

7장. 객체 배열과 ArrayList



ArrayList



객체 배열 만들기

■ 객체 배열

동일한 기본 자료형(int 등) 변수 여러 개를 배열로 사용할 수 있듯이 참조 자료형 변수도 여러 개를 배열로 사용할 수 있다.

```
public class Book {  
    String bookName;  
    String author;  
  
    public Book(String bookName, String author) {  
        this.bookName = bookName;  
        this.author = author;  
    }  
  
    public void showBookInfo() {  
        System.out.println(bookName + ", " + author);  
    }  
}  
  
public String getBookName() {  
    return bookName;  
}  
  
public void setBookName(String bookName) {  
    this.bookName = bookName;  
}  
  
public String getAuthor() {  
    return author;  
}  
  
public void setAuthor(String author) {  
    this.author = author;  
}
```



객체 배열

■ 객체 배열 만들기

- 배열만 생성한 경우 요소는 null로 초기화 됨

```
public class BookArray {  
  
    public static void main(String[] args) {  
        Book[] library = new Book[5];  
  
        for(int i = 0; i < library.length; i++) {  
            System.out.println(library[i]);  
        }  
    }  
}
```

library[0] library[1] library[2] library[3] library[4]

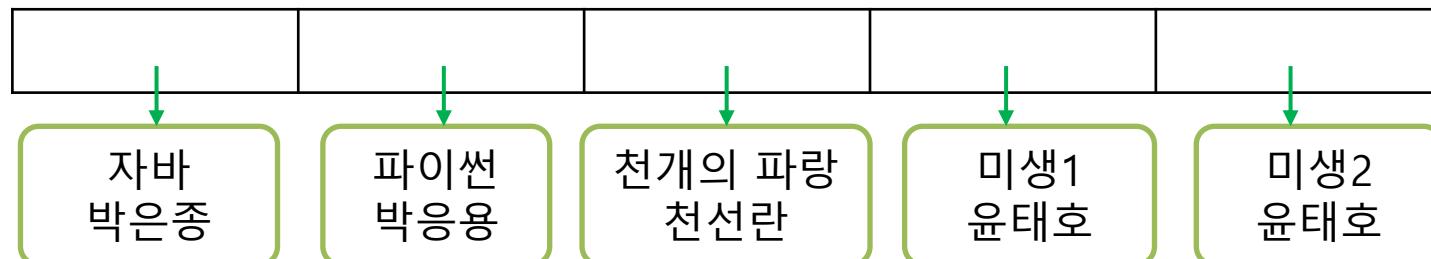
null	null	null	null	null
------	------	------	------	------



객체 배열 만들기

```
//배열에 저장  
library[0] = new Book("자바프로그래밍 입문", "박은종");  
library[1] = new Book("점프 투 파이썬", "박응용");  
library[2] = new Book("천개의 파랑", "천선란");  
library[3] = new Book("미생1", "윤태호");  
library[4] = new Book("미생2", "윤태호");  
  
//출력  
for(int i=0; i<library.length; i++) {  
    library[i].showBookInfo();  
}  
  
//메모리 주소 출력  
for(Book book : library) {  
    System.out.println(book);  
}
```

library[0] library[1] library[2] library[3] Library[4]



