



스마트인재개발원
Smart Human Resources Development

임 경 남 연구원



학습목표

1. Collection에 대해 이해한다.
2. ArrayList를 생성하고 다양한 기능을 사용한다.
3. ArrayList를 활용하여 뮤직플레이어를 만든다.



객체

여러 마리의 **포켓몬**을

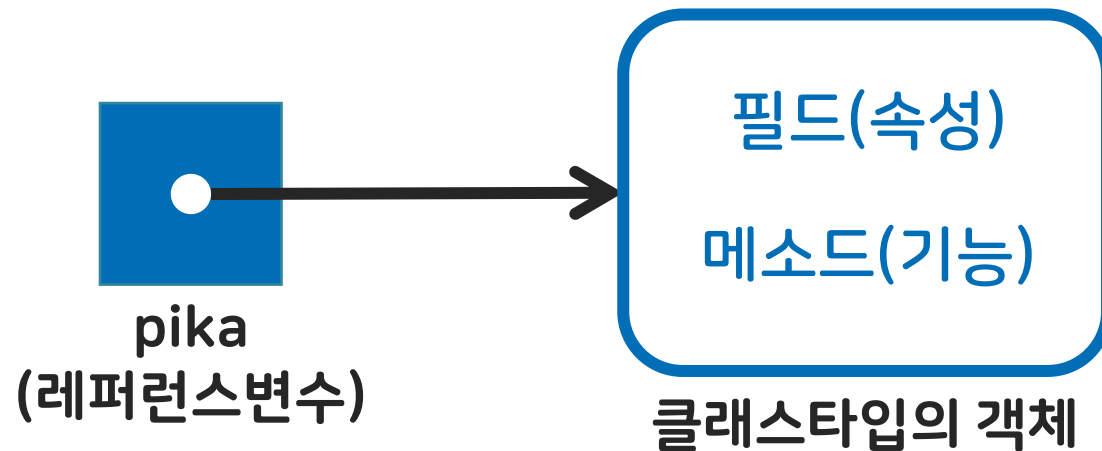
객체 배열 **하나의 가방**에 담아서

관리하는 것이 편하지 않을까?

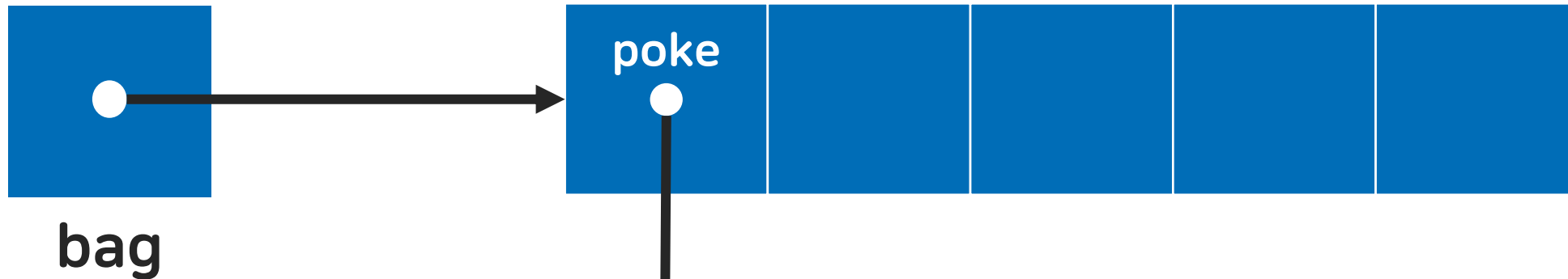
- `Poke[] bag = new Poke[5];`



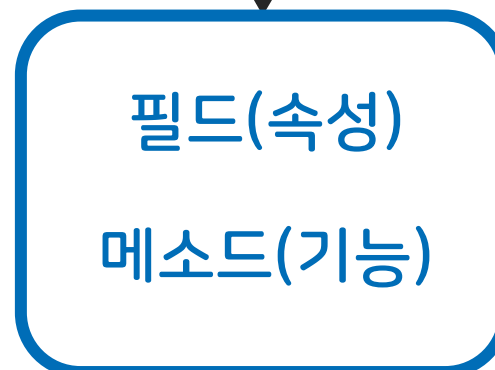
`Poke pika = new Poke();`



- `Poke[] bag = new Poke[5];`
- `bag[0] = new Poke();`



Tip!
참조의 참조!
이차원 배열과 비슷하죠?



