

자바프로그래밍및실습 과제 3

214823 박종현

May 2022

1 과제 1

1.1 소스코드

```
1  /**
2   * @author @ShapeLayer
3   */
4
5  /**
6   * Average.java
7   * CLI 환경에서 실행하고자 한다면 `java Average.java 1 2 3 5 7 8 1 2 3 4`를 입력하세요.
8   */
9
10 public class Main {
11     public static void main(String[] args) {
12         int sums = 0; // 합계 기록할 변수 선언
13         // `args` 변수의 길이만큼 반복문 시작
14         for (int i = 0; i < args.length; i++) {
15             // `args[i]`는 String형 값이므로 int형으로 캐스팅한 뒤 `sums`에 가산
16             sums += Integer.parseInt(args[i]);
17         }
18         // `sums`와 `args.length` 모두 정수이므로 둘 중 하나를 실수로 캐스팅해야 실수 계산이 수행됨
19         // 실수로 캐스팅하지 않으면 몫 연산 수행
20         System.out.println("입력받은 인자 값의 평균은 : " + sums/(float)args.length);
21     }
22 }
```

1.2 실행 예제

1.2.1 예제 1

```
java Average.java 1 2 3 5 7 8 1 2 3 4
입력받은 인자 값의 평균은 : 3.6
```

```
shapelayer@ShapeLayer-MacBookAir Prob1 % java Average.java 1 2 3 5 7 8 1 2 3 4
입력받은 인자 값의 평균은 : 3.6
```

1.2.2 예제 2

```
java Average.java 1 2 3 4 5 6 7 8 9 10
입력받은 인자 값의 평균은 : 5.5
```

2 과제 2

2.1 소스코드

```
1  /**
2   * @author @ShapeLayer
3   */
4
5  import java.util.Scanner;
6
7  public class Main {
8     public static void main(String[] args) {
9         // 표준 입력을 위해 Scanner 객체 선언
10         Scanner sc = new Scanner(System.in);
11
12         // 문제 제시 값
```

```

13 String course[] = {"java", "c++", "HTML5", "컴퓨터구조", "안드로이드"};
14 int score[] = {95, 88, 76, 62, 55};
15 String name;
16
17 // 입력값이 "그만"일 때까지 프로그램이 작동해야하므로 무한 루프 시작
18 while(true) {
19     // 과목 이름 입력
20     System.out.print("과목 이름 >> ");
21     name = sc.next();
22
23     // 입력값이 "그만"인 경우 프로그램 종료
24     if (name.equals("그만")) { break; }
25
26     // 검색 중 입력값에 해당하는 값이 존재하고 출력했는지 여부를 확인하기 위해
27     // 부울형 `flag` 변수 선언
28     // 아직 존재하는지 확인하지 않았고, 결과를 출력하지 않았으므로 false로 지정
29     boolean flag = false;
30     for (int i = 0; i < course.length; i++) {
31         // 과제 첨부 파일의 "실행 예"에서 Java를 입력했는데 java의 결과가 출력되는 것을 확인할 수 있음
32         // 따라서 이 프로그램은 대소문자를 구분하지 않는 것으로 처리함
33         // * 입력값과 키(과목)값 모두 소문자로 변환한 후 두 값이 같은지 확인
34         if (course[i].toLowerCase().equals(name.toLowerCase())) {
35             System.out.println(String.format("%s의 점수는 %d", name, score[i]));
36             flag = true; // 대상이 존재하므로 `flag`를 true로 변경
37         }
38     }
39     // `flag`가 false라면 대상이 존재하지 않고 결과를 출력한 사실도 없으므로 없는 과목이라고 출력
40     if (!flag) System.out.println("없는 과목입니다.");
41 }
42 }
43 }

```

2.2 실행 예제

2.2.1 예제 1

```

java Prob2.java
과목 이름 >> Jaba
없는 과목입니다.
과목 이름 >> Java
Java의 점수는 95
과목 이름 >> 안드로이드
안드로이드의 점수는 55
과목 이름 >> 그만

