

# 프로그래밍 언어 응용

chapter07

## 배열

제공된 자료는 훈련생의 수업을 돕기 위한 것으로, 타인과 공유하시면 안됩니다.

# Contents

part.1

배열

part.2

다차원 배열

part.3

ArrayList 클래스

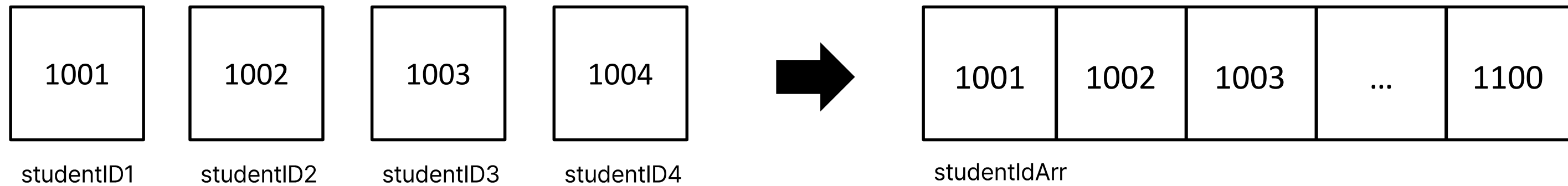
part.4

배열 응용 프로그램 만들기

**배열이란?**

- 자료형이 동일한 자료형의 데이터를 순차적으로 저장하는 자료구조이다.
- 대량의 데이터를 관리할 때 사용한다.

학생의 학번

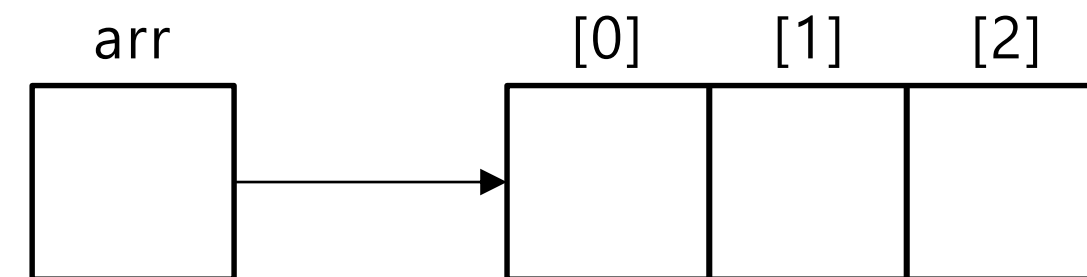


배열을 선언할 때는 [ ] 대괄호 기호를 사용하여 배열 형식을 나타낸다.

배열을 생성할 때는 "new" 키워드를 사용하고, 배열의 길이를 지정해야 한다.

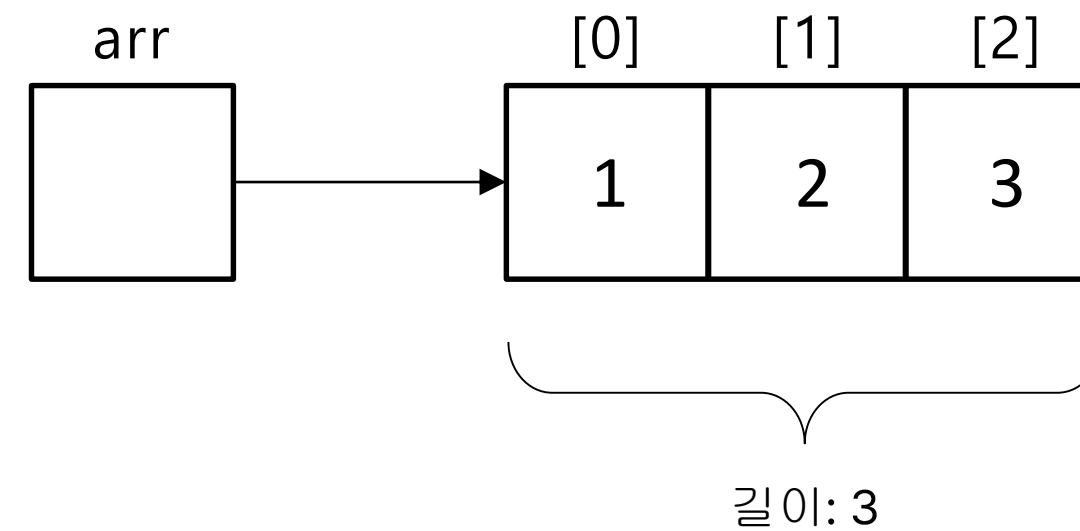
자료형[ ] 배열이름 = new 자료형[크기];

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



배열을 선언하고 동시에 값을 초기화할 때, { } 안에 값을 나열하여 초기화할 수 있다.  
이 때 배열의 길이는 중괄호 안에 나열된 요소의 개수에 따라 자동으로 결정된다.  
따라서 "new int[ ]" 코드를 생략할 수 있다.

