

Chap 05. Arrays

1. 배열 선언, 생성, 초기화
 - Primitive type Array
 - Reference type Array
2. 배열 Access
3. System.arraycopy()
4. 2차원 배열

- 정의 : 동일한 타입의 여러 데이터를 하나의 이름으로 관리하는 것

- 종류

 - Primitive type array

 - Reference type array

- 배열은 객체이다.

- 배열의 이름은 배열객체의 주소 값을 갖는 Reference 변수이다.

- 선언 : 어느 type을 배열의 element로 가질지 선언한다.

```
int[ ] numbers; ⇔ int numbers[];
```

```
MyDate[ ] dates; ⇔ MyDate dates[];
```

- 생성 : 배열도 객체이므로 new 키워드로 생성하며, [] 안에 몇 개의 element를 가질지 size를 지정한다.

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

```
dates = new MyDate[10];
```

- 초기화 : 사용하기 전에, 처음 값을 할당하는 것을 말한다.

```
numbers[0] = 1;
```

```
dates[0] = new MyDate();
```

□ 선언시 주의점.

`int[] a, b; // a, b 둘다 Array에 대한 Reference`

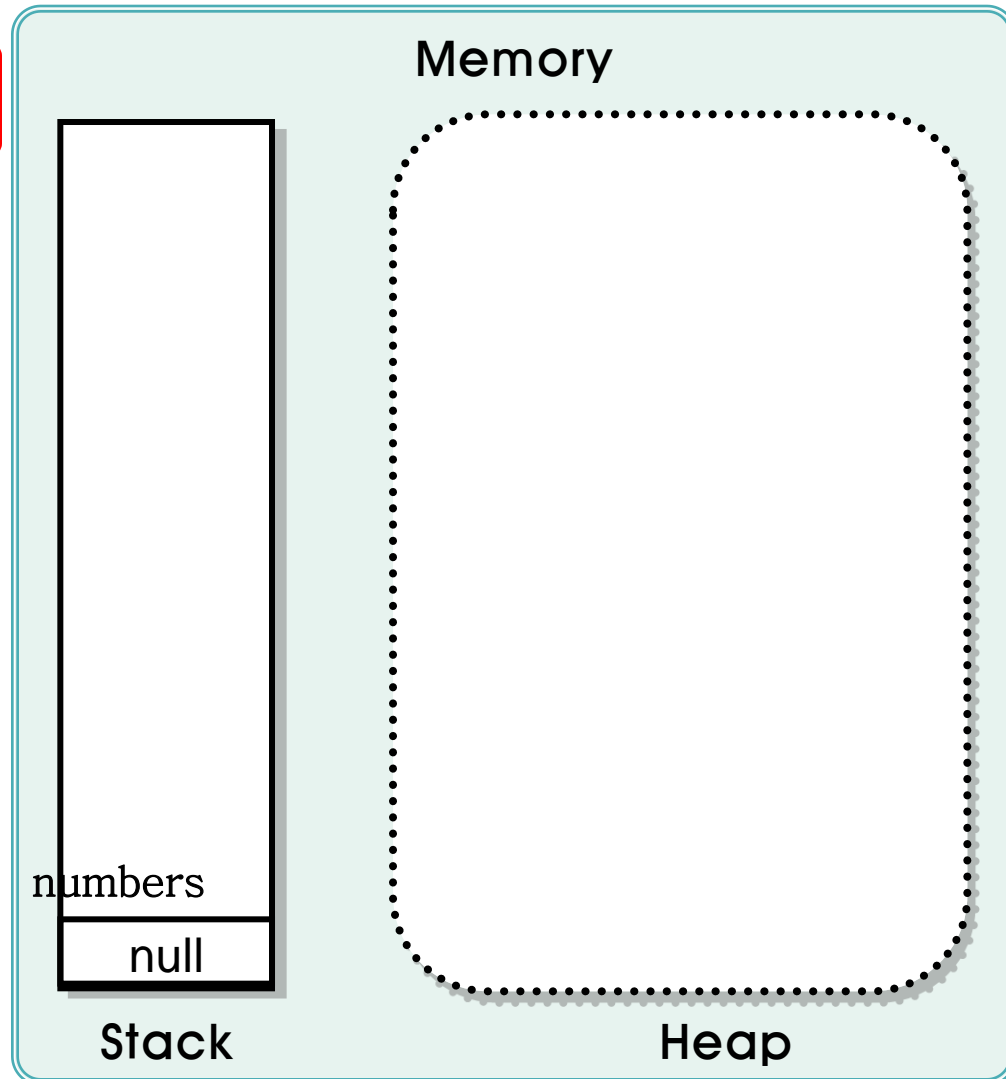
`int a[], b; // a는 Array에 대한 Reference, b는 int 형`