```

```

shapelayer@Shapelayer-MacBookAir java % java Prob2.java
과목 이름 >> Jaba
없는 과목입니다.
과목 이름 >> Java
Java의 점수는 95
과목 이름 >> 안드로이드
안드로이드의 점수는 55
과목 이름 >> 그만

```

2.2.2 예제 2

```

java Prob2.java
과목 이름 >> HTML5
HTML5의 점수는 76
과목 이름 >> html5

```

html5의 점수는 76
과목 이름 >> 그만

3 과제 3

3.1 소스코드

```
1  /**
2   * @author @ShapeLayer
3   */
4
5  import java.util.Scanner;
6  // 예외 처리를 위해 InputMismatchException 클래스 импорт
7  import java.util.InputMismatchException;
8
9  public class Main {
10     public static void main(String[] args) {
11         int n, m;
12
13         // 문제 제시 코드
14         Scanner scanner = new Scanner(System.in);
15         while (true) {
16             System.out.print("곱하고자 하는 두수 입력>>");
17
18             // 예외 처리를 위해 try-catch문 사용
19             try {
20                 n = scanner.nextInt();
21                 m = scanner.nextInt();
22                 break;
23             } catch (InputMismatchException e) {
24                 // `InputMismatchException` 오류를 캐치하도록 지정.
25                 // e는 오류(error) 정보를 담고 있음
26                 System.out.println("실수는 입력하면 안됩니다.");
27                 scanner.nextLine();
28             }
29         }
30         System.out.println(n + "x" + m + "=" + n*m);
31         scanner.close();
32     }
33 }
```

3.2 실행 예제

3.2.1 예제 1

```
java Prob3.java
곱하고자 하는 두수 입력>>2.5 4
실수는 입력하면 안됩니다.
곱하고자 하는 두수 입력>>4 3.5
실수는 입력하면 안됩니다.
곱하고자 하는 두수 입력>>4 3
4x3=12
```

```
shapelayer@ShapeLayer-MacBookAir java % java Prob3.java
곱하고자 하는 두 수 입력 >> 2.5 4
실수는 입력하면 안됩니다.
곱하고자 하는 두 수 입력 >> 4 3.5
실수는 입력하면 안됩니다.
곱하고자 하는 두 수 입력 >> 4 3
4x3=12
```

3.2.2 예제 2

문제에서 제시하지 않은 조건: 채점 대상 아님

1. InputMismatchException은 제시한 입력 타입과 맞지 않으면 발생: String형도 Integer가 아니므로 "실수는 입력하면 안됩니다." 출력됨
2. 문제 제시 코드에서 변수 타입을 Integer로 설정하였으므로, 연산 결과가 Integer 범위를 초과하면 잘못된 값이 출력됨

```
java Prob3.java
곱하고자 하는 두 수 입력 >> 문자열입력 문자열입력
실수는 입력하면 안됩니다.
곱하고자 하는 두 수 입력 >> 10000000 10000000
10000000x10000000=276447232
```

