

□ 개념 확인

- (1) **[생성형 AI 활용]** 제시된 코드에서 배열 데이터 타입을 다양하게 적용할 수 있는 방법을 제시하고 테스트 하세요

```
class Cube {  
    private int[] dim;  
    public Cube(int[] dim){  
        this.dim=dim;  
    }  
    public void set(int index, int value ) {  
        this.dim[index] = value;  
    }  
    public int get(int index) {  
        return dim[index];  
    }  
  
    public String toString() {  
        return Arrays.toString(data);  
    }  
}
```

생성형 AI	Perplexity
질문	<p>제시된 코드에서 배열 데이터 타입을 다양하게 적용할 수 있는 방법을 제시하고 테스트 하세요</p> <pre>class Cube { private int[] dim; public Cube(int[] dim){ this.dim=dim; } public void set(int index, int value) { this.dim[index] = value; } public int get(int index) { return dim[index]; } public String toString() { return Arrays.toString(data); } }</pre>

1. 문제 해석

- 현재 Cube 클래스는 int[] 타입만 지원합니다.
- 배열 데이터 타입(예: double[], String[] 등)도 사용할 수 있게 하라는 요구입니다.

2. 해결 방법: 제네릭(Generics) 사용

Java의 제네릭(Generic)을 활용하면 다양한 타입의 배열을 지원할 수 있습니다.

수정된 코드 예시

```
java
import java.util.Arrays;

class Cube<T> {
    private T[] dim;

    public Cube(T[] dim) {
        this.dim = dim;
    }

    public void set(int index, T value) {
        this.dim[index] = value;
    }

    public T get(int index) {
        return dim[index];
    }

    public String toString() {
        return Arrays.toString(dim);
    }
}
```

3. 테스트 코드 예시

(1) int 배열 테스트

```
java
Cube<Integer> intCube = new Cube<>(new Integer[]{1, 2, 3});
intCube.set(1, 10);
System.out.println(intCube); // [1, 10, 3]
```

(2) double 배열 테스트

	<pre>java Cube<Double> doubleCube = new Cube<>(new Double[]{1.1, 2.2, 3.3}); doubleCube.set(0, 9.9); System.out.println(doubleCube); // [9.9, 2.2, 3.3]</pre> <p>(3) String 배열 테스트</p> <pre>java Cube<String> stringCube = new Cube<>(new String[]{"a", "b", "c"}); stringCube.set(2, "z"); System.out.println(stringCube); // [a, b, z]</pre> <hr/> <p>4. 요약</p> <ul style="list-style-type: none">• 제네릭 사용으로 Cube 클래스가 다양한 타입의 배열(int[], double[], String[] 등)을 지원할 수 있습니다.• 각 타입별로 객체를 생성해 set/get 메서드 및 toString()을 테스트할 수 있습니다. <hr/> <p>이렇게 하면 다양한 배열 타입을 Cube 클래스에서 손쉽게 사용할 수 있습니다.</p>
--	---

(2) **[생성형 AI 활용]** 다음과 같은 코드가 컴파일 되지 않는다면 원인은 무엇일까요? 원인과 해결 방법을 함께 제시하세요

```
class MyAlgorithm {
    public static <T> T max(T x, T y) {
        return x > y ? x : y;
    }
}

public class Test {
    public static void main(String[] args) {
        System.out.println(MyAlgorithm.max(34, 20));
    }
}
```

생성형 AI	Perplexity
질문	<p>다음과 같은 코드가 컴파일 되지 않는다면 원인은 무엇일까요? 원인과 해결 방법을 함께 제시하세요</p> <pre>class MyAlgorithm { public static <T> T max(T x, T y) { return x > y ? x : y; } } public class Test { public static void main(String[] args) {</pre>