클래스타입의 객체

만약 포켓몬을 하나 더
추가하고 싶다면?

Collection

- 요소(Element)라고 불리는 **가변 개수의 객체들의 집합**
- 여러 개의 **객체**를 보관할 수 있게 만들어진 클래스들의 집합
- 고정 크기의 배열을 다루는 불편함 해소

특징

기본타입의 값을 넣으면
자동으로 wrapper 클래스 변환

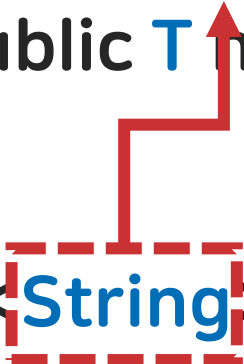
1. 요소의 개수에 따라 **자동 크기 조절**
2. 요소의 추가,삭제에 따른 **요소의 이동 자동 관리**
3. **제네릭(generic) 기법**으로 구현

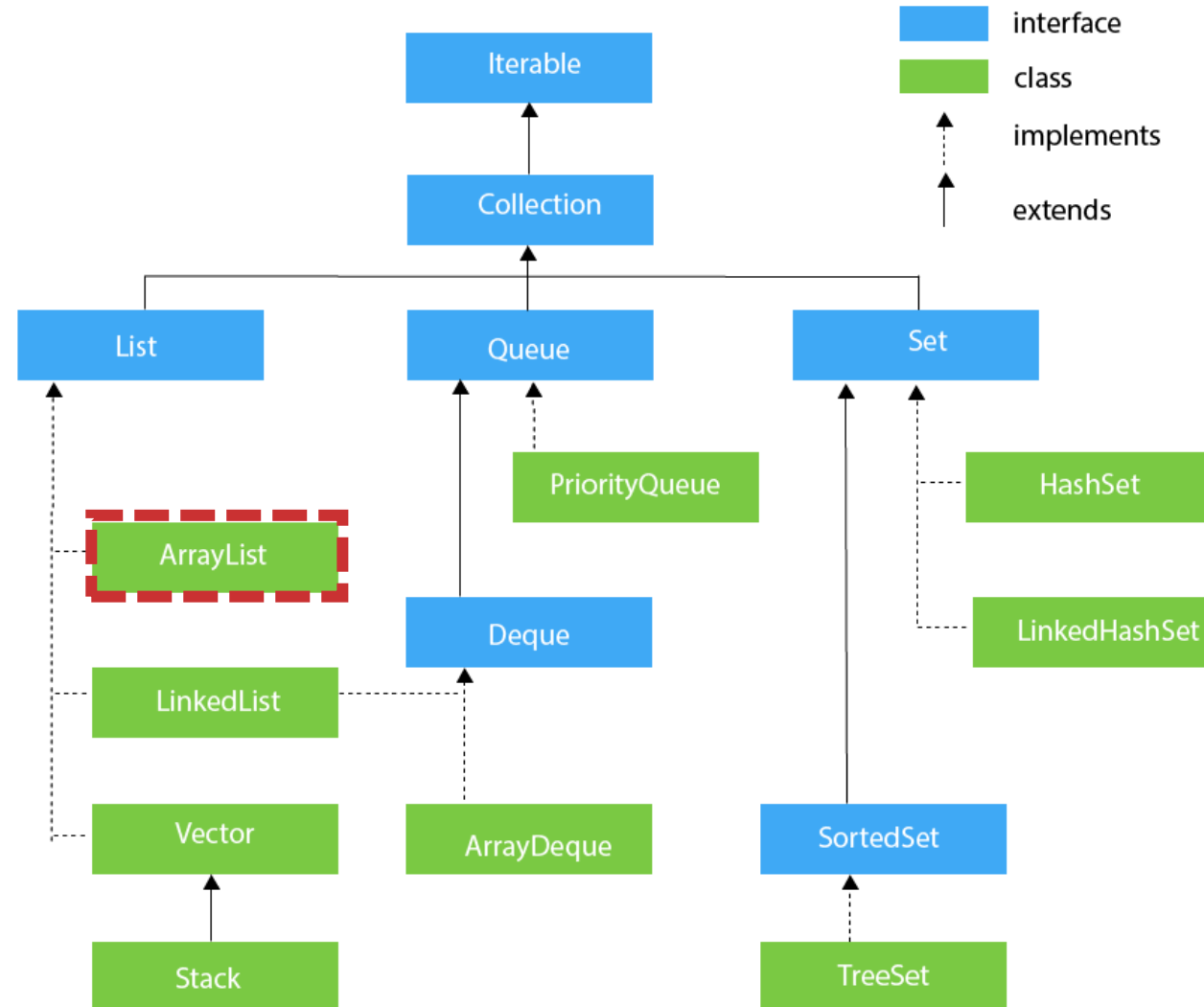
- 제네릭(Generic)