4 과제 4

4.1 소스코드

```
1  /**
2   * @author @ShapeLayer
3   */
4
5  import java.util.Scanner;
6
7  public class Main {
8      public static void main(String[] args) {
9          // 표준 입력을 위해 Scanner 객체 선언
10         Scanner sc = new Scanner(System.in);
11         String gets;
12
13         // 문제 제시 값
14         String str[] = {"가위", "바위", "보"};
15
16         System.out.println("컴퓨터와 가위 바위 보 게임을 합니다.");
17         // "그만"이 입력될때까지 반복해야하므로 무한 루프 선언
18         while (true) {
19             System.out.print("가위 바위 보! >> ");
20             gets = sc.next();
21             // 문제 제시 코드
22             int n = (int)(Math.random()*3);
23
24             // "그만"이 입력된다면 무한 루프 종료
25             if (gets.equals("그만")) break;
26
27             // 문제에서 gets.equals()를 사용하여 값을 확인하라고 지정하였으니 아래와 같이 케이스별로 처리
28             if (str[n].equals("가위")) {
29                 if (gets.equals("가위"))
30                     System.out.println(String.format("사용자 = %s, 컴퓨터 %s, 비겼습니다.", gets, str[n]));
31                 if (gets.equals("바위"))
32                     System.out.println(String.format("사용자 = %s, 컴퓨터 %s, 사용자가 이겼습니다.", gets, str[n]));
```

```

33     if (gets.equals("보"))
34         System.out.println(String.format("사용자 = %s, 컴퓨터 %s, 컴퓨터가 이겼습니다.", gets, str[n]));
35     }
36     else if (str[n].equals("바위")) {
37         if (gets.equals("가위"))
38             System.out.println(String.format("사용자 = %s, 컴퓨터 %s, 컴퓨터가 이겼습니다.", gets, str[n]));
39         if (gets.equals("바위"))
40             System.out.println(String.format("사용자 = %s, 컴퓨터 %s, 비겼습니다.", gets, str[n]));
41         if (gets.equals("보"))
42             System.out.println(String.format("사용자 = %s, 컴퓨터 %s, 사용자가 이겼습니다.", gets, str[n]));
43     }
44     else if (str[n].equals("보")) {
45         if (gets.equals("가위"))
46             System.out.println(String.format("사용자 = %s, 컴퓨터 %s, 사용자가 이겼습니다.", gets, str[n]));
47         if (gets.equals("바위"))
48             System.out.println(String.format("사용자 = %s, 컴퓨터 %s, 컴퓨터가 이겼습니다.", gets, str[n]));
49         if (gets.equals("보"))
50             System.out.println(String.format("사용자 = %s, 컴퓨터 %s, 비겼습니다.", gets, str[n]));
51     }
52     // fallback
53     // "가위", "바위", "보" 외의 값을 입력했을 때 처리는 제시하지 않았으므로
54     // 별도의 처리를 거치지 않고 루프를 계속함 (= 계속해서 입력을 받아옴)
55 }
56 System.out.println("게임을 종료합니다.");
57 }
58 }

```

4.2 실행 예제

4.2.1 예제 1

```

java Prob4.java
컴퓨터와 가위 바위 보 게임을 합니다.
가위 바위 보! >> 바위
사용자 = 바위, 컴퓨터 가위, 사용자가 이겼습니다.
가위 바위 보! >> 가위
사용자 = 가위, 컴퓨터 보, 사용자가 이겼습니다.
가위 바위 보! >> 그만
게임을 종료합니다.

```

```

[shapelayer@ShapeLayer-MacBookAir java % java Prob4.java
컴퓨터와 가위 바위 보 게임을 합니다.
가위 바위 보! >> 바위
사용자 = 바위, 컴퓨터 가위, 사용자가 이겼습니다.
가위 바위 보! >> 가위
사용자 = 가위, 컴퓨터 바위, 컴퓨터가 이겼습니다.
가위 바위 보! >> 그만
게임을 종료합니다.

```

5 과제 5

5.1 소스코드

```

1  /**
2   * @author @ShapeLayer
3   */
4
5  // 코드 Javadoc 문서화
6  // 참조: https://docs.oracle.com/en/java/javase/17/docs/specs/javadoc/doc-comment-spec.html