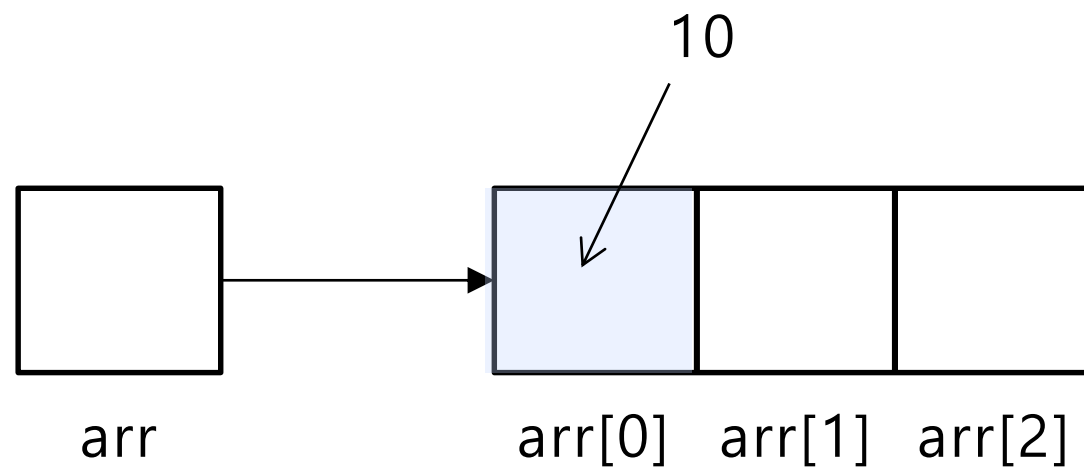
```
int[ ] arr = new int[ ] { 1, 2, 3 };
```



배열의 각 요소에 값을 넣거나 값을 가져올 때는 [ ] 인덱스 연산자를 사용한다.

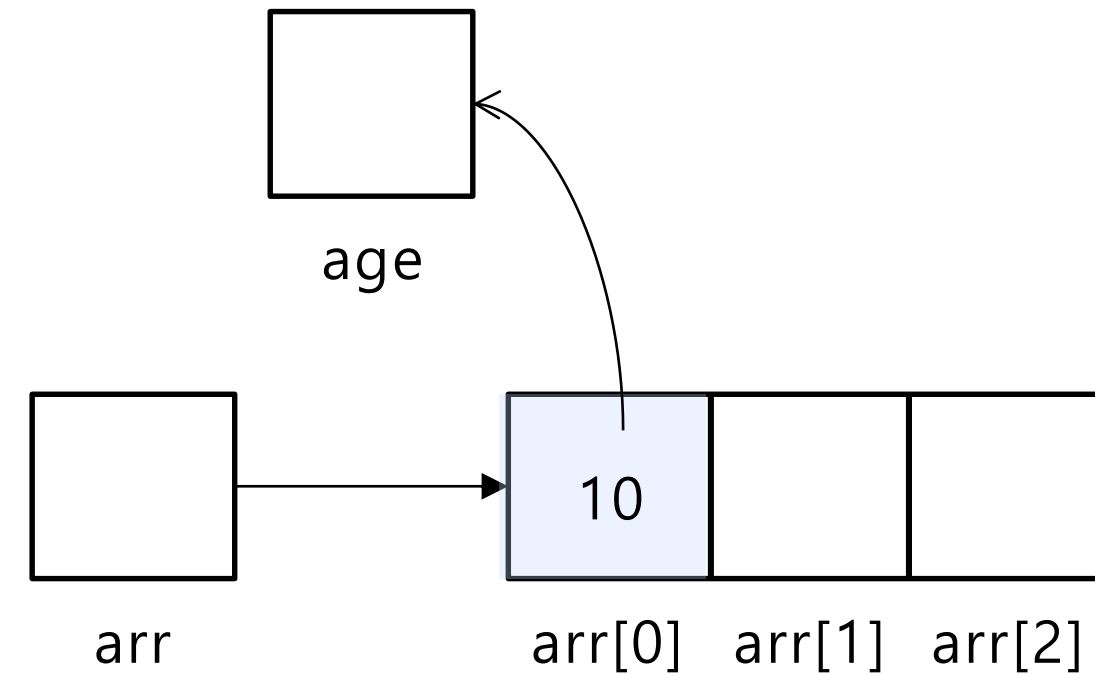
인덱스 연산자는 배열 요소의 위치를 찾아 주는 역할을 한다.