객체 배열 복사하기

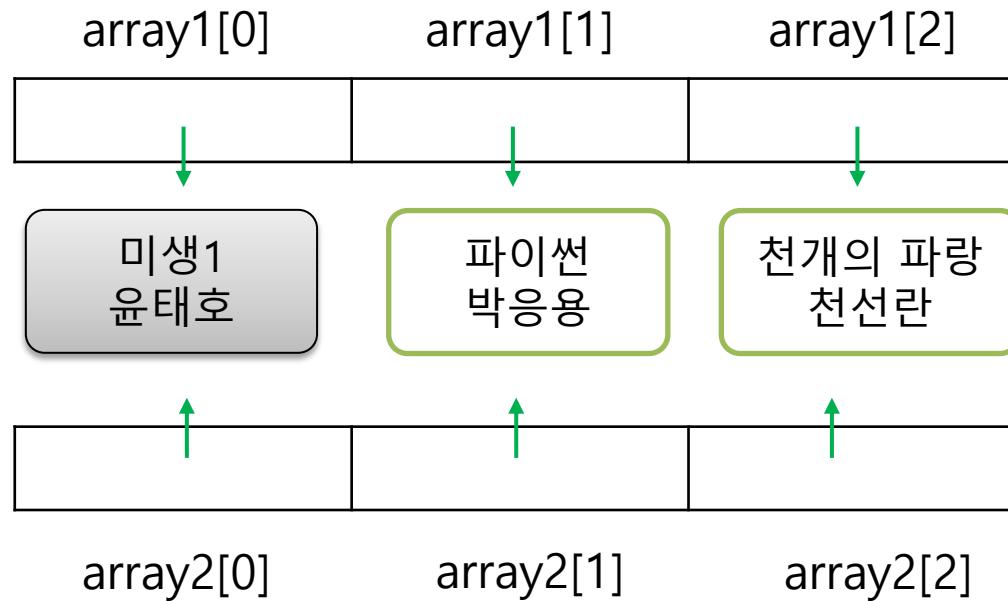
❖ 객체 배열 복사하기

```
public class ObjectCopy1 {  
  
    public static void main(String[] args) {  
        Book[] array1 = new Book[3];  
        Book[] array2 = new Book[3];  
  
        array1[0] = new Book("자바프로그래밍 입문", "박은종");  
        array1[1] = new Book("점프 투 파이썬", "박응용");  
        array1[2] = new Book("천개의 파랑", "천선란");  
  
        //배열 복사  
        for(int i = 0; i < array1.length; i++) {  
            array2[i] = array1[i];  
        }  
  
        //System.arraycopy(array1, 0, array2, 0, 3);  
  
        System.out.println("==== array1 출력 ====");  
        for(int i = 0; i < array1.length; i++) {  
            array1[i].showBookInfo();  
        }  
  
        System.out.println("==== array2 출력 ====");  
        for(Book book : array2) {  
            book.showBookInfo();  
        }  
    }  
}
```



객체 배열 – 얕은 복사

- 객체 배열의 얕은 복사(shallow copy)



객체 배열 – 얕은 복사

■ 객체 배열의 얕은 복사

```
public class ShallowCopy {  
    public static void main(String[] args) {  
        // 객체 배열의 얕은 복사  
        Book[] array1 = new Book[3];  
        Book[] array2 = new Book[3];  
  
        array1[0] = new Book("자바프로그래밍 입문", "박은종");  
        array1[1] = new Book("점프 투 파이썬", "박응용");  
        array1[2] = new Book("천개의 파랑", "천선란");  
  
        //array1의 첫번째 요소값 수정  
        array1[0].setBookName("미생1");  
        array1[0].setAuthor("윤태호");  
    }  
}
```