```

```

7
8 import java.util.Scanner;
9
10 public class Main {
11     public static void main(String[] args) {
12         // 문제 제시 코드
13         TV myTV = new TV("LG", 2017, 32);
14         myTV.show();
15     }
16 }
17
18 /**
19  * TV 클래스
20  *
21  * TV 클래스는 생산된 TV에 대한 정보를 담을 수 있습니다.
22  * 이름, 생산자(생산업체), 인치 정보를 담고 처리할 수 있고, 그 내용을 출력할 수 있습니다.
23  */
24 class TV {
25     /**
26      * TV 이름을 표현합니다.
27      */
28     String name;
29
30     /**
31      * 생산자(생산업체)를 표현합니다.
32      */
33     int manufactures;
34
35     /**
36      * 크기를 인치 단위로 표현합니다.
37      */
38     int inch;
39
40     /**
41      * TV 객체를 생성합니다.
42      *
43      * @param name TV의 이름
44      * @param manufactures 생산자(생산업체)
45      * @param inch 크기(인치)
46      */
47     public TV(String name, int manufactures, int inch) {
48         // `this.name`은 TV 객체의 name 속성, `name`은 생성자의 매개변수
49         // 아래와 같이 작성한다면 생성자의 매개변수를 객체의 각 속성으로 저장할 수 있음
50         this.name = name;
51         this.manufactures = manufactures;
52         this.inch = inch;
53     }
54
55     /**
56      * TV 정보를 콘솔에 표준 출력합니다.
57      *
58      * @return {@code null}
59      */
60     public void show() {
61         System.out.println(String.format("%s에서 만든 %d년형 %d인치 TV", this.name, this.manufactures,
62 ↵         this.inch));
63     }
64 }

```

5.2 실행 예제

5.2.1 예제 1

java Prob5.java
LG에서 만든 2017년형 32인치 TV

```
[shapelayer@ShapeLayer-MacBookAir java % java Prob5.java  
LG에서 만든 2017년형 32인치 TV
```

6 과제 6

6.1 소스코드

```
1  /**  
2   * @author @ShapeLayer  
3   */  
4  
5  // 코드 Javadoc 문서화  
6  // 참조: https://docs.oracle.com/en/java/javase/17/docs/specs/javadoc/doc-comment-spec.html  
7  
8  import java.util.Scanner;  
9  
10 public class Main {  
11     public static void main(String[] args) {  
12         // 문제 제시 코드  
13         Scanner scanner = new Scanner(System.in);  
14         System.out.print("수학, 과학, 영어순으로 3개의 점수 입력>>");  
15         int math = scanner.nextInt();  
16         int science = scanner.nextInt();  
17         int english = scanner.nextInt();  
18         Grade me = new Grade(math, science, english);  
19         System.out.println("평균은 " + me.average());  
20         // average()는 평균을 계산하여 리턴  
21         scanner.close();  
22     }  
23 }  
24  
25 /**  
26  * Grade 클래스  
27  * Grade 클래스는 수학, 과학, 영어 세 과목의 점수를 처리할 수 있는 클래스입니다.  
28  * 세 과목을 한 번에 관리할 수 있고, 세 과목의 평균을 쉽게 확인할 수 있도록 지원합니다.  
29  */  
30 class Grade {  
31     // 자바에서 접근 제어자는 기본값이 private이므로 접근 제어자를 생략하면 기본값 private로 지정됨  
32  
33     /**  
34      * 수학 과목 점수  
35      */  
36     int math;  
37  
38     /**  
39      * 과학 과목 점수  
40      */  
41     int science;  
42  
43     /**  
44      * 영어 과목 점수  
45      */  
46     int english;  
47
```



```

48  /**
49   * Grade 객체를 생성합니다.
50   * @param math 수학 과목 점수
51   * @param science 과학 과목 점수
52   * @param english 영어 과목 점수
53   */
54  public Grade(int math, int science, int english) {
55      this.math = math;
56      this.science = science;
57      this.english = english;
58  }
59
60  /**
61   * 수학 과학 영어 세 과목의 평균값을 정수로 반환합니다.
62   * @return {@code int} 세 과목의 평균값
63   */
64  int average() {
65      return (this.math + this.science + this.english) / 3;
66  }
67  }

```

6.2 실행 예제

6.2.1 예제 1

```

java Prob6.java
수학, 과학, 영어순으로 3개의 점수 입력>>90 88 96
평균은 91

```

```

[shapelayer@ShapeLayer-MacBookAir java % java Prob6.java
수학, 과학, 영어순으로 3개의 점수 입력>>90 88 96
평균은 91

