

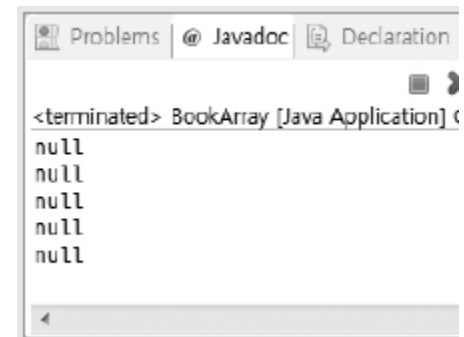
JAVA 프로그래밍

객체 배열 & ArrayList

객체 배열

- 참조 자료형을 선언하는 객체 배열
- 배열만 생성한 경우 요소는 null로 초기화 됨
- 각 요소를 new를 활용하여 생성하여 저장해야 함

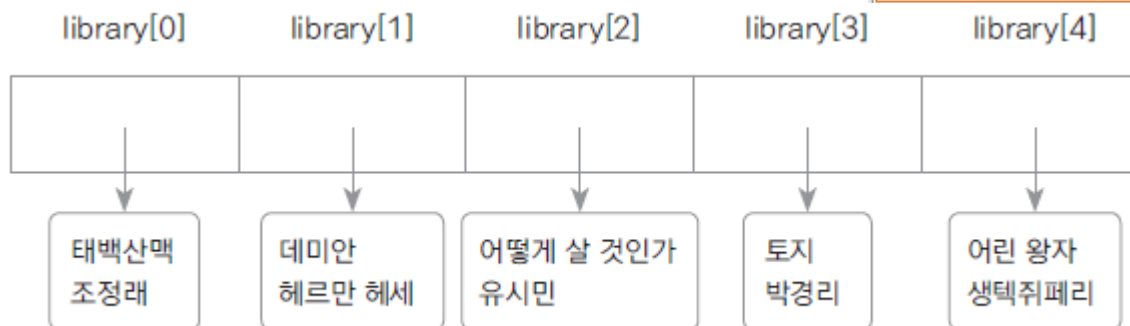
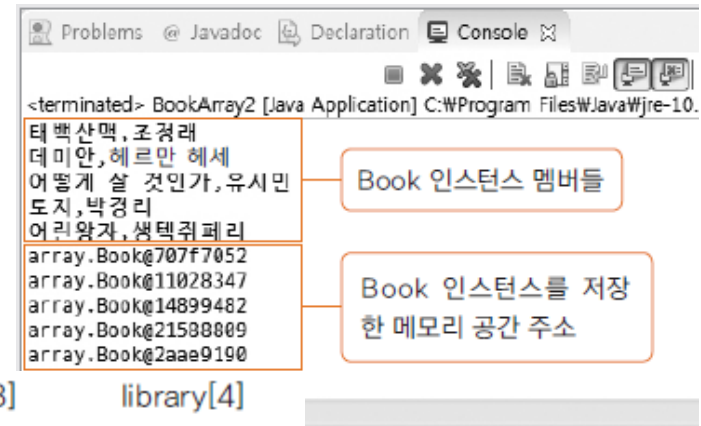
```
public class BookArray {  
  
    public static void main(String[] args) {  
        Book[] library = new Book[5];  
  
        for(int i=0; i<library.length; i++){  
            System.out.println(library[i]);  
        }  
    }  
}
```