□ 선언

```
int[ ] numbers = null;
```

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

```
for ( int inx = 0 ; inx < numbers.length ; inx ++ ) {  
    numbers[inx] = inx + 1;  
    System.out.println( numbers[inx] );  
}
```

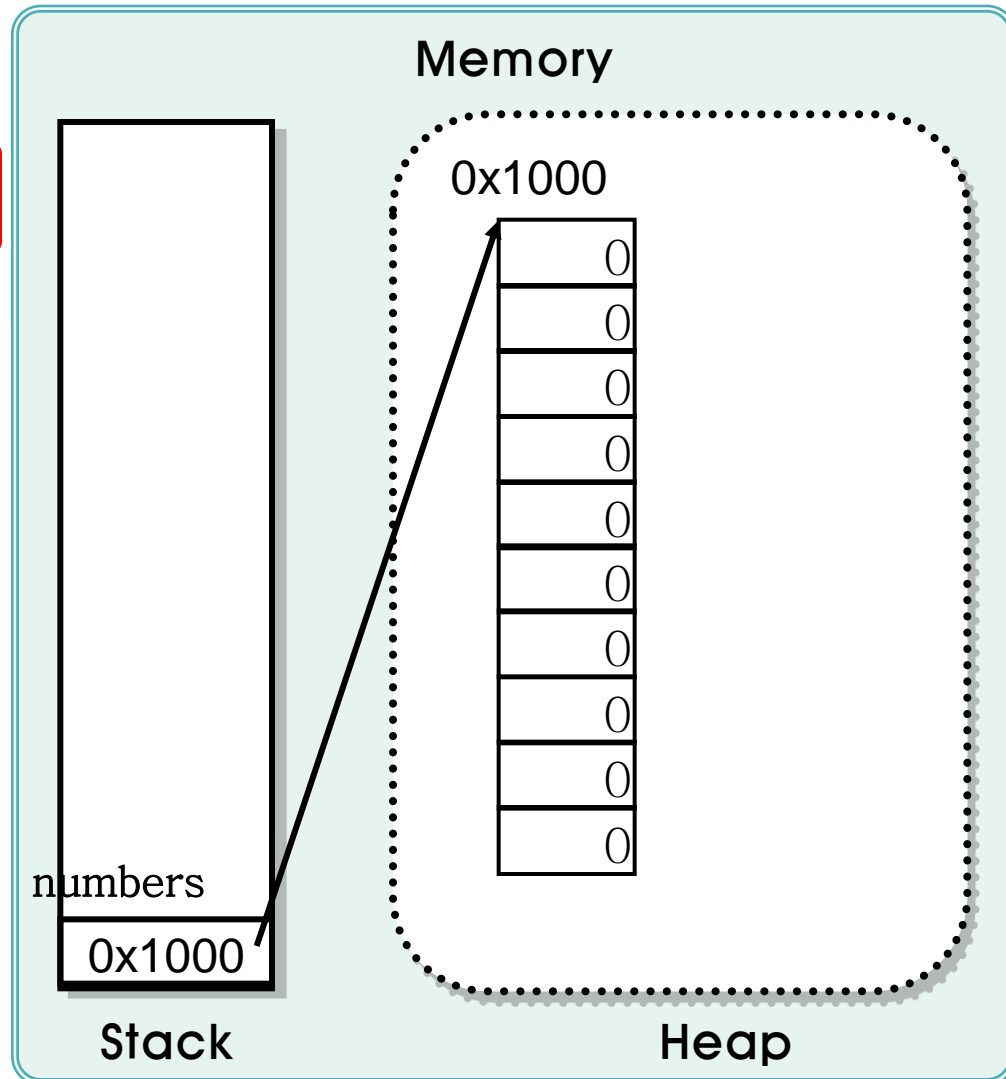


□ 생성

```
int[ ] numbers = null;
```

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

```
for ( int inx = 0 ; inx < numbers.length ; inx ++ ) {  
    numbers[inx] = inx + 1;  
    System.out.println( numbers[inx] );  
}
```