```

6.2.2 예제 2

```

java Prob6.java
수학, 과학, 영어순으로 3개의 점수 입력>>85 90 95
평균은 90

```

7 과제 7

7.1 소스코드

```

1  /**
2   * @author @ShapeLayer
3   */
4
5  // 코드 Javadoc 문서화
6  // 참조: https://docs.oracle.com/en/java/javase/17/docs/specs/javadoc/doc-comment-spec.html
7
8  import java.util.Scanner;
9
10 /**
11  * Phonebook 클래스
12  *
13  * Phonebook 클래스는 주소록 애플리케이션 클래스입니다.
14  * 사용자는 다른 사람들의 프로필을 등록하고, 등록된 프로필을 참조할 수 있습니다.

```

```

15  */
16  public class Phonebook {
17      /**
18       * 프로그램 진입 점
19       * @param args
20       */
21      public static void main(String[] args) {
22          Scanner sc = new Scanner(System.in);
23          System.out.print("인원수 >> ");
24
25          // 인원수 입력 받아옴
26          int n = sc.nextInt();
27          // 입력값을 크기로 갖는 Phone 배열 생성
28          Phone[] phones = new Phone[n];
29
30          /* 프로필(전화번호부) 등록 프로세스 */
31          // for문으로 배열의 요소들을 순차적으로 탐색
32          // 사용자로부터 입력받고, 입력받은 값으로 초기화 진행
33          for (int i = 0; i < n; i++) {
34              System.out.print("이름과 전화번호 (이름과 번호는 빈 칸없이 입력)>> ");
35              String name = sc.next();
36              String tel = sc.next();
37              // Phone 배열의 i번째 요소를 사용자의 입력값으로 초기화
38              phones[i] = new Phone(name, tel);
39          }
40          System.out.println("저장되었습니다...");
41
42          // "그만"이 입력될 때까지 계속해서 검색해야하므로 무한 루프 시작
43          while (true) {
44              System.out.print("검색할 이름 >> ");
45              String gets = sc.next();
46
47              // 입력값이 "그만"이라면 루프 종료
48              if (gets.equals("그만")) break;
49
50              // 검색 결과를 잘 찾아내고 출력했는지 확인하기 위해 `flag` 변수 선언
51              // 선언 시기에는 아직 검색을 시작하지 않았으므로 false로 설정
52              boolean flag = false;
53
54              // 검색 시작
55              for (int i = 0; i < n; i++) {
56                  if (phones[i].getName().equals(gets)) {
57                      // 검색 결과가 존재하므로 `flag`를 true로 지정
58                      flag = true;
59                      // 결과 출력
60                      System.out.println(String.format("%s의 번호는 %s입니다.", phones[i].getName(), phones[i].getTel()));
61                  }
62              }
63              // `flag`가 `false`라면 검색 결과가 존재하지 않았다는 것이므로 fallback 처리
64              // 문제에서 제시한 예제에서 "이(가) 없습니다."로 출력함
65              if (!flag) { System.out.println(String.format("%s 이(가) 없습니다.", gets)); }
66          }
67          System.out.println("프로그램을 종료합니다.");
68      }
69  }
70
71  /**
72   * Phone 클래스
73   *
74   * Phone 클래스는 Phonebook 클래스가 관리하는 프로필 단위입니다.
75   * 하나의 프로필은 하나의 Phone 객체입니다.
76   */

```

```

77 class Phone {
78     /**
79      * 대상의 이름입니다.
80      */
81     private String name;
82     /**
83      * 대상의 전화번호 정보입니다.
84      */
85     private String tel;
86
87     /**
88      * Phone 객체를 생성합니다.
89      * @param name 대상의 이름
90      * @param tel 대상의 전화번호
91      */
92     public Phone(String name, String tel) {
93         this.name = name; this.tel = tel;
94     }
95
96     /**
97      * {@code getter} 이름을 반환합니다.
98      * @return {@code String} 이름
99      */
100    public String getName() { return name; }
101
102    /**
103     * {@code getter} 전화번호를 반환합니다.
104     * @return {@code String} 전화번호
105     */
106    public String getTel() { return tel; }
107 }