	<pre>System.out.println(MyAlgorithm.max(34, 20)); } }</pre>
답변	<p>답변: 컴파일이 되지 않는 원인과 해결 방법</p> <p>1. 원인</p> <p>아래 코드가 컴파일되지 않는 이유는 제네릭 타입 T에 대해 '>' 연산자를 사용할 수 없기 때문입니다.</p> <pre>java public static <T> T max(T x, T y) { return x > y ? x : y; // 오류 발생! }</pre> <ul style="list-style-type: none">자바의 제네릭 타입 T는 어떤 타입이든 올 수 있기 때문에, T가 정수(Integer)나 실수(Double)처럼 '>' 연산자를 지원한다는 보장이 없습니다.자바에서 '>', '<' 같은 비교 연산자는 기본형(primitive type)에만 사용 가능합니다. 참조형(객체)에는 사용할 수 없습니다 25. <p>2. 해결 방법</p> <p>T가 반드시 비교 가능한 타입임을 명시해야 합니다. 자바에서는 비교 가능한 객체는 Comparable 인터페이스를 구현합니다. 따라서, **타입 파라미터에 상한 제한(bound)**을 걸어야 합니다.</p> <p>수정 코드</p> <pre>java class MyAlgorithm { public static <T extends Comparable<T>> T max(T x, T y) { return x.compareTo(y) > 0 ? x : y; } } public class Test { public static void main(String[] args) { System.out.println(MyAlgorithm.max(34, 20)); } }</pre> <ul style="list-style-type: none"><T extends Comparable<T>>로 선언하면, T는 반드시 compareTo() 메서드를 가진 타입이어야 하므로 안전하게 비교할 수 있습니다 25.compareTo()는 두 객체를 비교해 크면 양수, 같으면 0, 작으면 음수를 반환합니다. <p>3. 요약</p>

	<ul style="list-style-type: none">원인: 제네릭 타입 T에 대해 '>' 연산자를 쓸 수 없어서 컴파일 오류 발생 25. <p>해결: T에 Comparable<T> 상한 제한을 걸고, compareTo()로 비교하도록 수정 25.</p>
--	--

(3) **[생성형 AI 활용]** List, Set, Map 인터페이스의 차이점을 제시하세요

생성형 AI	Perplexity																												
질문	List, Set, Map 인터페이스의 차이점을 제시하세요																												
답변	<div><h2>List, Set, Map 인터페이스의 차이점</h2><table><tr><th>특성</th><th>List</th><th>Set</th><th>Map</th></tr><tr><td>순서 유지</td><td>O</td><td>X (단, LinkedHashSet은 O)</td><td>X (단, LinkedHashMap은 O)</td></tr><tr><td>중복 허용</td><td>O</td><td>X</td><td>키: X / 값: O</td></tr><tr><td>접근 방식</td><td>인덱스(index)로 접근</td><td>요소 존재 여부로 접근</td><td>키(key)로 값(value)에 접근</td></tr><tr><td>구조</td><td>값(value)들의 집합</td><td>유일한 값(value)들의 집합</td><td>키(key)와 값(value) 쌍의 집합</td></tr><tr><td>주요 구현체</td><td>ArrayList, LinkedList 등</td><td>HashSet, LinkedHashSet</td><td>TreeSet, HashMap, TreeMap, LinkedHashMap</td></tr><tr><td>사용 예시</td><td>순서가 중요한 데이터</td><td>중복 없는 유일한 값 저장</td><td>키-값 쌍으로 데이터 저장</td></tr></table></div> <div><h2>정리</h2><ul style="list-style-type: none">• List<ul style="list-style-type: none">○ 데이터의 <i>순서</i>를 유지하며, <i>중복</i>을 허용합니다.○ 인덱스를 통해 원하는 위치에 접근하거나 값을 추가/삭제할 수 있습니다.○ 예: ArrayList, LinkedList• Set<ul style="list-style-type: none">○ 데이터의 <i>순서</i>를 유지하지 않으며(예외: LinkedHashSet), <i>중복</i>을 허용하지 않습니다.○ 같은 값이 여러 번 들어갈 수 없습니다.○ 예: HashSet, TreeSet• Map<ul style="list-style-type: none">○ <i>*키(key)*</i>와 <i>값(value)</i> 쌍으로 저장합니다.○ <i>키</i>는 중복을 허용하지 않지만, <i>값</i>은 중복될 수 있습니다.○ 인덱스 대신 <i>키</i>를 통해 값을 조회합니다.○ 예: HashMap, TreeMap</div>	특성	List	Set	Map	순서 유지	O	X (단, LinkedHashSet은 O)	X (단, LinkedHashMap은 O)	중복 허용	O	X	키: X / 값: O	접근 방식	인덱스(index)로 접근	요소 존재 여부로 접근	키(key)로 값(value)에 접근	구조	값(value)들의 집합	유일한 값(value)들의 집합	키(key)와 값(value) 쌍의 집합	주요 구현체	ArrayList, LinkedList 등	HashSet, LinkedHashSet	TreeSet, HashMap, TreeMap, LinkedHashMap	사용 예시	순서가 중요한 데이터	중복 없는 유일한 값 저장	키-값 쌍으로 데이터 저장
특성	List	Set	Map																										
순서 유지	O	X (단, LinkedHashSet은 O)	X (단, LinkedHashMap은 O)																										
중복 허용	O	X	키: X / 값: O																										
접근 방식	인덱스(index)로 접근	요소 존재 여부로 접근	키(key)로 값(value)에 접근																										
구조	값(value)들의 집합	유일한 값(value)들의 집합	키(key)와 값(value) 쌍의 집합																										
주요 구현체	ArrayList, LinkedList 등	HashSet, LinkedHashSet	TreeSet, HashMap, TreeMap, LinkedHashMap																										
사용 예시	순서가 중요한 데이터	중복 없는 유일한 값 저장	키-값 쌍으로 데이터 저장																										

