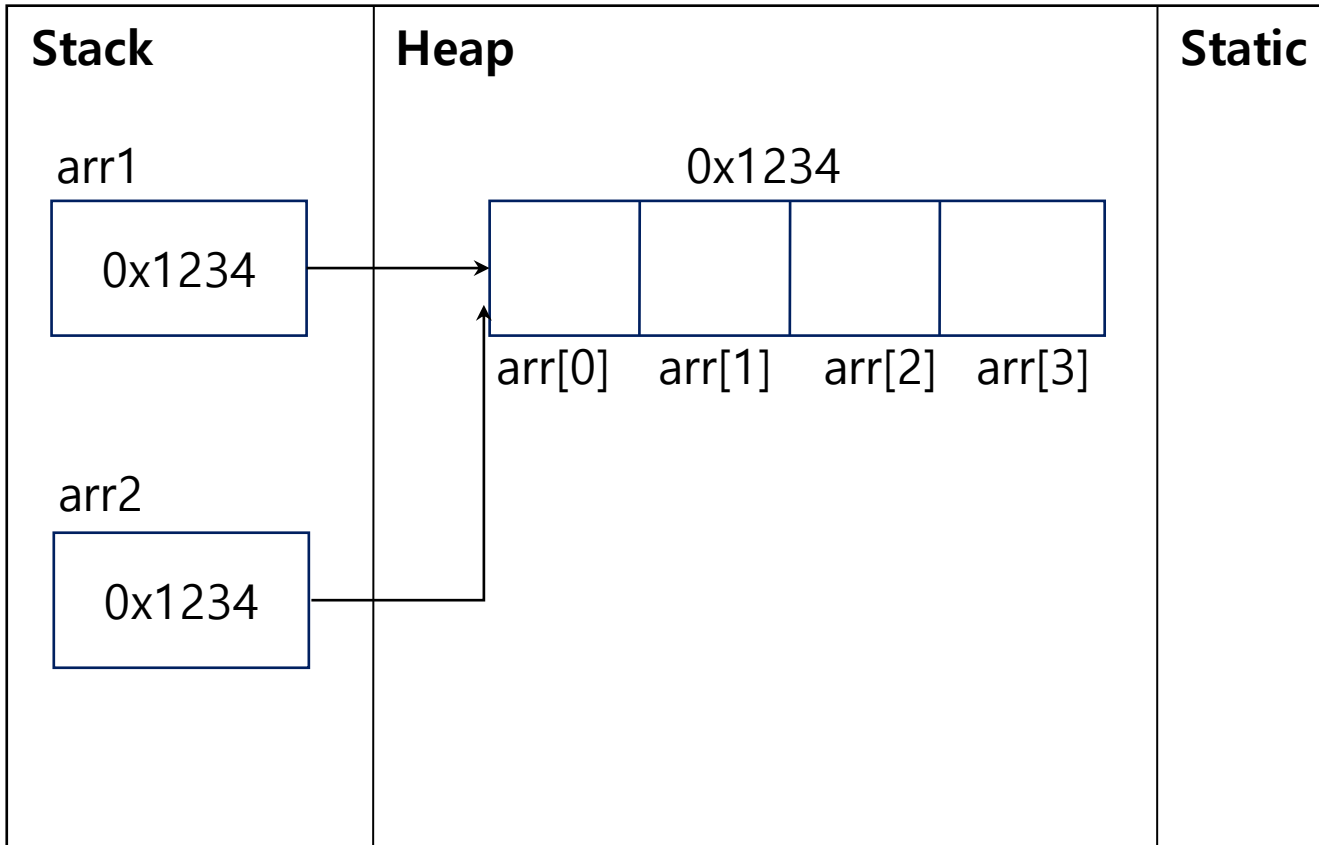
▶ 배열 복사

✓ 얇은 복사

객체의 주소 값만 가져와 참조형 변수에 저장하고 하나의 객체를 두 변수가 참조하는 것

```
int[] arr1 = new int[4];
```

```
int[] arr2 = arr1;
```



▶ 배열 복사

✓ 깊은 복사

새로운 배열 객체를 생성하여 기존 배열의 데이터를 복사하는 것

```
int[] arr1 = {1,2,3,4};  
int[] arr2 = new int[4];  
for(int i = 0; i < arr1.length; i++) {  
    arr2[i] = arr1[i];  
}
```

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

```
arr2 = Arrays.copyOf(arr1, arr1.length);
```

```
arr2 = arr1.clone();
```

