

섹션1. 2. 자바의 핵심 - 객체지향 프로그래밍

배열과 ArrayList(1)

배열 :

(선언)

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

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

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

```
numbers[1]=2;
```

```
numbers[2]=3;
```

(초기화)

int => 0

double => 0.0

객체=> null

ArrayList :

객체배열 사용하기(2)

기본 자료형 배열 vs String/ 객체 배열

: 기본 자료형 배열은 배열 선언 후 바로 값을 담아서 사용할 수 있지만, 객체 배열은 객체배열 생성후 자리별 객체를 각각 생성하는 2단계 과정을 거친다.

1st. 객체 배열 선언(주소 자리 생성)

2nd. 자리별 객체 생성

배열 복사하기

System.arraycopy(원본배열, 원본배열 복사 시작 위치, 복사할 배열, 복사할 배열 복사 시작 위치, 복사 길이); // 얕은 복사, 서로 같은 객체 가리킴

for문 객체 생성 및 복사 // 깊은 복사, 서로 다른 객체 가리킴

향상된 for문

: 배열 요소의 처음부터 끝까지 모든 요소를 참조할 때 편리한 반복문

```
for(배열 요소 자료형 변수 선언 : 배열){  
반복 실행할 코드;  
}
```

다차원배열(3)

: 2차원 이상의 배열

ArrayList 클래스(4)

: 자바에서 제공되는 객체 배열이 구현된 클래스, 여러 메서드와 속성 등을 사용하여 객체 배열을 편리하게 관리할 수 있음

(선언)

```
ArrayList<객체> list = new ArrayList<객체>(); // 객체 생략 가능, 생략시 디폴트는  
Object
```

(주요 메서드)

ArrayList 클래스 주요 메서드

메서드	설명
boolean add(E e)	요소 하나를 배열에 추가합니다. E는 요소의 자료형을 의미합니다.
int size()	배열에 추가된 요소 전체 개수를 반환합니다.
E get(int index)	배열의 index 위치에 있는 요소 값을 반환합니다.
E remove(int index)	배열의 index 위치에 있는 요소 값을 제거하고 그 값을 반환합니다.
boolean isEmpty()	배열이 비어 있는지 확인합니다.

cf. 함수 살펴보기 : 함수명, 파라미터, 리턴타입