```

7.2 실행 예제

7.2.1 예제 1

```

java Phonebook.java
인원수 >> 3
이름과 전화번호 (이름과 번호는 빈 칸없이 입력)>> 황기태 777-7777
이름과 전화번호 (이름과 번호는 빈 칸없이 입력)>> 나명품 999-7777
이름과 전화번호 (이름과 번호는 빈 칸없이 입력)>> 최자바 333-1234
저장되었습니다...
검색할 이름 >> 황기순
황기순 이(가) 없습니다.
검색할 이름 >> 최자바
최자바의 번호는 333-1234입니다.
검색할 이름 >> 그만
프로그램을 종료합니다.

```

```

[shapelayer@ShapeLayer-MacBookAir Prob7 % java Phonebook.java
인원수 >> 3
이름과 전화번호 (이름과 번호는 빈 칸없이 입력)>> 황기태 777-7777
이름과 전화번호 (이름과 번호는 빈 칸없이 입력)>> 나명품 999-7777
이름과 전화번호 (이름과 번호는 빈 칸없이 입력)>> 최자바 333-1234
저장되었습니다...
검색할 이름 >> 황기순
황기순 이(가) 없습니다.
검색할 이름 >> 최자바
최자바의 번호는 333-1234입니다.
검색할 이름 >> 그만
프로그램을 종료합니다.

```

8 과제 8

8.1 소스코드

```
1  /**
2   * @author @ShapeLayer
3   */
4
5  // 코드 Javadoc 문서화
6  // 참조: https://docs.oracle.com/en/java/javase/17/docs/specs/javadoc/doc-comment-spec.html
7
8  import java.util.Arrays;
9
10 public class StaticEx {
11     public static void main(String[] args) {
12         // 문제 제시 코드
13         int[] array1 = { 1, 5, 7, 9 };
14         int[] array2 = { 3, 6, -1, 100, 77 };
15         int[] array3 = ArrayUtil.concat(array1, array2);
16         ArrayUtil.print(array3);
17     }
18 }
19
20 /**
21  * ArrayUtil 클래스
22  *
23  * ArrayUtil 클래스는 배열을 좀 더 쉽게 제어할 수 있는 메서드를 포함하고 있습니다.
24  */
25 class ArrayUtil {
26     /**
27      * 두 정수 배열을 하나로 병합합니다.
28      * @param a {@code int[]} 첫 번째 정수 배열
29      * @param b {@code int[]} 두 번째 정수 배열
30      * @return {@code int[]} 배열 병합 결과
31      */
32     public static int[] concat(int[] a, int[] b) {
33         // 배열 a와 b를 연결한 새로운 배열 리턴
34         int[] newArr = new int[a.length + b.length];
35         System.arraycopy(a, 0, newArr, 0, a.length);
36         System.arraycopy(b, 0, newArr, a.length, b.length);
37
38         /* 시간복잡도 O(n) 인 처리 방법
39         for (int i = 0; i < a.length; i++) newArr[i] = a[i];
40         for (int i = 0; i < b.length; i++) newArr[a.length + i] = b[i];
41         */
42         return newArr;
43     }
44
45     /**
46      * 정수 배열을 출력합니다.
47      *
48      * @param a {@code int[]} 정수 배열
49      */
50     public static void print(int[] a) {
51         // 배열 a를 출력
52         // System.out.println(Arrays.toString(a));
53
54         // print()는 출력 후 개행하지 않으므로 한 줄을 여러 줄에 걸쳐서 출력
55         System.out.print("[ ");
56         for (int i = 0; i < a.length; i++)
57             System.out.print(a[i] + " ");
58         System.out.println("]");
59     }
60 }
```

```
59 }
60 }
```