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

객체 배열 만들기

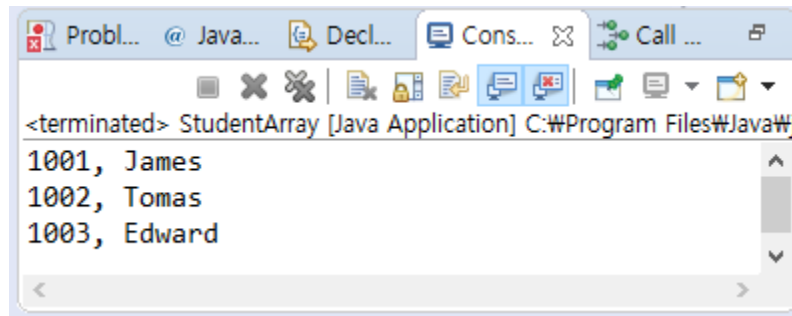
```
public class BookArray2 {  
  
    public static void main(String[] args) {  
        Book[] library = new Book[5];  
  
        library[0] = new Book("태백산맥", "조정래");  
        library[1] = new Book("데미안", "헤르만 헤세");  
        library[2] = new Book("어떻게 살 것인가", "유시민");  
        library[3] = new Book("토지", "박경리");  
        library[4] = new Book("어린왕자", "생텍쥐페리");  
  
        for(int i=0; i<library.length; i++){  
            library[i].showBookInfo();  
        }  
  
        for(int i=0; i<library.length; i++){  
            System.out.println(library[i]);  
        }  
    }  
}
```



Quiz 객체 배열

- Student 클래스를 만들고 멤버 변수(private)로 studentID, studentName을 선언
 - studentInfo() 메서드를 만들어 studentID, studentName 값을 출력
- StudentArray 클래스에서 Student 3명의 배열을 만드세요.
 - Student를 생성하여 저장한 후, 반복문을 사용해 Student 정보를 출력

[출력 결과]



```
<terminated> StudentArray [Java Application] C:\Program Files\Java\
1001, James
1002, Tomas
1003, Edward
```

배열 복사하기

- 기존 배열과 같은 배열을 만들거나 배열이 꽉 찬 경우 더 큰 배열을 만들고 기존 배열 자료를 복사할 수 있음
- `System.arraycopy(src, srcPos, dest, destPos, length);`

매개변수	설명
src	복사할 배열 이름
srcPos	복사할 배열의 첫 번째 위치
dest	복사해서 붙여 넣을 대상 배열 이름
destPos	복사해서 대상 배열에 붙여 넣기를 시작할 첫 번째 위치
length	src에서 dest로 자료를 복사할 요소 개수

객체 배열 복사하기

- ✓ **얕은 복사** : 배열 요소의 주소만 복사되므로 배열 요소가 변경되면
(shallow copy) 복사된 배열의 값도 변경 됨

```
public class ObjectCopy1 {  
    public static void main(String[ ] args) {  
        Book[ ] bookArray1 = new Book[3];  
        Book[ ] bookArray2 = new Book[3];  
  
        bookArray1[0] = new Book("태백산맥", "조정래");  
        bookArray1[1] = new Book("데미안", "헤르만 헤세");  
        bookArray1[2] = new Book("어떻게 살 것인가", "유시민");  
        System.arraycopy(bookArray1, 0, bookArray2, 0, 3);  
  
        for(int i = 0; i < bookArray2.length; i++) {  
            bookArray2[i].showBookInfo( );  
        }  
    }  
}
```

객체 배열 복사하기

```
System.out.println("=== bookArray1 ===");
```

```
bookArray1[0].setBookName("나뭇");  
bookArray1[0].setAuthor("박완서");
```