요약:

- List: 순서 O, 중복 O
- Set: 순서 X, 중복 X

Map: 키-값 쌍, 키 중복 X, 값 중복 O, 순서 X (일부 구현체 예외)[2346](#)

(4) 아래의 프로그램 코드 ~~ 부분에 배열 원소 2개를 서로 교환하는 제네릭 메소드를 작성하고 테스트하세요.

```
class MyAlgorithm {  
    public static <T> void swap(T[] a, int i, int j) { //a[i]와 a[j]를 서로 교환
```

~~

```
}  
}
```

```
=== 바꾸기 수행 전 ===  
[23, 56, 75, 32, 78, 44]  
[자, 바, 프, 로, 그, 램]  
=== 바꾸기 수행 후 ===  
[23, 56, 78, 32, 75, 44]  
[자, 램, 프, 로, 그, 바]
```

```
public class Test {  
    public static void main(String[] args) {  
        Integer[] dim= {23, 56,75,32,78,44};  
        Character[] cdim = {'자', '바', '프','로','그','램'};  
  
        //이곳에 제시된 결과를 참고하여 나머지 부분을 채워 넣으세요  
  
    }  
}
```

[프로그램 소스]

```
package practice4;  
  
public class MyAlgorithm {  
    public static <T> void swap(T[] a, int i, int j) { // a[i]와 a[j]를 서로 교환  
        T temp = a[i];  
        a[i] = a[j];  
        a[j] = temp;  
    }  
}
```

```
package practice4;  
  
import java.util.Arrays;  
  
public class Test {  
    public static void main(String[] args) {  
        Integer[] dim= {23, 56,75,32,78,44};  
        Character[] cdim = {'자', '바', '프','로','그','램'};  
  
        System.out.println("=== 바꾸기 수행 전 ===");  
        System.out.println(Arrays.toString(dim));  
        System.out.println(Arrays.toString(cdim));  
    }  
}
```

```

MyAlgorithm.swap(dim, 2, 4);
MyAlgorithm.swap(cdim, 1, 5);

System.out.println("=== 바꾸기 수행 후 ===");
System.out.println(Arrays.toString(dim));
System.out.println(Arrays.toString(cdim));
}
}

```

[실행 결과]

```

/Users/hataeyeong/Library/Ja
=== 바꾸기 수행 전 ===
[23, 56, 75, 32, 78, 44]
[자, 바, 프, 로, 그, 램]
=== 바꾸기 수행 후 ===
[23, 56, 78, 32, 75, 44]
[자, 램, 프, 로, 그, 바]

종료 코드 0(으)로 완료된 프로세스