배열의 길이가 n이라면, 인덱스는 0부터 시작하여 n-1까지의 값을 가진다.



```
arr[0] = 10;
```

인덱스



```
int age = arr[0];
```

**객체배열이란?**

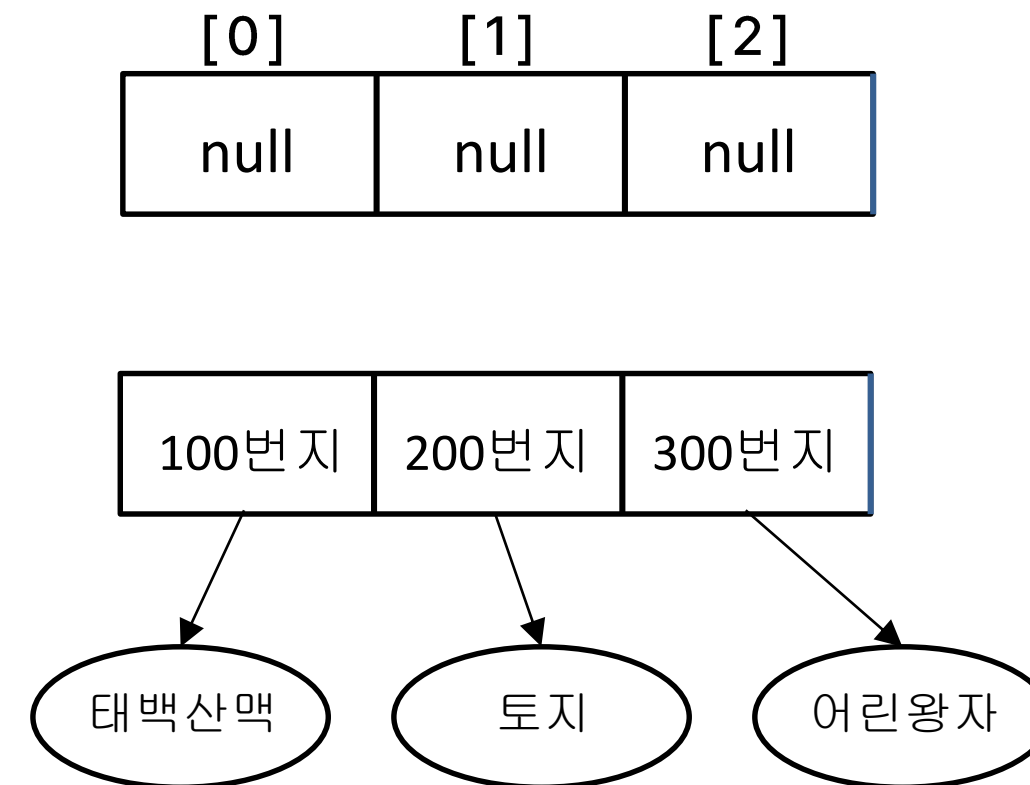
- 객체배열은 참조자료형을 저장하는 배열을 의미한다.
- 이 배열은 실제 객체가 아닌 주소를 저장한다.
- 객체 배열의 각 요소에는 인스턴스를 생성하고 저장할 수 있다.