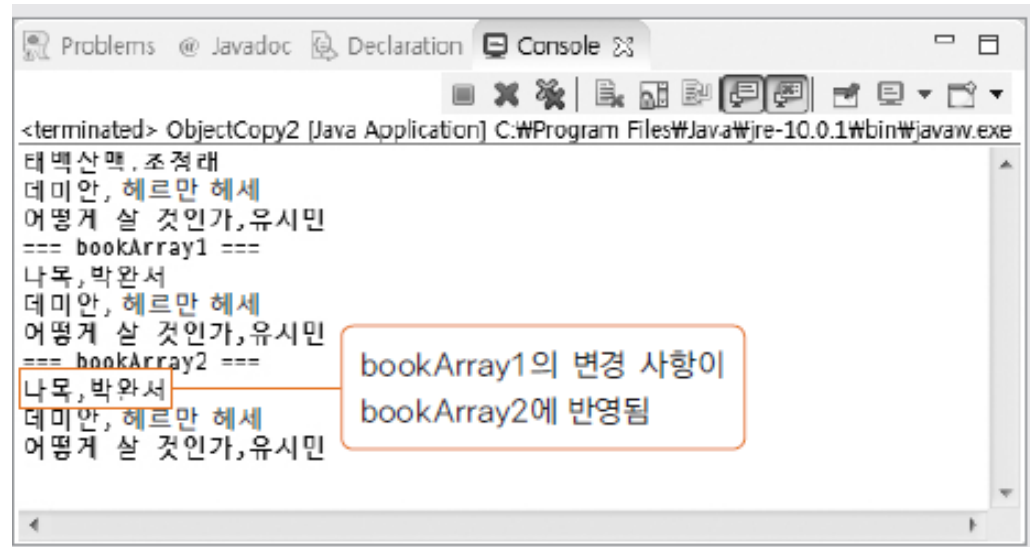
bookArray1 배열의 첫 번째 요소 값 변경

```
System.out.println("=== bookArray2 ===");
```

```
for(int i = 0; i < bookArray2.length; i++) {
```

```
    bookArray2[i].showBookInfo( );
```

bookArray2 배열 요소 값도 변경되어 출력

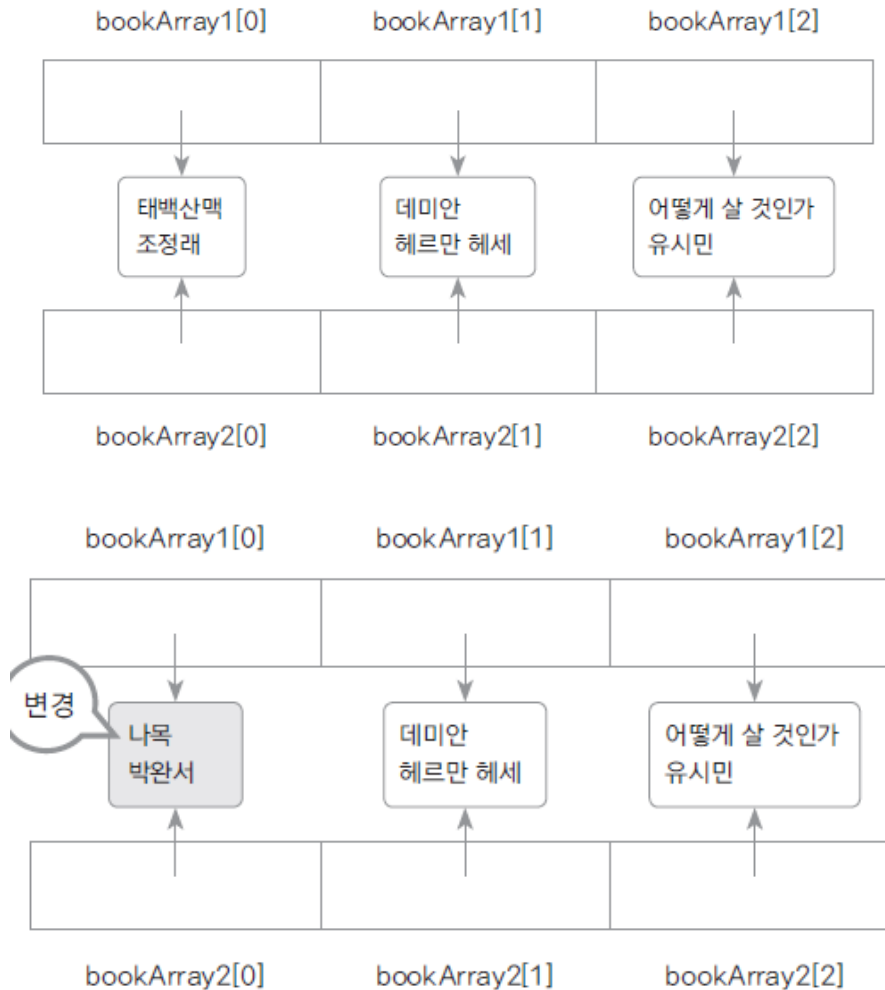


```
<terminated> ObjectCopy2 [Java Application] C:\Program Files\Java\jre-10.0.1\bin\javaw.exe  
태백산맥, 조정래  
데미안, 헤르만 헤세  
어떻게 살 것인가, 유시민  
=== bookArray1 ===  
나뭇, 박완서  
데미안, 헤르만 헤세  
어떻게 살 것인가, 유시민  
=== bookArray2 ===  
나뭇, 박완서  
데미안, 헤르만 헤세  
어떻게 살 것인가, 유시민
```