```

- (5) 타입 매개 변수 T를 가지는 클래스 MyMath를 작성합니다. MyMath에는 평균을 구하는 getAverage() 메소드를 추가합니다. 단, Byte, Short, Integer, Double, Float, Long 타입의 데이터에 대하여 평균을 구할 수 있도록 합니다.

```

public class Test{
    public static void main(String[] args) {
        Integer[] list = { 32,5,36,8,41,74,59,9,11};
        Byte[] blist = {4,12,7,84,23,4};

        MyMath<Integer> m = new MyMath<>();
        MyMath<Byte> bm = new MyMath<>();
        System.out.printf("average(Integer) : %.2f\n",m.getAverage(list));
        System.out.printf("average(Byte) :%.2f\n", bm.getAverage(blist));
    }
}

```

[프로그램 소스]

```

package practice5;

// Byte, Short, Integer, Double, Float, Long 은 Number 의 하위 클래스임으로, Number 의 하위
// 클래스로 제한
class MyMath<T extends Number> {
    public double getAverage(T[] numbers) {

```

```

// 데이터가 비어있는지 확인
if (numbers == null || numbers.length == 0) {
    return 0.0;
}
// 총합 초기화
double sum = 0.0;

// 평균 계산
for (T num : numbers) {
    sum += num.doubleValue(); // 실수형 타입으로 변환하여 총합에 더한다.
}
return sum / numbers.length;
}
}

```

```

package practice5;

public class Test{
    public static void main(String[] args) {
        Integer[] list = { 32,5,36,8,41,74,59,9,11};
        Byte[] blist = {4,12,7,84,23,4};

        MyMath<Integer> m = new MyMath<>();
        MyMath<Byte> bm = new MyMath<>();
        System.out.printf("average(Integer) : %.2f\n",m.getAverage(list));
        System.out.printf("average(Byte) :%.2f\n", bm.getAverage(blist));
    }
}

```

[실행 결과]

```

/Users/hataeyeong/Library/Java
average(Integer) : 30.56
average(Byte) : 22.33
종료 코드 0(으)로 완료된 프로세스

```

- (6) 다음과 같이 리스트가 생성되었을 때 제시된 문제를 해결하는 프로그램을 작성하세요
- ```

String[] s = { "사과", "배", "바나나" };
ArrayList<String> list = new ArrayList(Arrays.asList(s)); //배열을 리스트로 변환

```



```

----- 리스트 출력 -----
사과
배
자몽
바나나
포도

인덱스 3 : 바나나
배가 저장된 위치 : 1
----- 리스트 출력 -----
사과
배
메론
포도
배

```

- ① "포도"를 리스트 마지막에 삽입
- ② "자몽"을 인덱스 2에 삽입
- ③ 리스트에 저장된 데이터 출력
- ④ 인덱스 3에 저장된 데이터 출력
- ⑤ 인덱스 2에 저장된 데이터를 "메론"으로 변경
- ⑥ "배"를 리스트 마지막에 삽입
- ⑦ "배"가 저장된 위치를 출력
- ⑧ 리스트에서 "바나나" 데이터 삭제 - remove(Object obj), remove(int index)
- ⑨ 리스트에 저장된 데이터 출력

#### [프로그램 소스]

```

package practice6;

import java.util.ArrayList;
import java.util.Arrays;

public class Main {
 public static void main(String[] args) {
 String[] s = { "사과", "배", "바나나" };
 ArrayList<String> list = new ArrayList(Arrays.asList(s)); //배열을 리스트로 변환

 // ① "포도"를 리스트 마지막에 삽입
 list.add(list.size(), "포도");

 // ② "자몽"을 인덱스 2에 삽입
 list.add(2, "자몽");

 // ③ 리스트에 저장된 데이터 출력
 System.out.println("----- 리스트 출력 -----");
 for (String str: list) {
 System.out.println(str);
 }

 // ④ 인덱스 3에 저장된 데이터 출력
 String a = list.get(3);
 System.out.println("-----");
 System.out.println("인덱스 3 : " + a);

 // ⑦ "배"가 저장된 위치를 출력
 int idx = list.indexOf("배");
 System.out.println("배가 저장된 위치 : " + idx);

 // ⑤ 인덱스 2에 저장된 데이터를 "메론"으로 변경
 list.remove(2);
 list.add(2, "메론");

 // ⑥ "배"를 리스트 마지막에 삽입
 list.add(list.size(), "배");

 // ⑧ 리스트에서 "바나나" 데이터 삭제 - remove(Object obj), remove(int index)
 list.remove("바나나");

 // ⑨ 리스트에 저장된 데이터 출력
 System.out.println("----- 리스트 출력 -----");
 for (String str: list) {
 System.out.println(str);
 }
 }
}