객체 배열 – 얕은 복사

■ 객체 배열의 얕은 복사

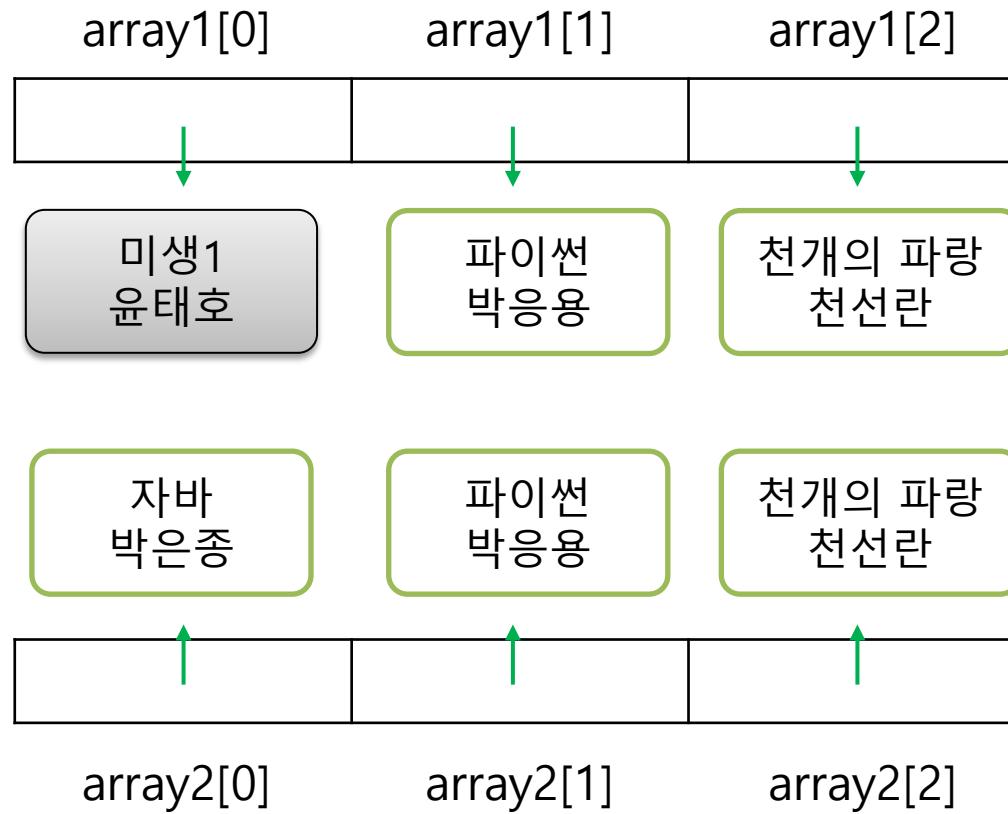
```
//배열 복사  
//System.arraycopy(array1, 0, array2, 0, 3);  
  
for(int i = 0; i < array1.length; i++) {  
    array2[i] = array1[i];  
}  
  
System.out.println("== array1 출력 ==");  
for(int i = 0; i < array1.length; i++) {  
    array1[i].showBookInfo();  
}  
  
System.out.println("== array2 출력 ==");  
for(Book book : array2) {  
    book.showBookInfo();  
}
```

```
== array1 출력 ==  
미생1, 윤태호  
점프 투 파이썬, 박응용  
천개의 파랑, 천선란  
== array2 출력 ==  
미생1, 윤태호  
점프 투 파이썬, 박응용  
천개의 파랑, 천선란
```



객체 배열 – 깊은 복사

■ 객체 배열의 깊은 복사(deep copy)



객체 배열 – 깊은 복사

■ 객체 배열의 깊은 복사

```
public class DeepCopy {  
  
    public static void main(String[] args) {  
        // 객체 배열의 깊은 복사  
        Book[] array1 = new Book[3];  
        Book[] array2 = new Book[3];  
  
        array1[0] = new Book("자바프로그래밍 입문", "박은종");  
        array1[1] = new Book("점프 투 파이썬", "박응용");  
        array1[2] = new Book("천개의 파랑", "천선란");  
  
        //기본 생성자로 array2 배열 인스턴스 생성  
        array2[0] = new Book();  
        array2[1] = new Book();  
        array2[2] = new Book();  
  
        //array1 배열 요소를 array2 배열 인스턴스에 복사  
        for(int i = 0; i < array1.length; i++) {  
            array2[i].setBookName(array1[i].getBookName());  
            array2[i].setAuthor(array1[i].getAuthor());  
        }  
    }  
}
```



객체 배열 – 깊은 복사

■ 객체 배열의 깊은 복사

```
//array1의 첫번째 요소값 수정  
array1[0].setBookName("미생1");  
array1[0].setAuthor("윤태호");  
  
System.out.println("== array1 출력 ==");  
for(int i = 0; i < array1.length; i++) {  
    array1[i].showBookInfo();  
}  
  
System.out.println("== array2 출력 ==");  
for(Book book : array2) {  
    book.showBookInfo();  
}
```

```
== array1 출력 ==  
미생1, 윤태호  
점프 투 파이썬, 박응용  
천개의 파랑, 천선란  
== array2 출력 ==  
자바프로그래밍 입문, 박은종  
점프 투 파이썬, 박응용  
천개의 파랑, 천선란
```