bookArray1의 변경 사항이
bookArray2에 반영됨

객체 배열 복사하기

- 배열의 요소 값을 변경하는 경우



객체 배열의 깊은 복사

```
bookArray2[0] = new Book( );  
bookArray2[1] = new Book( );  
bookArray2[2] = new Book( );
```

디폴트 생성자로 bookArray2
배열 인스턴스 생성

```
for(int i = 0; i < bookArray1.length; i++) {  
    bookArray2[i].setBookName(bookArray1[i].getBookName( ));  
    bookArray2[i].setAuthor(bookArray1[i].getAuthor( ));  
}
```

bookArray1 배열
요소를 새로 생성한
bookArray2 배열
인스턴스에 복사

```
bookArray1[0].setBookName("나목");  
bookArray1[0].setAuthor("박완서");
```

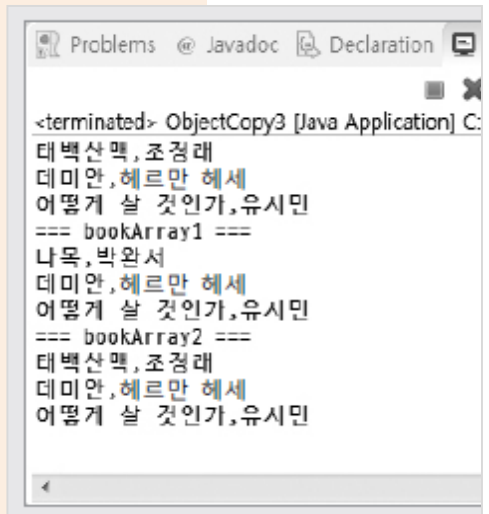
bookArray1 첫 번째 배열
요소 값 수정

```
System.out.println("=== bookArray1 ===");  
for(int i = 0; i < bookArray1.length; i++) {  
    bookArray1[i].showBookInfo( ); //bookArray1 배열 요소 값 출력  
}
```

```
System.out.println("=== bookArray2 ===");
```

```
for(int i = 0; i < bookArray2.length; i++) {  
    bookArray2[i].showBookInfo( ); //bookArray2 배열 요소 값 출력  
}
```