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

```
arr[0] = new Book("태백산맥");
```

```
arr[1] = new Book("토지");
```

```
arr[2] = new Book("어린왕자");
```

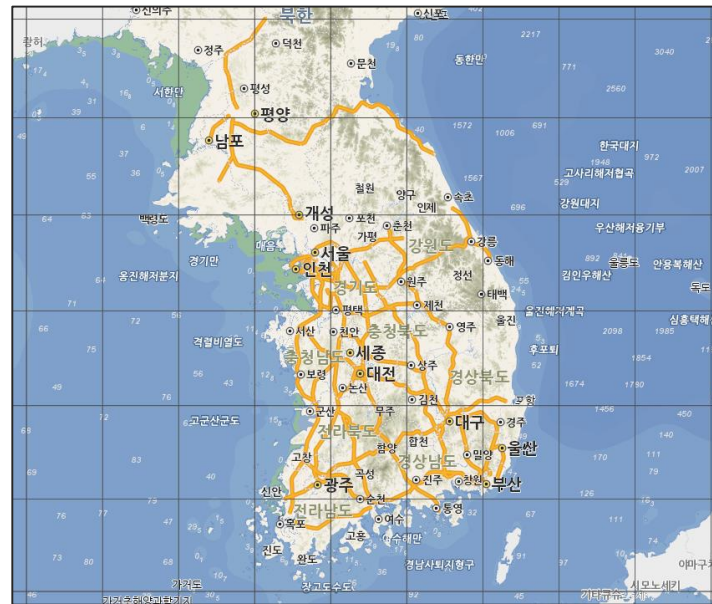


# 다차원배열

## 다차원배열은 무엇일까?

### 다차원 배열이란?

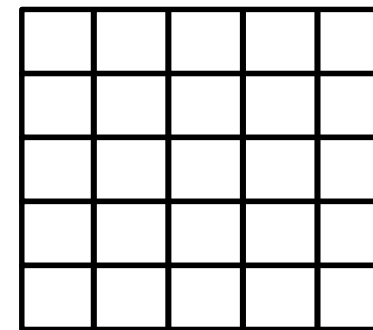
- 다차원배열은 테이블 형태의 데이터를 저장하거나, 평면이나 공간을 표현할 때 사용한다.
- 이차원 이상으로 구현된 배열을 다차원배열이라고 한다.
- 다차원 배열은 일차원 배열을 여러 개 조합하여 만들 수 있다.



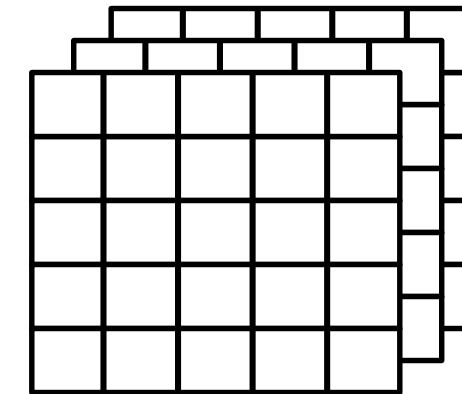
격자 형태의 지도



1차원 배열



2차원 배열

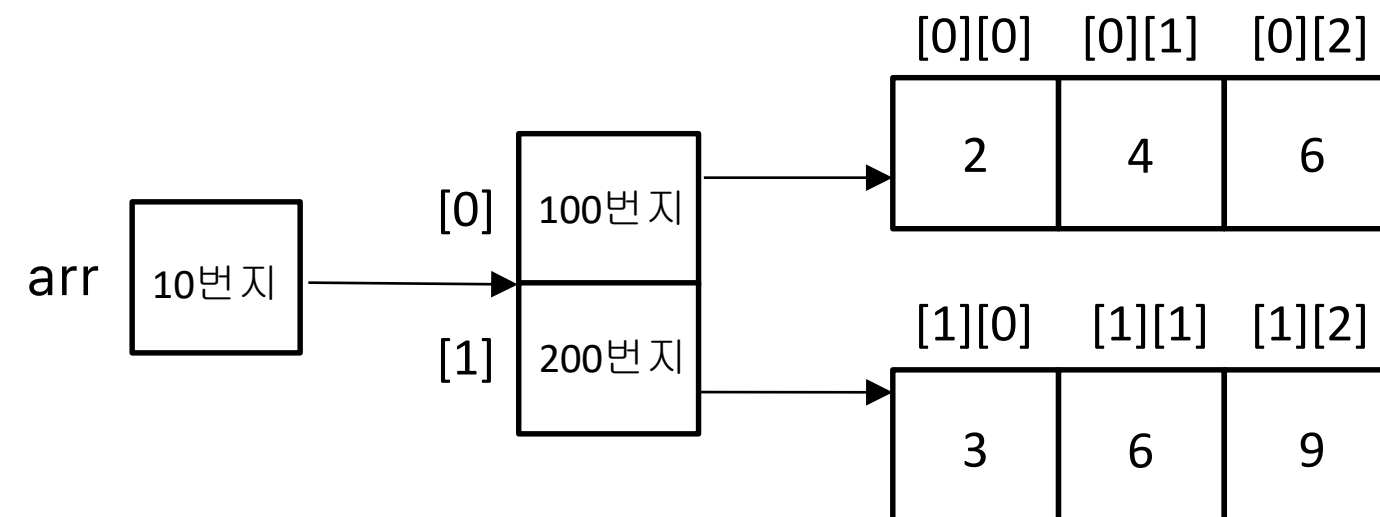
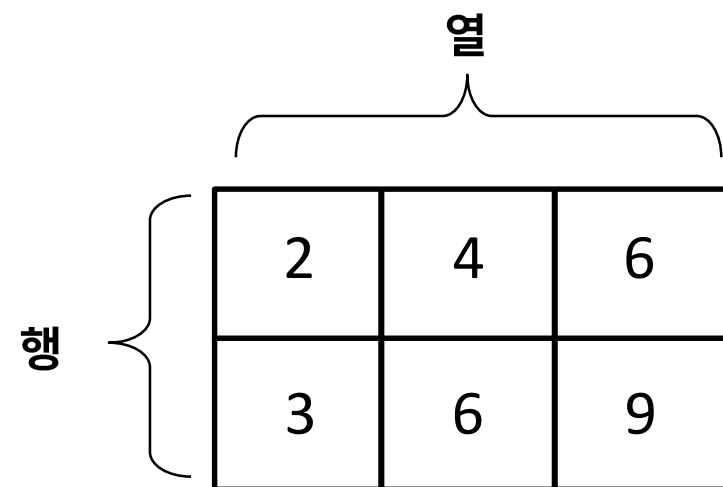