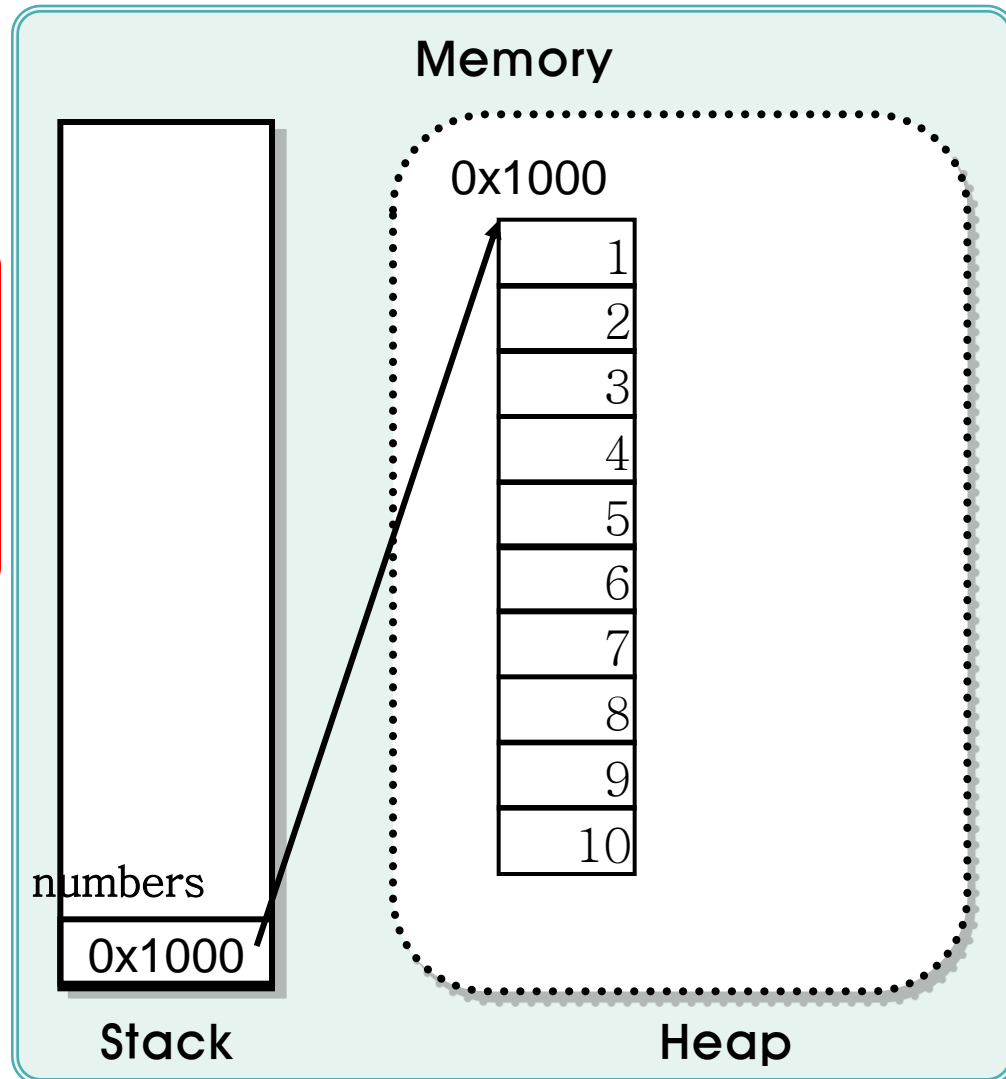


□ 초기화

```
int[ ] numbers = null;
```

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

```
for ( int inx = 0 ; inx < numbers.length ; inx ++ ) {  
    numbers[inx] = inx + 1;  
    System.out.println( numbers[inx] );  
}
```