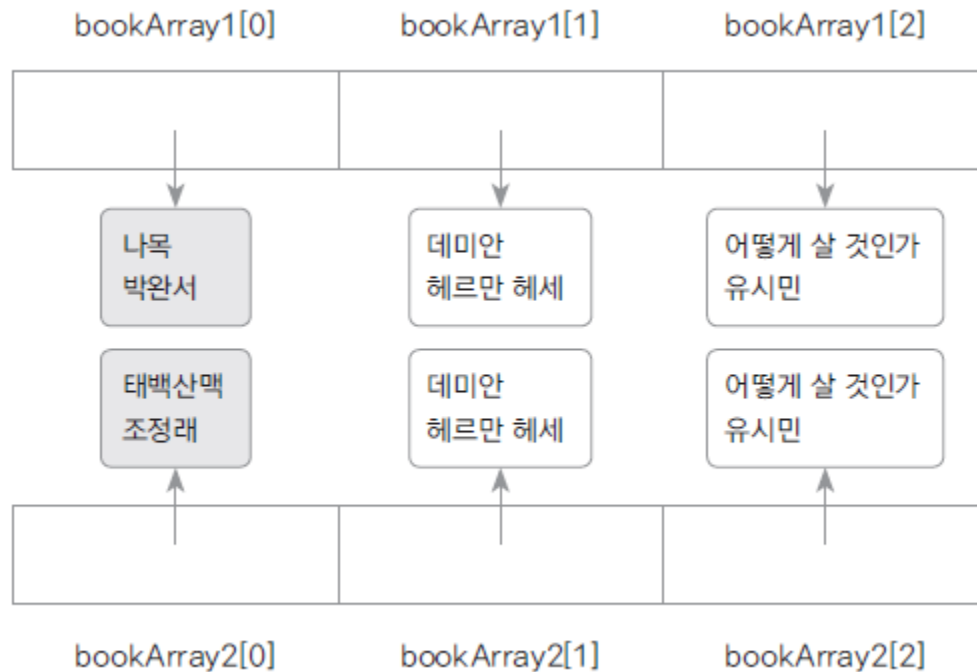
bookArray1 배열
요소 값과 다른 내
용이 출력됨



```
<terminated> ObjectCopy3 [Java Application] C:  
태백산맥, 조정래  
데미안, 헤르만 헤세  
어떻게 살 것인가, 유시민  
=== bookArray1 ===  
나목, 박완서  
데미안, 헤르만 헤세  
어떻게 살 것인가, 유시민  
=== bookArray2 ===  
태백산맥, 조정래  
데미안, 헤르만 헤세  
어떻게 살 것인가, 유시민
```

객체 배열의 깊은 복사

- ✓ 깊은 복사 : 서로 다른 인스턴스의 메모리를 요소로 가지게 됨
(deep copy)



ArrayList 클래스

- 기존 배열은 길이를 정하여 선언하므로 사용 중 부족한 경우, 다른 배열로 복사하는 코드를 직접 구현해야 함
- 중간의 요소가 삭제되거나 삽입되는 경우도 나머지 요소에 대한 조정하는 코드를 구현해야 함
- ArrayList 클래스는 자바에서 제공되는 객체 배열이 구현된 클래스
- 여러 메서드와 속성 등을 사용해 객체 배열을 편리하게 관리할 수 있음
- 가장 많이 사용하는 객체 배열 클래스

ArrayList 클래스 주요 메서드

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

- 요소를 추가하거나 제거할 때, 각 내부에서 코드가 모두 구현되어 있으므로 배열을 직접 선언하여 사용하는 것보다 편리함

ArrayList 클래스 사용하기

- `ArrayList<E> 배열 이름 = new ArrayList<E>();`
- 사용할 객체를 `E` 위치에 넣고, ArrayList 클래스를 활용하여 추가하거나 참조할 수 있음

```
ArrayList<Book> library = new ArrayList<Book>( );
```

ArrayList 선언

```
library.add(new Book("태백산맥", "조정래"));  
library.add(new Book("데미안", "헤르만 헤세"));  
library.add(new Book("어떻게 살 것인가", "유시민"));  
library.add(new Book("토지", "박경리"));  
library.add(new Book("어린왕자", "생텍쥐페리"));
```

add() 메서드로 요소 값 추가

```
for(int i = 0; i < library.size( ); i++) {  
    Book book = library.get(i);  
    book.showBookInfo( );  
}
```

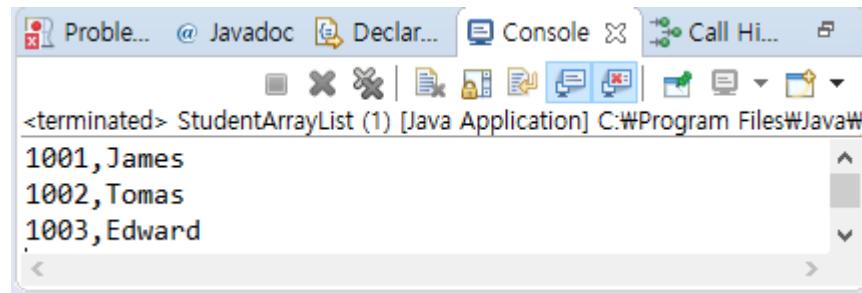
배열에 추가된 요소 개수만큼 출력

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

Quiz ArrayList

- 객체 배열 Quiz에서 만든 Student 클래스를 이용하여 StudentArrayList 클래스를 만들고 ArrayList<Student> 자료형의 ArrayList를 선언
- ArrayList에 학생 3명을 추가하고 그 정보를 출력 결과와 같이 구현