3차원 배열



이차원 배열은 수학의 행렬 구조와 비슷하고, 일차원 배열을 여러 개 묶어서 만들 수 있다.  
이러한 구조를 이용하여 행렬 형태의 데이터를 표현할 수 있다.

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



3개 크기의 1차원 배열

# ArrayList 클래스

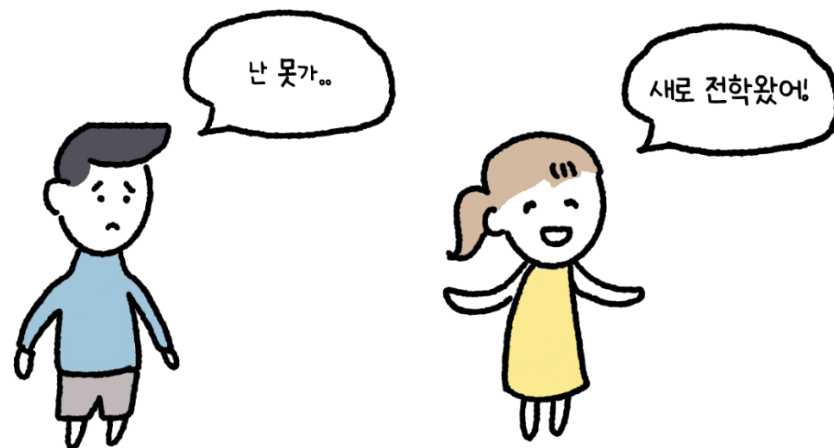
## 기본배열의 단점과 ArrayList

### ArrayList란?

- 리스트는 순서가 있는 데이터의 모음으로, 배열과 비슷한 기능을 제공한다.
- 리스트는 요소의 순서를 유지하면서, 동적으로 요소를 추가하거나 삭제할 수 있다.

### 배열과 리스트의 차이점

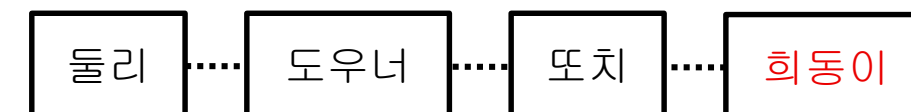
- 한번 생성된 배열의 크기는 변경할 수 없고, 연속된 메모리 공간을 사용한다.
- 리스트는 요소가 추가 또는 삭제됨에 따라 동적으로 크기를 변경할 수 있고, 메모리 공간을 유연하게 사용한다.