□ 선언

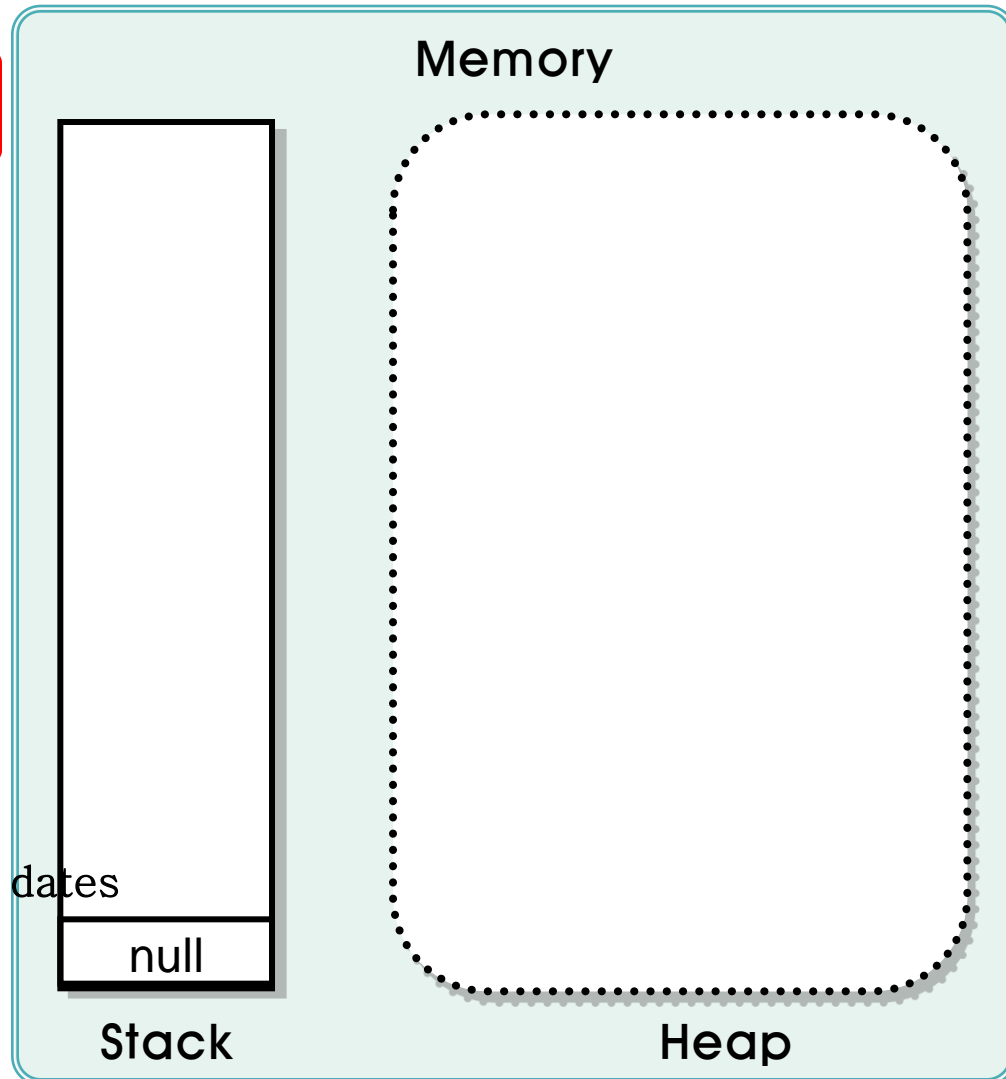
```
Date[ ] dates = null;
```

```
dates = new Date[10];
```

```
dates[0] = new Date( 22, 7, 1964 );
```

```
dates[1] = new Date( 12, 7, 2006 );
```

```
System.out.println( dates[0].getDay() );
```