```

```
}
}
```

[실행 결과]



```
실행 Main x
----- 리스트 출력 -----
사과
배
자몽
바나나
포도

인덱스 3 : 바나나
배가 저장된 위치 : 1
----- 리스트 출력 -----
사과
배
메론
포도
배

종료 코드 0(으)로 완료된 프로세스
```

(7) 두 자리수 로또 번호 6개를 생성하는 프로그램을 작성하세요. 단, 중복을 허용하지 않으며 HashSet을 사용합니다

두 자리수 로또 번호 6개를 출력합니다.  
[53, 22, 40, 25, 30, 63]

[프로그램 소스]

```
package practice7;

import java.util.HashSet;
import java.util.Random;
import java.util.Set;

public class Main {
 public static void main(String[] args) {
 Set<Integer> set = new HashSet<Integer>();
 Random random = new Random();

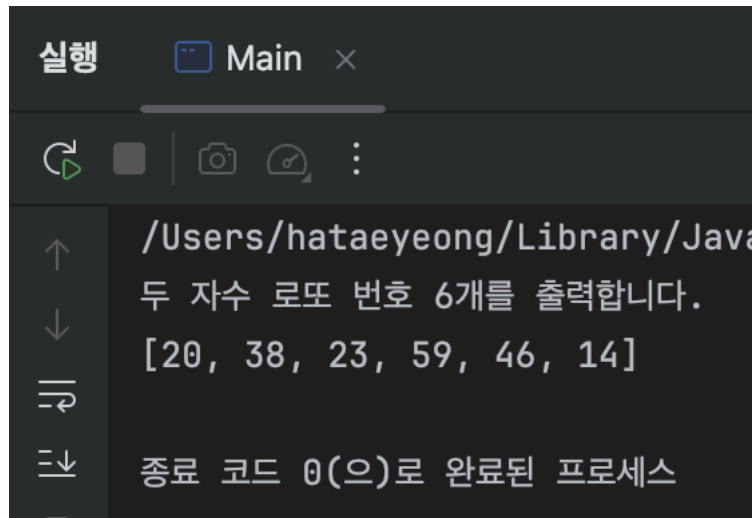
 while(set.size() < 6) {
 int num = random.nextInt(90) + 10; // 10 ~ 99 의 정수
 set.add(num);
 }
 System.out.println("두 자수 로또 번호 6 개를 출력합니다.");
 }
}
```

```

 System.out.println(set);
 }
}

```

#### [실행 결과]



- (8) Map을 사용하여 마라톤 참가자를 처리하는 프로그램을 작성하세요. 참가 번호는 선착순으로 사용하며 이름은 입력을 받습니다. key는 참가번호, value는 이름을 사용합니다. 참가 번호를 입력하면 이름을 출력합니다.

```

마라톤 참가자를 등록합니다.
참가자 이름을 입력하세요 (종료하려면 'exit' 입력):
참가자 이름: hallym
등록 완료! 현재 참가자 수: 1
참가자 이름: content
등록 완료! 현재 참가자 수: 2
참가자 이름: bigdata
등록 완료! 현재 참가자 수: 3
참가자 이름: exit

참가자 목록:
참가번호 1번: hallym
참가번호 2번: content
참가번호 3번: bigdata

```

#### [프로그램 소스]

```

package practice8;

class Member {
 private int number;
 private String name;

 public Member(int number, String name) {
 this.number = number;
 this.name = name;
 }

 @Override
 public String toString() {
 return "참가번호 " + number + "번 : " + name;
 }
}