학생목록(배열)

둘리	도우너	또치
----	-----	----

학생목록(리스트)



# ArrayList 클래스

## ArrayList 클래스의 주요 메소드

메소드	설명
<code>boolean add(E e)</code>	요소 하나를 배열에 추가한다
<code>int size()</code>	리스트의 크기를 반환한다
<code>E get(int index)</code>	리스트의 index 위치에 있는 요소를 반환한다
<code>E remove(int index)</code>	리스트의 index 위치에 있는 요소를 삭제한다
<code>boolean isEmpty()</code>	리스트가 비어 있는지 확인한다

### 리스트의 특징

- 리스트를 생성할 때는 <>기호 안에 자료형을 선언해야 하고, 자료형은 클래스 타입만 사용 할 수 있다.
- 리스트는 요소가 추가되거나 삭제될 때, 동적으로 크기가 변경된다.
- 리스트는 데이터를 관리할 수 있는 다양한 속성과 메소드를 제공한다.

```
ArrayList<Integer> list = new ArrayList<Integer>();
```

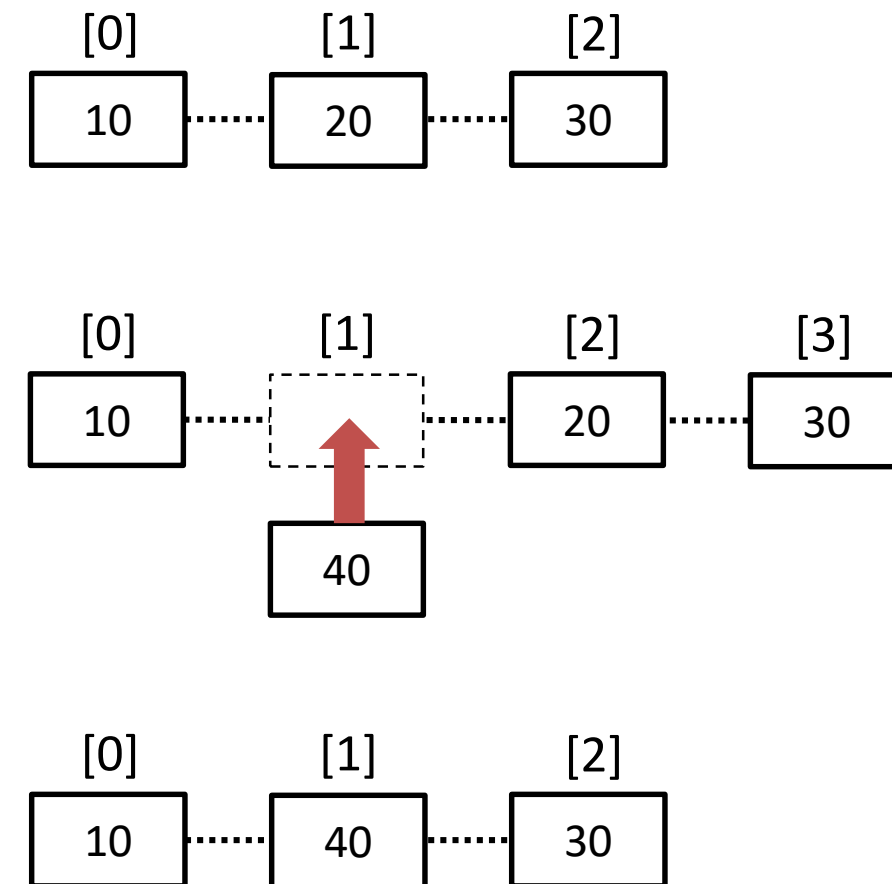
```
list.add(10);
```

```
list.add(20);
```

```
list.add(30);
```

```
list.add(1, 40);
```

```
list.remove(2);
```



1. 학생 클래스 설계
  - 학생의 기본정보와 학생이 수강하는 과목들을 저장한다.
2. 과목 클래스 설계
  - 과목의 이름과 성적을 저장한다.
3. 프로그램 구현
  - 학생 객체를 생성하고, 해당 학생이 수강하는 과목을 추가한다.
  - 학생마다 수강하는 과목의 수가 다를 수 있다.

학생1



이름: 돌리  
 학년: 1  
 수강과목: 국어, 수학 (2)

학생2



이름: 도우너  
 학년: 3  
 수강과목: 국어, 영어, 과학, 사회 (4)

학생 클래스

이름  
 학년  
 수강 과목 리스트

과목 클래스



과목의 이름  
 점수