ArrayList 클래스

기존 배열의 단점과 ArrayList

- 배열의 길이가 미리 정해져 있어 길이가 늘어났을 때 새로 배열을 만들어야 함
- 배열의 요소가 변경되거나 삭제 되었을 때 요소를 비워둘 수 없으므로 배열 요소 위치를 변경해야 함.

ArrayList 클래스

- 배열 길이를 정하지 않으며 배열 길이와 상관 없이 객체를 추가.
- 배열 중간의 어떤 요소 값이 제거되면 그 다음 요소 값을 하나씩 앞으로 이동함

Class ArrayList<E>

java.lang.Object
 java.util.AbstractCollection<E>
 java.util.AbstractList<E>
 java.util.ArrayList<E>

Type Parameters:

E - the type of elements in this list

All Implemented Interfaces:

Serializable, Cloneable, Iterable<E>, Collection<E>, List<E>, I

Direct Known Subclasses:

AttributeList, RoleList, RoleUnresolvedList

```
public class ArrayList<E>
extends AbstractList<E>
implements List<E>, RandomAccess, Cloneable, Serializable
```



ArrayList 클래스

■ ArrayList 클래스 사용하기

```
ArrayList<E> 리스트이름 = new ArrayList<E>();
```

■ ArrayList 클래스의 주요 메서드

메서드	설 명
add()	요소 하나를 배열에 추가합니다.
size()	요소 전체 개수를 반환합니다.
get(index)	index위치의 요소 값을 반환합니다.
set(index, 요소)	index 위치의 요소 값을 변경합니다.
remove(index)	index위치의 요소 값을 제거합니다.
isEmpty()	배열이 비어있는지 확인합니다.



ArrayList 클래스(객체)

- String 클래스로
ArrayList로 구현

```
public class Cart {  
    public static void main(String[] args) {  
        //ArrayList의 객체 생성  
        ArrayList<String> cart = new ArrayList<>();  
  
        //요소 저장  
        cart.add("커피");  
        cart.add("계란");  
        cart.add("생수");  
  
        //요소의 개수  
        System.out.println("장바구니 품목: " + cart.size() + "개");  
  
        //전체 출력  
        for(int i=0; i<cart.size(); i++) {  
            String c = cart.get(i);  
            System.out.println(c);  
        }  
        System.out.println();  
  
        //자료 수정 - 커피를 라면으로 수정  
        cart.set(0, "라면");  
  
        //자료 삭제 - 생수  
        cart.remove(1);  
  
        //향상 for문 조회  
        for(String c : cart)  
            System.out.println(c);  
    }  
}
```