□ 생성

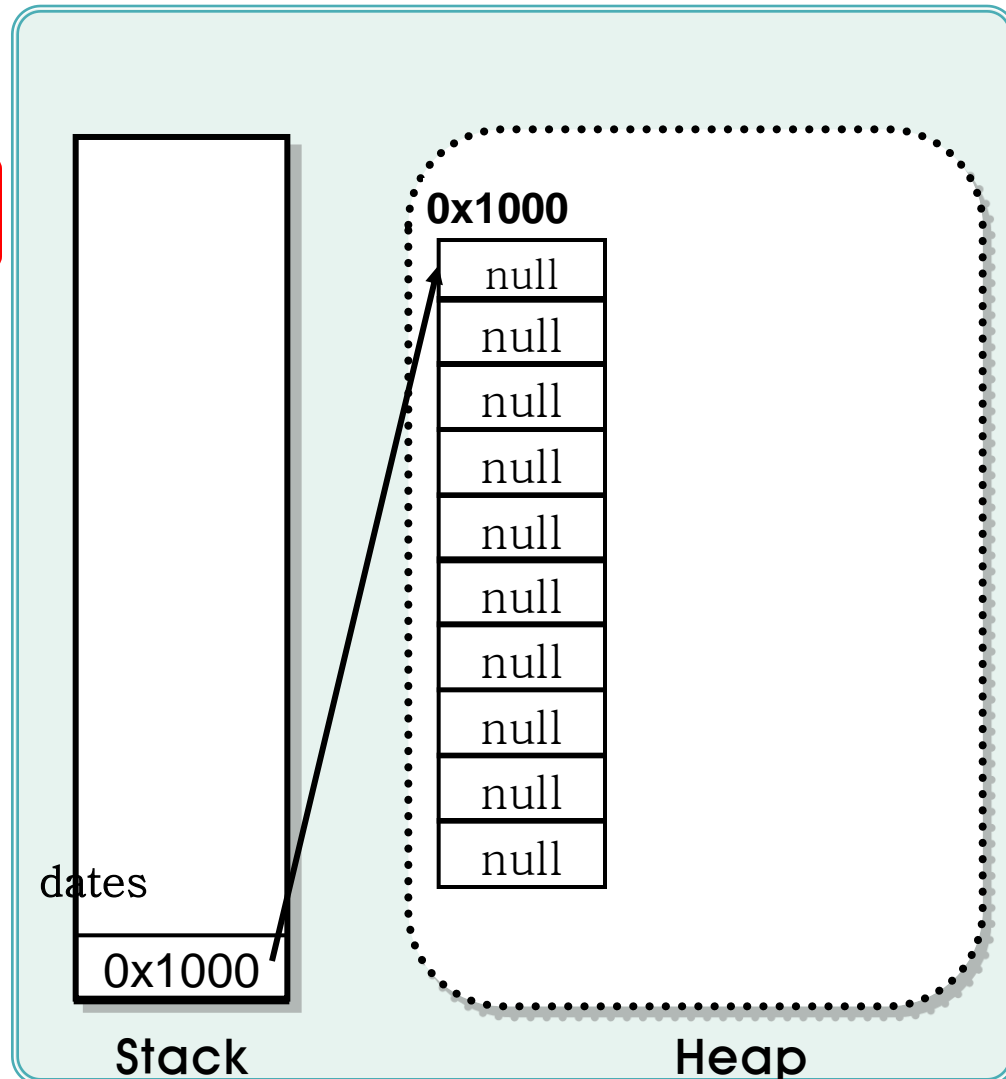
```
Date[ ] dates = null;
```

```
dates = new Date[10];
```

```
dates[0] = new Date( 22, 7, 1964 );
```

```
dates[1] = new Date( 12, 7, 2006 );
```

```
System.out.println( dates[0].getDay() );
```



□ 초기화

```
Date[ ] dates = null;
```

```
dates = new Date[10];
```

```
dates[0] = new Date( 22, 7, 1964 );
```