```

```

package practice8;

import java.util.HashMap;
import java.util.Map;
import java.util.Scanner;

```

```

public class Main {
 public static void main(String[] args){
 Scanner sc = new Scanner(System.in);
 Map<Integer, Member> map = new HashMap<>();
 int count = 0;
 System.out.println("마라톤 참가자를 등록합니다.");
 System.out.println("참가자 이름을 입력해주세요 (종료하려면 'exit' 입력):");
 while(true) {
 System.out.print("참가자 이름 : ");
 String name = sc.next();
 if(name.equals("exit")) {
 break;
 }
 count++;
 map.put(count, new Member(count, name));
 System.out.println("등록 완료! 현재 참가자 수 : " + count);
 }
 System.out.println("\n 참가자 목록:");
 for (Member m : map.values()){
 System.out.println(m);
 }
 }
}

```

#### [실행 결과]

```

/Users/hataeyeong/Library/Java/JavaVirtualMachines/...
마라톤 참가자를 등록합니다.
참가자 이름을 입력해주세요 (종료하려면 'exit' 입력):
참가자 이름 : hallym
등록 완료! 현재 참가자 수 : 1
참가자 이름 : content
등록 완료! 현재 참가자 수 : 2
참가자 이름 : bigdata
등록 완료! 현재 참가자 수 : 3
참가자 이름 : exit

참가자 목록:
참가번호 1번 : hallym
참가번호 2번 : content
참가번호 3번 : bigdata

종료 코드 0(으)로 완료된 프로세스

```

#### □ 응용 프로그래밍

- (9) 학생들의 정보를 ArrayList에 저장하고 검색할 수 있는 프로그램을 작성하세요. 학생들의 정보는 Student라는 클래스를 사용합니다. Student는 학생의 이름, 전화번호를 필드로 가지며 접근자와 설정자, 객체내용을 문자열로 반환하는 메소드로 구성합니다. ArrayListTest 클래스에서 학생들의 정보를 ArrayList에 추가, 검색, 삭제

하는 간단한 메뉴를 제공합니다

```
public class ArrayListTest{
 public static void main(String[] args) {
 ArrayList<Student> list = new ArrayList<Student>();
 Scanner sc = new Scanner(System.in);

 for(int i=0;i<10; i++){
 list.add(new Student("name"+i, "phone"+i));
 }
 for(Student s : list)
 System.out.println(s);

 //사용자에게 이름을 입력 받아 추가 - 본인작성
 //사용자에게 이름을 입력 받아 전화번호 검색 - 본인 작성
 //사용자에게 이름을 입력 받아 삭제 - 본인 작성
 //모든 사용자 정보 출력 - 본인 작성
 }
}
```