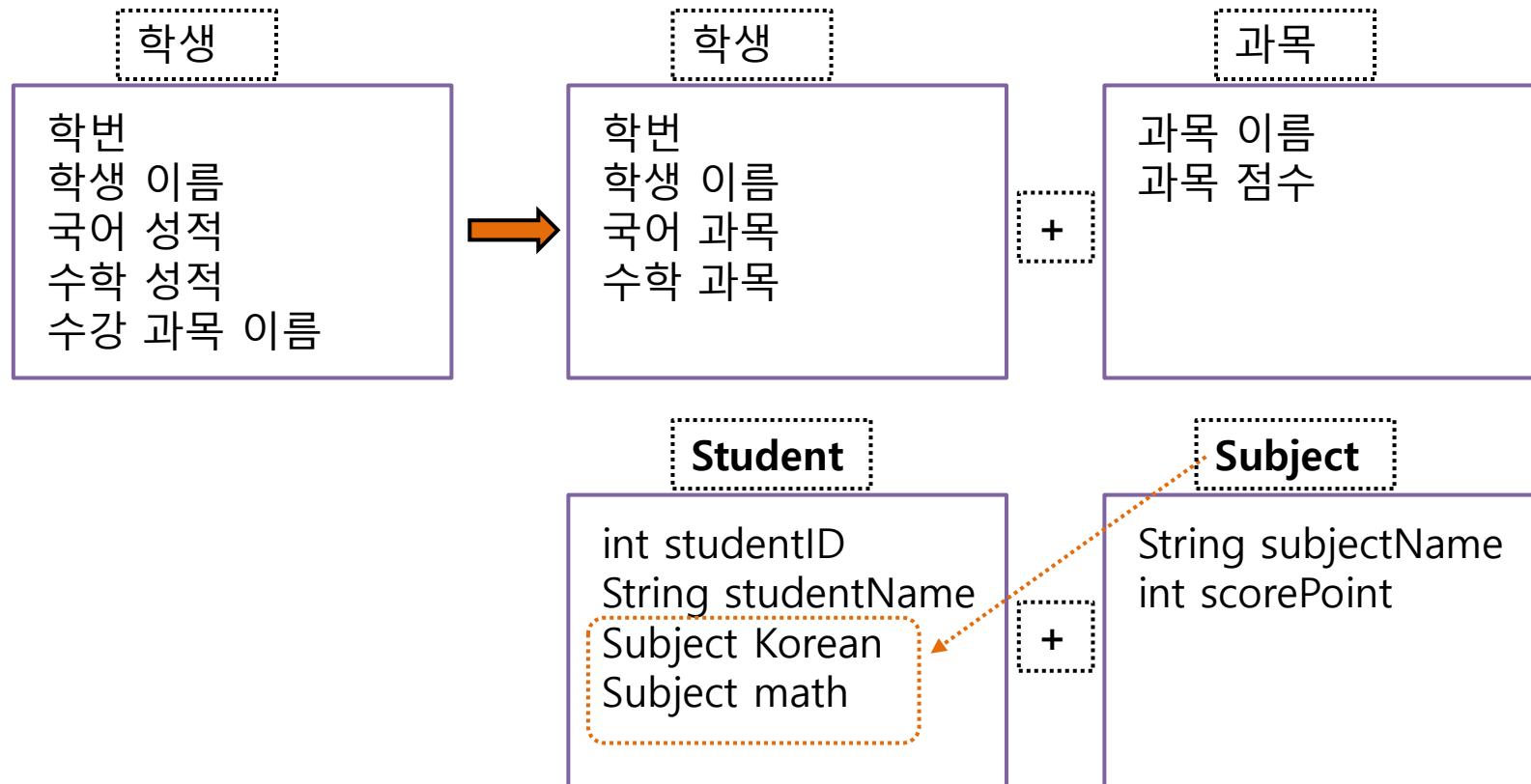
ArrayList 클래스

- Book 클래스로 ArrayList로 구현

```
public class BookList {  
  
    public static void main(String[] args) {  
        ArrayList<Book> bookList = new ArrayList<>();  
  
        Book book1 = new Book("자바프로그래밍 입문", "박은종");  
        Book book2 = new Book("점프 투 파이썬", "박응용");  
        Book book3 = new Book("천개의 파랑", "천선란");  
  
        //저장  
        bookList.add(book1);  
        bookList.add(book2);  
        bookList.add(book3);  
  
        //출력  
        for(int i = 0; i < bookList.size(); i++) {  
            Book book = bookList.get(i);  
            book.showBookInfo();  
        }  
    }  
}
```



클래스(자료형) 참조



문제점 : 이 클래스는 학생에 대한 클래스인데 과목 변수가 계속 늘어남

해결책 : 과목이름과 성적을 과목(Subject) 클래스로 분리함.