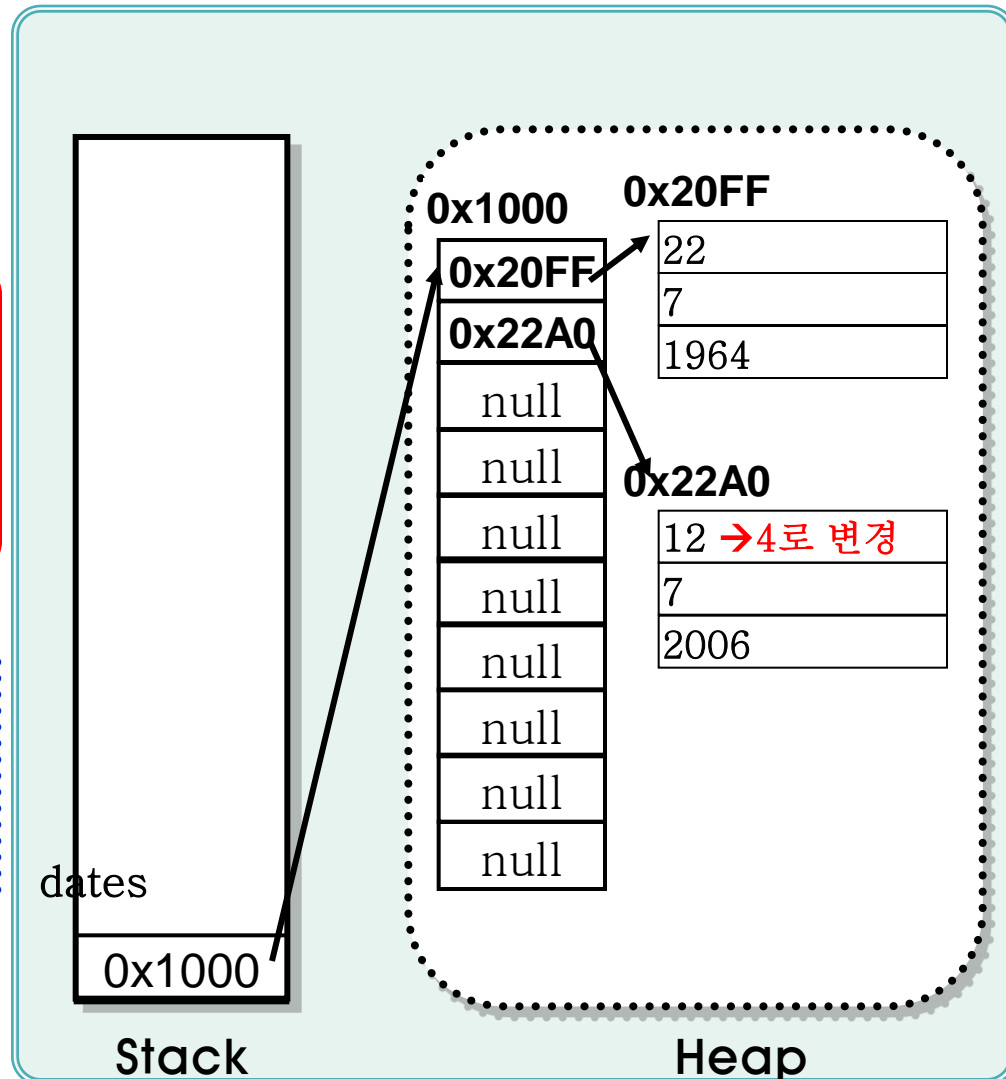
```
dates[1] = new Date( 12, 7, 2006 );
```

```
System.out.println( dates[0].getDay() );
```

```
// access
```

```
dates[1].setDay( 4 );
```


```
System.out.println( dates[1].getDay() );
```



□ 생성 후, 초기화

```
int[] numbers = new int[3];  
numbers[0] = 10;  
numbers[1] = 20;  
numbers[2] = 30;
```

```
MyDate[] date = new MyDate[3];  
date[0] = new MyDate(10, 08, 2005);  
date[1] = new MyDate(11, 08, 2005);  
date[2] = new MyDate(12, 08, 2005);
```