---

#### [프로그램 소스]

```
package practice9;

class Student {
 private String name;
 private String phone;

 public Student(String name, String phone) {
 this.name = name;
 this.phone = phone;
 }

 public String getName() {
 return name;
 }

 public String getPhone() {
 return phone;
 }

 @Override
 public String toString() {
 return "이름 : " + name + ", 전화번호 : " + phone;
 }
}
```

```
package practice9;

import java.util.ArrayList;
import java.util.Scanner;

public class ArrayListTest{
 public static void main(String[] args) {
 ArrayList<Student> list = new ArrayList<Student>();
 Scanner sc = new Scanner(System.in);
```

```

for (int i = 0; i < 10; i++) {
 list.add(new Student("name" + i, "phone" + i));
}
for (Student s : list)
 System.out.println(s);

boolean running = true;
while(running) {
 System.out.println("1. 등록, 2. 검색, 3. 삭제, 4. 전체조회, 5. 종료");
 int menu = 0;
 System.out.print("메뉴 번호를 입력해주세요 : ");
 menu = sc.nextInt();
 switch (menu){
 case 1: //사용자에게 이름을 입력 받아 추가 - 본인작성
 System.out.print("이름을 입력하세요 : ");
 String name = sc.next();
 System.out.print("전화번호를 입력하세요 : ");
 String phone = sc.next();
 list.add(new Student(name, phone));
 System.out.println("등록완료되었습니다.");
 break;
 case 2: //사용자에게 이름을 입력 받아 전화번호 검색 - 본인 작성
 System.out.print("전화번호 검색할 이름을 입력해주세요 : ");
 String search = sc.next();
 boolean found = false;

 for (Student s : list) {
 if (s.getName().equals(search)) {
 System.out.println(search + "의 전화번호 : " + s.getPhone());
 found = true;
 break;
 }
 }
 if (!found) {
 System.out.println(search + "의 정보를 찾을 수 없습니다.");
 }
 break;
 case 3: //사용자에게 이름을 입력 받아 삭제 - 본인 작성
 System.out.print("삭제할 이름을 입력해주세요 : ");
 String del = sc.next();
 Student toRemove = null;
 // 향상된 for 문(for-each)에서 직접 삭제하면
 ConcurrentModificationException 발생할 수 있다.
 for (Student s : list) {
 if (s.getName().equals(del)) {
 toRemove = s;
 break;
 }
 }
 if (toRemove != null) {
 list.remove(toRemove);
 System.out.println(del + "의 정보가 삭제되었습니다.");
 } else {
 System.out.println(del + "의 정보를 찾을 수 없습니다.");
 }
 break;
 case 4: //모든 사용자 정보 출력 - 본인 작성
 for (Student s : list) {
 System.out.println(s);
 }
 break;
 case 5:
 running = false;
 break;
 default:
 }
}

```

```
 System.out.println("잘못된 입력하였습니다.");
 }
}
}
```

[실행 결과]

```
실행 ArrayListTest ×
/Users/hataeyeong/Library/Java/JavaVirtual
이름 : name0, 전화번호 : phone0
이름 : name1, 전화번호 : phone1
이름 : name2, 전화번호 : phone2
이름 : name3, 전화번호 : phone3
이름 : name4, 전화번호 : phone4
이름 : name5, 전화번호 : phone5
이름 : name6, 전화번호 : phone6
이름 : name7, 전화번호 : phone7
이름 : name8, 전화번호 : phone8
이름 : name9, 전화번호 : phone9
1. 등록, 2. 검색, 3. 삭제, 4. 전체조회, 5. 종료
메뉴 번호를 입력해주세요 : 6
잘못된 입력하였습니다.
1. 등록, 2. 검색, 3. 삭제, 4. 전체조회, 5. 종료
메뉴 번호를 입력해주세요 : 1
이름을 입력하세요 : hallym
전화번호를 입력하세요 : 010-1234-1234
등록완료되었습니다.
```

```
1. 등록, 2. 검색, 3. 삭제, 4. 전체조회, 5. 종료
메뉴 번호를 입력해주세요 : 2
전화번호 검색할 이름을 입력해주세요 : name2
name2의 전화번호 : phone2
1. 등록, 2. 검색, 3. 삭제, 4. 전체조회, 5. 종료
메뉴 번호를 입력해주세요 : 3
삭제할 이름을 입력해주세요 : name3
name3의 정보가 삭제되었습니다.
1. 등록, 2. 검색, 3. 삭제, 4. 전체조회, 5. 종료
메뉴 번호를 입력해주세요 : 4
이름 : name0, 전화번호 : phone0
이름 : name1, 전화번호 : phone1
이름 : name2, 전화번호 : phone2
이름 : name4, 전화번호 : phone4
이름 : name5, 전화번호 : phone5
이름 : name6, 전화번호 : phone6
이름 : name7, 전화번호 : phone7
이름 : name8, 전화번호 : phone8
이름 : name9, 전화번호 : phone9
이름 : hallym, 전화번호 : 010-1234-1234
1. 등록, 2. 검색, 3. 삭제, 4. 전체조회, 5. 종료
메뉴 번호를 입력해주세요 : 5

종료 코드 0(으)로 완료된 프로세스
```

(10) 초등학생 영어 학습을 돕기 위한 프로그램을 HashMap을 이용하여 작성하시오.