클래스(자료형) 참조

- 과목 클래스

```
public class Subject {  
    private String subjectName;  
    private int scorePoint;  
  
    public String getSubjectName() {  
        return subjectName;  
    }  
    public void setSubjectName(String subjectName) {  
        this.subjectName = subjectName;  
    }  
    public int getScorePoint() {  
        return scorePoint;  
    }  
    public void setScorePoint(int scorePoint) {  
        this.scorePoint = scorePoint;  
    }  
}
```



클래스(자료형) 참조

- 학생 클래스

```
public class Student {  
    int studentID;          //학번  
    String studentName;    //이름  
    Subject korean;  
    Subject math;  
  
    public Student(int studentID, String studentName) {  
        this.studentID = studentID;  
        this.studentName = studentName;  
        korean = new Subject();  
        math = new Subject();  
    }  
}
```



클래스(자료형) 참조

- 학생 클래스

```
public void koreanSubject(String name, int score) {  
    korean.setSubjectName(name);  
    korean.setScorePoint(score);  
}  
  
public void mathSubject(String name, int score) {  
    math.setSubjectName(name);  
    math.setScorePoint(score);  
}  
  
public void showInfo() {  
    System.out.println("학생 " + studentName + "의 국어 점수는 " +  
        korean.getScorePoint() + "점이고, 수학 점수는 " + math.getScorePoint() + "점입니다.");  
}
```



클래스(자료형) 참조

StudentTest.java

```
//학생의 국어, 수학과목 점수 생성
Student park = new Student(1001, "박마늘");
park.koreanSubject("국어", 90);
park.mathSubject("수학", 80);
park.showInfo();

Student lee = new Student(1002, "이양파");
lee.koreanSubject("국어", 70);
lee.mathSubject("수학", 60);
lee.showInfo();
```

학생 박마늘의 국어 점수는 90점이고, 수학 점수는 80점입니다.
학생 이양파의 국어 점수는 70점이고, 수학 점수는 60점입니다.



ArrayList 응용 프로그램

■ 학생 성적 출력 프로그램(ArrayList로 구현)

```
public class Student2 {  
    private int studentID;  
    private String studentName;  
    private ArrayList<Subject> subjectList;  
  
    public Student2(int studentID, String studentName) {  
        this.studentID = studentID;  
        this.studentName = studentName;  
        subjectList = new ArrayList<>();  
    }  
  
    public void addSubject(String name, int score) {  
        Subject subject = new Subject();  
        subject.setSubjectName(name);  
        subject.setScorePoint(score);  
        subjectList.add(subject);  
    }  
}
```



ArrayList 응용 프로그램

```
public void addSubject(String name, int score) {
    Subject subject = new Subject();
    subject.setSubjectName(name);
    subject.setScorePoint(score);
    subjectList.add(subject);
}

public void showStudentInfo() {
    int total = 0;
    double avg = 0.0;

    for(int i=0; i<subjectList.size(); i++) {
        Subject s = subjectList.get(i);
        total += s.getScorePoint();
        System.out.println("학생 " + studentName + "의 " + s.getSubjectName() +
                           "과목 성적은 " + s.getScorePoint() + "점 입니다.");
    }
    /*for(Subject s : subjectList) {
        total += s.getScorePoint();
        System.out.println("학생 " + studentName + "의 " + s.getSubjectName() +
                           "과목 성적은 " + s.getScorePoint() + "점 입니다.");
    }*/
    avg = (double)total / subjectList.size();
    System.out.printf("학생 %s님의 총점은 %d점이고, 평균은 %.2f점 입니다.\n" ,
                      studentName, total, avg);
}
```



ArrayList 응용 프로그램

```
public class ScoreList {  
  
    public static void main(String[] args) {  
        Student s1 = new Student(1001, "이양파");  
        Student s2 = new Student(1002, "박마늘");  
  
        s1.addSubject("국어", 95);  
        s1.addSubject("수학", 77);  
        s1.addSubject("Java", 82);  
        s1.showInfo();  
        System.out.println("=====");  
  
        s2.addSubject("국어", 84);  
        s2.addSubject("수학", 78);  
        s2.addSubject("Java", 95);  
        s2.showInfo();  
    }  
}
```

학생 이양파의 국어 점수는 95점 입니다.
학생 이양파의 수학 점수는 77점 입니다.
학생 이양파의 Java 점수는 82점 입니다.
학생 이양파의 총점은 254점, 평균은 84.67점입니다.
=====

학생 박마늘의 국어 점수는 84점 입니다.
학생 박마늘의 수학 점수는 78점 입니다.
학생 박마늘의 Java 점수는 95점 입니다.
학생 박마늘의 총점은 257점, 평균은 85.67점입니다.