동 일

□ 생성과 동시에 초기화

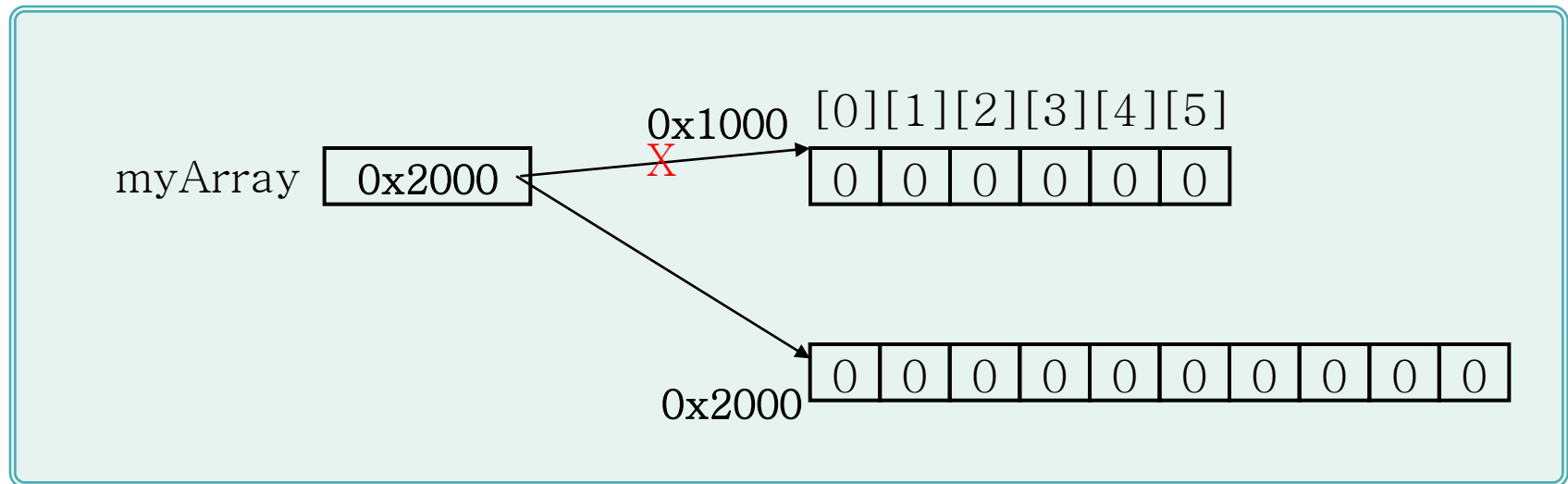
```
int[] numbers = { 1, 2, 3 };
```

```
MyDate[] date =  
    { new MyDate(10, 08, 2005),  
      new MyDate(11, 08, 2005),  
      new MyDate(12, 08, 2005)  
    };
```

□ 배열은 크기를 바꿀 수 없다.

```
int[] myArray = new int[6];
```

```
myArray = new int[10]; //size가 늘어나는 것이 아니라, 새로운 배열  
이 생성
```



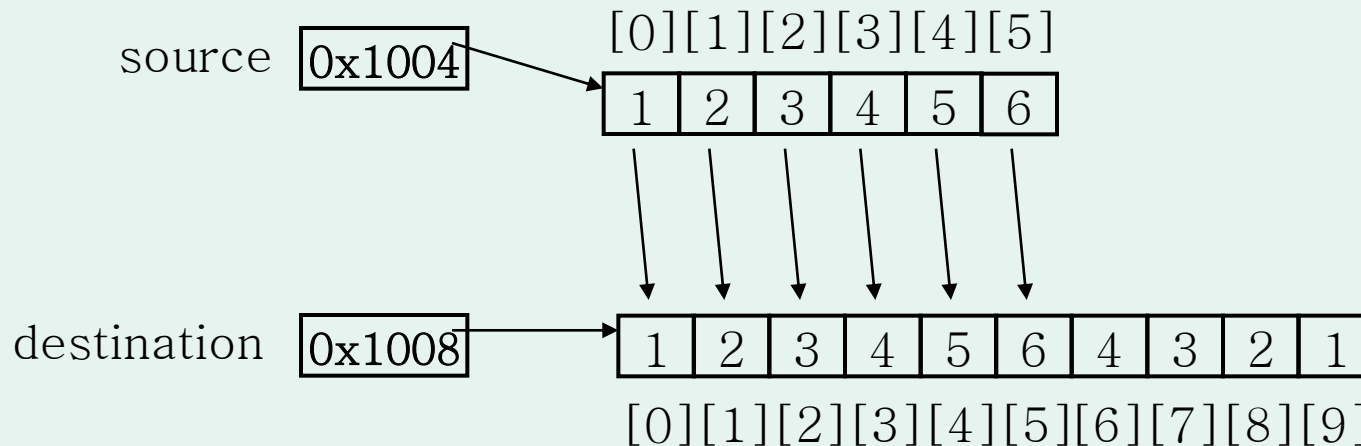
```
//original array
```

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

```
//new larger array
```

```
int[ ] destination = { 10, 9, 8, 7, 6, 5, 4, 3, 2, 1 };
```

```
System.arraycopy( source, 0, destination, 0, source.length );
```



//original array

```
String[ ] str1 = { "A1", "A2", "A3" };
```

//new larger array

```
String[ ] str2 = { "B1", "B2", "B3" };
```

```
System.arraycopy( str1, 0, str2, 0, str1.length );
```