- 클래스 내부에서 사용할 데이터 타입을 외부에서 지정하는 기법

```
class Person<T>{  
    public T name;  
}
```

```
Person<String> person = new Person<String>;
```





ArrayList<자료형> 변수명 = new ArrayList<자료형>[()];

클래스이기때문에,
생성자 반드시 사용

- java.util.ArrayList 가변 크기 배열을 구현한 **클래스**
- ArrayList에 객체 삽입/삭제
 - 리스트의 맨 뒤에 객체 추가 : 공간이 부족하면 자동으로 늘림
 - 리스트의 중간에 객체 삽입 : 삽입된 뒤의 객체는 뒤로 밀림
 - 임의의 위치의 객체 삭제 : 객체 삭제 후 자동 자리 이동

메소드	설명
boolean add(E e)	ArrayList의 맨 뒤에 요소 추가
void add(int index, E element)	지정된 인덱스에 지정된 객체를 삽입
void clear()	ArrayList의 모든 요소 삭제
E get(int index)	지정된 인덱스의 요소 반환
E remove(int index)	지정된 인덱스의 요소 삭제
boolean remove(Object o)	지정된 객체와 같은 첫 번째 요소를 ArrayList에서 삭제
int size()	ArrayList가 포함하는 요소의 개수 반환

1. 아래와 같은 메뉴를 출력하고, [4] 종료를 선택하기 전까지 프로그램을 반복시켜주세요.

```
Markers Properties Servers Data Source Explorer Snippets Console ×
test [Java Application] C:\Users\cloud\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.2.v20220201-1208\jre\bin\javaw.exe (2022. 9
[1]노래추가 [2]노래삭제 [3]노래조회 [4]종료 >> 1
[1]노래추가 [2]노래삭제 [3]노래조회 [4]종료 >> 4
프로그램을 종료합니다.
```

2. [1]노래추가를 선택하면 노래 목록을 보여준 뒤 노래를 어느 위치에 추가할 것인지 입력 받으세요.

```
Markers Properties Servers Data Source Explorer Snippets Console ×
test [Java Application] C:\Users\cloud\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.2.v20220201-1208\jre\bin\javaw.exe (2022. 9.
[1]노래추가 [2]노래삭제 [3]노래조회 [4]종료 >> 1
=====현재 재생 목록=====
재생목록이 없습니다.
=====
[1]마지막 위치에 추가 [2]원하는 위치에 추가 >>
```