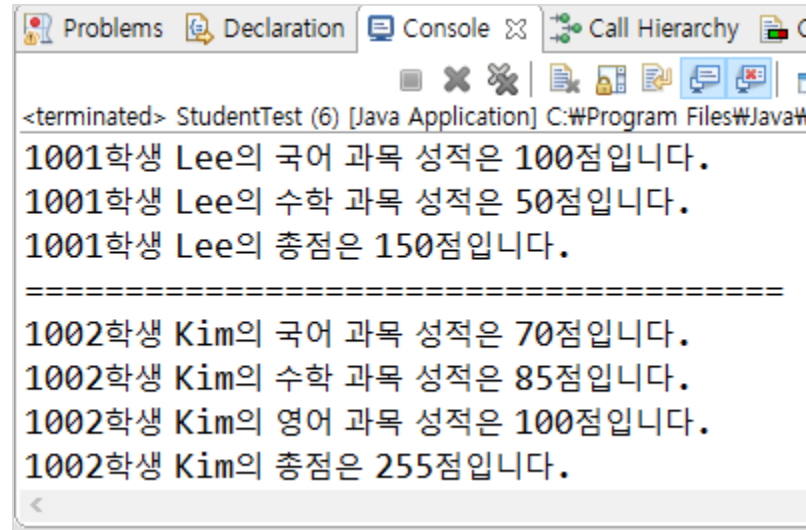
[출력 결과]



```
<terminated> StudentArrayList (1) [Java Application] C:#Program Files#Java#  
1001, James  
1002, Tomas  
1003, Edward
```

ArrayList 응용 프로그램

- 1001학번 Lee 와 1002학번 Kim, 두 학생이 있습니다.
Lee 학생은 2과목을 수강했고 국어 점수가 100점, 수학 점수가 50점입니다.
Kim 학생은 3과목을 수강하여 국어 점수가 70점, 수학 점수가 85점, 영어 점수가 100점입니다.
- Student 클래스와 Subject 클래스를 생성한 후, 두 학생의 과목 성적과 총점을 출력해 보면?



```
<terminated> StudentTest (6) [Java Application] C:\Program Files\Java\W
1001학생 Lee의 국어 과목 성적은 100점입니다.
1001학생 Lee의 수학 과목 성적은 50점입니다.
1001학생 Lee의 총점은 150점입니다.
=====
1002학생 Kim의 국어 과목 성적은 70점입니다.
1002학생 Kim의 수학 과목 성적은 85점입니다.
1002학생 Kim의 영어 과목 성적은 100점입니다.
1002학생 Kim의 총점은 255점입니다.
```

Student 클래스 구현하기

```
package arraylist;  
import java.util.ArrayList;
```

```
public class Student {
```

```
    int studentID;  
    String studentName;  
    ArrayList<Subject> subjectList;
```

```
    public Student(int studentID, String studentName) {  
        this.studentID = studentID;  
        this.studentName = studentName;  
        subjectList = new ArrayList<Subject>();  
    }
```

```
    public void addSubject(String name, int score) {  
        Subject subject = new Subject(); // Subject 생성  
  
        subject.setName(name);           // 과목명 추가  
        subject.setScorePoint(score);    // 점수 추가  
        subjectList.add(subject);        // 배열에 저장하기  
    }
```

