

Single - Dimentional Array

Preview

- 배열 : 변수들을 묶어서 선언하는 것
- 선언과 접근방법은?

배열이 필요한 이유?

! i. 동일한 자료형의 변수를 여러 개 선언해야 할 경우 묶어서 나란히 선언이 가능하다.

```
int num1, num2, num3;  
int num4, num5, num6;  
int num7, num8, num9;
```

→ `int[] arr = new int[9];`

ii. 순차적 접근이 가능하다.

```
num1=num2=num3=10;  
num4=num5=num6=10;  
num7=num8=num9=10;  
(변수 이름을 일일이 지정을 해줘야 된다.)
```

```
for(int i=0; i<9; i++) {  
    arr[i] = 10;  
}
```

이것이 가능한 이유는 변수 단독선언 시 메모리주소에 순차적으로 나열됨을 보장받을 수 없다.
하지만 배열은 메모리공간에 순차적으로 나란히 할당된다.

Single - Dimentional Array (1차원 배열)

- Java에서의 Array는 둘 이상의 Variable Data를 저장하는 Instance이다.
- 특수한 Instance이므로 선언방법도 독특하다.
- 일반적인 Instance의 선언방법
`Friend frn = new Friend();`

Single - Dimentional Array의 선언

① 5개의 int형 변수를 저장할 수 있는
Array Instance를 생성하고

Data Type[]	Reference Variable Name	=	new	Data Type	[Array Length]
int[]	ref	=	new	int	[5]

② Array Instance를 참조할 수 있는 참조변수를 생성하는데
[]는 Array Instance를 의미한다.

- 재미있는 사실은 위의 선언문장에서 배열길이를 5로 전달하는데 Array Instance가 생성이 되면 Instance의 Member인 변수 length에 5가 저장된다는 사실이다.
- Single - Dimentional Array의 다양한 선언
 - i. 기본자료형

```
int[] arr1 = new int[7];  
double[] arr2 = new double[10];  
boolean[] arr3 = new boolean[6];
```
 - ii. 클래스

Array Instance는 Variable을 저장할 수 있는 Instance인데, Array Instance는 자료형이 Class가 될 수 있다.
(Class는 사용자 정의 데이터 타입이다.)
그러므로 해당 Class의 Instance를 저장할 수 있다.
하지만 Array Instance에 저장되는 것은 인스턴스가 아니고 인스턴스의 메모리주소 참조값이다.

그러므로 해당 Class의 Instance를 저장할 수 있다.

하지만 Array Instance에 저장되는 것은 인스턴스가 아니고 인스턴스의 메모리주소 참조값이다.

Instance 저장 방법

- ① 참조변수를 저장해서, 참조변수의 참조값을 통해 참조하고 있는 Instance의 접근이 가능하다.
- ② 그리고 인스턴스를 생성해서 반환하는 참조값을 저장할 수 도 있다.

```
FruitSeller seller = new FruitSeller();
```

```
FruitSeller[] arr4 = new FruitSeller[5];
```

```
FruitBuyer[] arr5 = new FruitBuyer[8];
```

```
① arr4[0] = seller;
```

```
② arr5[0] = new FruitBuyer();
```

Single - Dimentional Array의 접근

배열요소는 각각 지정된 Index값이 있다.

Index는 0부터 시작하고 n번째 요소의 index는 (n-1)이다.

```
// Primitive Array
public static void main(String[] args) {
    int[] arr = new int[3];
```

```
    arr[0] = 1;
```

```
    arr[1] = 2;
```

```
    arr[2] = 3;
```

index[0]	index[1]	index[2]
1	2	3

```
    int sum = arr[0] + arr[1] + arr[2];
```

```
    System.out.println(sum);
```

```
}
```

```
// Instance Array
```

```
public static void main(String[] args) {
    String[] strArr = new String[3];
```

```
    strArr[0] = new String("Java");
```

```
    strArr[1] = new String("Flex");
```

```
    strArr[2] = new String("Ruby");
```

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

```
    }
```

```
}
```

Single - Dimentional Array의 선언과 동시에 초기화

- i. `int[] arr = new int[3];`
 - ii. `int[] arr = new int[3]{1,2,3};`
 - iii. `int[] arr = new int[]{1,2,3};`
// 초기화 데이터의 갯수로 길이의 계산이 가능하기 때문에 길이정보 생략가능!
 - iv. `int[] arr = {1,2,3};`
// 이렇게 더 생략이 가능하다.
-