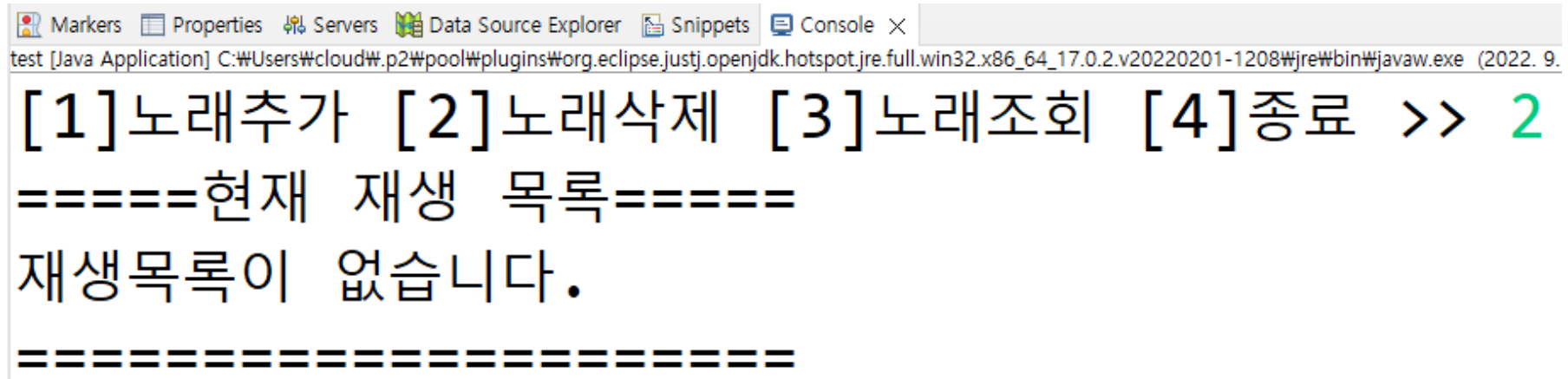
3. [1]마지막 위치에 추가를 선택하면 추가할 노래를 입력 받고,
ArrayList에 입력 받은 노래를 추가하세요.

```
Markers Properties Servers Data Source Explorer Snippets Console ×
test [Java Application] C:\Users\cloud\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.2.v20220201-1208\jre\bin\javaw.exe (2022. 9
[1]노래추가 [2]노래삭제 [3]노래조회 [4]종료 >> 1
=====현재 재생 목록=====
재생목록이 없습니다.
=====
[1]마지막 위치에 추가 [2]원하는 위치에 추가 >> 1
추가할 노래 입력 >> LoveDive
```

4. 원하는 위치에 추가를 선택할 시 노래와 위치를 입력 받고 ArrayList의 입력 받은 위치 인덱스에 노래를 추가하세요.

```
[1]노래추가 [2]노래삭제 [3]노래조회 [4]종료 >> 1
=====현재 재생 목록=====
1. LoveDive
=====
[1]마지막 위치에 추가 [2]원하는 위치에 추가 >> 2
추가할 노래 입력 >> 살짝설렘
추가할 위치 입력 >> 0
[1]노래추가 [2]노래삭제 [3]노래조회 [4]종료 >> 1
=====현재 재생 목록=====
1. 살짝설렘
2. LoveDive
=====
```

5. [2]노래 삭제를 선택하면 재생 목록을 보여주고, 노래가 없을 시에는
“재생 목록이 없습니다.” 라는 문장을 출력하세요.



```
test [Java Application] C:\Users\cloud\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.2.v20220201-1208\jre\bin\javaw.exe (2022. 9.  
[1]노래추가 [2]노래삭제 [3]노래조회 [4]종료 >> 2  
=====현재 재생 목록=====  
재생목록이 없습니다.  
=====
```

6. 노래가 있을 시에는 어떤 방법으로 삭제할 것인지 입력 받으세요.

[1]노래추가 [2]노래삭제 [3]노래조회 [4]종료 >> 2

=====현재 재생 목록=====

1. 살짝설렘

2. LoveDive

=====

[1]선택삭제 [2]전체삭제 >>

7. 선택삭제를 입력할 시 삭제할 노래를 선택하고 실제 ArrayList에서 삭제하세요.
전체삭제를 입력할 시 ArrayList안에 있는 모든 요소를 삭제하세요.

[1]선택삭제 [2]전체삭제 >> 1

삭제할 노래 선택 >> 1

노래가 삭제되었습니다.

[1]노래추가 [2]노래삭제 [3]노래조회 [4]종료 >> 2

|=====현재 재생 목록=====

1. LoveDive

=====

8. 노래조회를 선택하면 아래와 같이 ArrayList안에
저장된 모든 노래를 출력하세요.

[1]노래추가 [2]노래삭제 [3]노래조회 [4]종료 >> 3

=====현재 재생 목록=====

1. 살짝설렘
2. LoveDive
3. Attention

=====



다음시간에 배울 내용

상속(Inheritance)