학생이 수강하는 과목을
subjectList 배열에 추가
하는 메서드

Student 클래스 구현하기

```
public void showStudentInfo() {  
    int total = 0;  
  
    for(Subject s : subjectList) {  
        total += s.getScorePoint();  
        System.out.println(studentID + "학생 " + studentName + "의 " + s.getName()  
            + " 과목 성적은 " + s.getScorePoint() + "점입니다.");  
    }  
  
    System.out.println(studentID + "학생 " + studentName + "의 총점은 " + total + "점입니다.");  
}
```

Subject 클래스 구현하기

```
package arraylist;

public class Subject {
    private String name;
    private int scorePoint;

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public int getScorePoint() {
        return scorePoint;
    }

    public void setScorePoint(int scorePoint) {
        this.scorePoint = scorePoint;
    }
}
```

테스트 클래스 구현 : 학생 성적 출력하기

```
package arraylist;

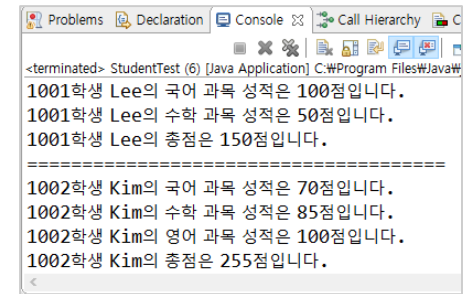
public class StudentTest {
    public static void main(String[] args) {
        Student studentLee = new Student(1001, "Lee");

        studentLee.addSubject("국어", 100);
        studentLee.addSubject("수학", 50);

        Student studentKim = new Student(1002, "Kim");

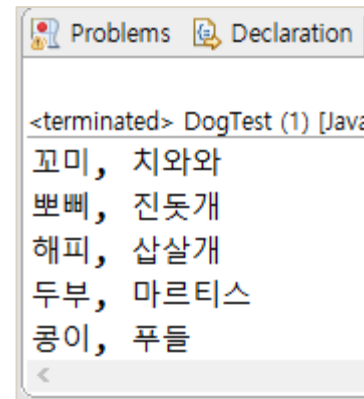
        studentKim.addSubject("국어", 70);
        studentKim.addSubject("수학", 85);
        studentKim.addSubject("영어", 100);

        studentLee.showStudentInfo();
        System.out.println("=====");
        studentKim.showStudentInfo();
    }
}
```



Quiz 객체 배열

- Dog 클래스의 멤버 변수로 강이지 이름, 종류가 있습니다. (private 접근제어자)
- 각 멤버 변수는 get(), set() 메서드가 있고, 강아지의 이름과 종류를 반환하는 showDogInfo() 메서드를 구현하여, 아래의 출력 결과와 같이 DogTest 클래스를 완성해 보세요.
 - 배열 길이가 5개인 배열을 만든 후 Dog 인스턴스를 5개 생성해 배열에 추가
 - for문과 Enhanced for문에서 Dog 클래스의 showDogInfo() 메서드를 사용하여 배열에 추가한 Dog 정보를 모두 출력



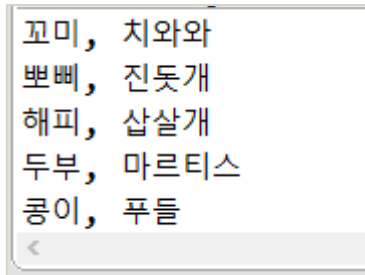
The screenshot shows a console window titled "Problems Declaration" with a tab for "<terminated> DogTest (1) [Java]". The output text is as follows:

```
<terminated> DogTest (1) [Java]
꼬미, 치와와
뽀빠, 진돗개
해피, 샴살개
두부, 마르티스
콩이, 푸들
```

Quiz ArrayList

- 앞서 작성한 소스 코드를 DogArrayListTest 클래스로 만들어 멤버 변수로 ArrayList를 사용
- Dog 인스턴스를 5개 생성하여 ArrayList에 추가하고 ArrayList의 정보를 출력하는 코드를 완성해 보세요.

[출력 결과]



```
꼬미, 치와와  
뽀삐, 진돗개  
해피, 샴살개  
두부, 마르티스  
콩이, 푸들  
<
```