- 영어 단어(key), 의미(value)로 구성한다
  - 새로운 단어를 추가하거나 기존의 단어를 삭제할 수 있도록 한다
  - HashMap에 저장된 모든 데이터 출력.
  - HashMap에 다음의 데이터로 초기화한다
- ```
String[] word= {"actor", "age", "answer", "arm", "autumn", "air", "airplane"};
String[] mean= {"배우", "나이", "대답", "팔", "가을", "공기", "비행기"};
```
- 힌트) 삭제할 때 containsKey(), remove() 메소드 사용


```

===== Map에 저장된 결과 =====
key= actor, value= 배우
key= answer, value= 대답
key= airplane, value= 비행기
key= autumn, value= 가을
key= air, value= 공기
key= arm, value= 팔
key= age, value= 나이

삭제 하고자 하는 영단어를 입력하시오 >> air
air 단어를 삭제합니다.
===== Map에 저장된 결과 =====
key= actor, value= 배우
key= answer, value= 대답
key= airplane, value= 비행기
key= autumn, value= 가을
key= arm, value= 팔
key= age, value= 나이

추가 하고자 하는 영단어를 입력하시오 >> apple
단어의 뜻을 입력하세요 >> 사과
===== Map에 저장된 결과 =====
key= actor, value= 배우
key= apple, value= 사과
key= answer, value= 대답
key= airplane, value= 비행기
key= autumn, value= 가을
key= arm, value= 팔
key= age, value= 나이

```

[프로그램 소스]

```

package practice10;

class English {
    private String word;
    private String mean;

    public English(String word, String mean) {
        this.word = word;
        this.mean = mean;
    }

    @Override
    public String toString() {
        return mean;
    }
}

```

```

package practice10;

import java.util.*;

public class Main {
    public static void printMap(Map<String, English> map) {
        System.out.println("===== Map 에 저장된 결과 =====");
        for (String key : map.keySet()) {
            English value = map.get(key);
            System.out.println("key=" + key + ", value=" + value);
        }
    }

    public static void main(String[] args) {
        Map<String, English> map = new HashMap<>();
        Scanner sc = new Scanner(System.in);

        String[] word = {"actor", "age", "answer", "arm", "autumn", "air",
            "airplane"};
        String[] mean = {"배우", "나이", "대답", "팔", "가을", "공기", "비행기"};

        for (int i = 0; i < word.length; i++) {
            map.put(word[i], new English(word[i], mean[i]));
        }
    }
}

```

```
printMap(map) ;

System.out.print("\n 삭제하고자 하는 영단어를 입력하시오 >> ");
String del = sc.next();

if (map.containsKey(del)) {
    map.remove(del);
    System.out.println(del + " 단어를 삭제합니다.");
} else {
    System.out.println(del + " 단어가 존재하지 않습니다.");
}

printMap(map) ;

System.out.print("\n 추가 하고자 하는 영단어를 입력하시오 >> ");
String addWord = sc.next();
System.out.print("단어의 뜻을 입력하시오 >> ");
String addMean = sc.next();
map.put(addWord, new English(addWord, addMean));

printMap(map) ;
}
}
```

[실행 결과]

실행

Main

Main



/Users/hataeyeong/Library/Java/JavaVir

===== Map에 저장된 결과 =====

key=actor, value=배우

key=answer, value=대답

key=airplane, value=비행기

key=autumn, value=가을

key=air, value=공기

key=arm, value=팔

key=age, value=나이

삭제하고자 하는 영단어를 입력하시오 >> *air*

air 단어를 삭제합니다.

===== Map에 저장된 결과 =====

key=actor, value=배우

key=answer, value=대답

key=airplane, value=비행기

key=autumn, value=가을

key=arm, value=팔

key=age, value=나이

추가 하고자 하는 영단어를 입력하시오 >> *apple*

단어의 뜻을 입력하시오 >> *사과*

===== Map에 저장된 결과 =====

key=actor, value=배우

key=apple, value=사과

key=answer, value=대답

key=airplane, value=비행기

key=autumn, value=가을

key=arm, value=팔

key=age, value=나이

종료 코드 0(으)로 완료된 프로세스