ArrayList 응용 프로그램

■ 실습 예제

다음과 같이 Dog 클래스가 있다.

DogArrayList 클래스를 만들어 멤버변수로 ArrayList를 사용한다.

Dog 인스턴스 5개를 생성하여 ArrayList에 추가하고, ArrayList 정보를 출력하는 코드를 작성하세요.

```
백구1, 진돗개  
백구2, 시츄  
백구3, 푸들  
백구4, 불독  
백구5, 치와와
```



ArrayList 응용 프로그램

```
public class Dog {  
    private String name;  
    private String type;  
  
    public Dog() {}  
  
    public Dog(String name, String type) {  
        this.name = name;  
        this.type = type;  
    }  
  
    public String getName() {  
        return name;  
    }  
  
    public void setName(String name) {  
        this.name = name;  
    }  
  
    public String getType() {  
        return type;  
    }  
  
    public void setType(String type) {  
        this.type = type;  
    }  
  
    public String showDogInfo() {  
        return name + ", " + type;  
    }  
}
```



ArrayList 응용 프로그램

```
public class DogArrayList {  
  
    public static void main(String[] args) {  
        ArrayList<Dog> dogList = new ArrayList<>();  
  
        Dog dog1 = new Dog();  
        dog1.setName("백구1");  
        dog1.setType("진돗개");  
  
        Dog dog2 = new Dog();  
        dog2.setName("백구1");  
        dog2.setType("진돗개");  
  
        Dog dog3 = new Dog();  
        dog3.setName("백구1");  
        dog3.setType("진돗개");  
  
        Dog dog4 = new Dog();  
        dog4.setName("백구1");  
        dog4.setType("진돗개");  
  
        Dog dog5 = new Dog();  
        dog5.setName("백구1");  
        dog5.setType("진돗개");  
  
        //dogList에 저장  
        dogList.add(dog1);  
        dogList.add(dog2);  
        dogList.add(dog3);  
        dogList.add(dog4);  
        dogList.add(dog5);  
  
        //출력  
        for(int i = 0; i < dogList.size(); i++) {  
            Dog dog = dogList.get(i);  
            System.out.println(dog.showDogInfo());  
        }  
  
        //향상 for  
        for(Dog dog : dogList)  
            System.out.println(dog.showDogInfo());  
    }  
}
```



ArrayList 응용 프로그램

```
public class DogArrayList2 {
    public static void main(String[] args) {
        ArrayList<Dog> dogList = new ArrayList<>();

        Dog dog1 = new Dog("백구1", "진돗개");
        Dog dog2 = new Dog("백구2", "시츄");
        Dog dog3 = new Dog("백구3", "푸들");
        Dog dog4 = new Dog("백구4", "불독");
        Dog dog5 = new Dog("백구5", "치와와");

        dogList.add(dog1);
        dogList.add(dog2);
        dogList.add(dog3);
        dogList.add(dog4);
        dogList.add(dog5);

        /*dogList.add(new Dog("백구1", "진돗개"));
        dogList.add(new Dog("백구2", "시츄"));
        dogList.add(new Dog("백구3", "푸들"));
        dogList.add(new Dog("백구4", "불독"));
        dogList.add(new Dog("백구5", "치와와"));*/

        for(Dog dog : dogList)
            System.out.println(dog.showDogInfo());

        for(int i = 0; i < dogList.size(); i++) {
            Dog dog = dogList.get(i);
            System.out.println(dog.showDogInfo());
        }
    }
}
```