8.2 실행 예제

8.2.1 예제 1

```
java StaticEx.java
[ 1 5 7 9 3 6 -1 100 77 ]
```

```
shapelayer@ShapeLayer-MacBookAir Prob8 % java StaticEx.java
[ 1 5 7 9 3 6 -1 100 77 ]
```

9 과제 9

9.1 소스코드

```
1  /**
2   * @author @ShapeLayer
3   */
4
5  // 코드 Javadoc 문서화
6  // 참조: https://docs.oracle.com/en/java/javase/17/docs/specs/javadoc/doc-comment-spec.html
7
8  import java.util.Scanner;
9
10 /**
11  * DicApp 클래스
12  *
13  * DicApp은 사전 애플리케이션 클래스입니다.
14  * 사용자는 사전을 로드하여 사전에 기록되어 있는 단어를 검색할 수 있습니다.
15  */
16 public class DicApp {
17     /**
18      * 단어가 한글 자모로 구성될 때 마지막 글자에 종성이 존재하는지 확인합니다.
19      *
20      * @param word 확인할 단어
21      * @return 마지막 글자에 종성이 존재하는지 여부
22      */
23     public static boolean isComplete(String word) {
24         char lastWord = word.charAt(word.length() - 1);
25         // 마지막 글자가 한국어가 아닌 경우 `false` 리턴
26         if (lastWord < 0xAC00 || lastWord > 0xD7A3) { return false; }
27         // 한글 자모를 분리하여 종성이 존재하는지 확인
28         // 참조 https://hanpsy.tistory.com/2
29         return (lastWord - 0xAC00) % 28 > 0; // 0인 경우 = 종성 데이터 없음 (= 종성 없음)
30     }
31
32     /**
33      * 프로그램 진입점
34      * @param args
35      */
36     public static void main(String[] args) {
37         System.out.println("한영 단어 검색 프로그램입니다.");
38         Scanner sc = new Scanner(System.in);
39
40         // "종료"가 입력될 때까지 프로그램이 작동해야하므로 무한 루프 시작
41         while (true) {
42             System.out.print("한글 단어? ");
```

```

43     String gets = sc.next();
44     // "종료" 입력시 무한 루프 종료
45     if (gets.equals("종료")) break;
46
47     // Dictionary.kor2Eng 메서드는 정적 메서드이므로 객체를 생성하지 않고 호출
48     // 결과가 있다면 String 형으로 반환, 없다면 null 반환
49     String res = Dictionary.kor2Eng(gets);
50     System.out.println(String.format("%s%s %s",
51         gets,
52         DicApp.isComplete(gets) ? "은" : "는", // 은/는 여부 확인 메서드 호출. 종성이 있다면 "은", 없다면 "는"
53         res != null ? res : "저의 사전에 없습니다." // kor2Eng 결과가 null 이 아니라면 그대로 출력, null 이라면
↪     결과 없음 출력
54     ));
55 }
56 System.out.println("프로그램을 종료합니다.");
57 }
58 }
59
60 /**
61  * Dictionary 클래스
62  *
63  * Dictionary 클래스는 특정 문구를 키 값 삼아 대응되는 다른 문구로 변경할 수 있는 메서드를 제공합니다.
64  */
65 class Dictionary {
66     // 문제 제시 코드
67     /**
68      * 한국어 단어를 기록합니다.
69      */
70     private static String[] kor = {"사랑", "아기", "돈", "미래", "희망"};
71     /**
72      * 영어 단어를 기록합니다.
73      */
74     private static String[] eng = {"love", "baby", "money", "future", "hope"};
75
76     /**
77      * 입력받은 한국어 단어를 영어 단어로 반환합니다.
78      * 만약 목록에 등록되어 있지 않은 단어라면 {@code null}을 반환합니다.
79      *
80      * @param word
81      * @return {@code String?} 영단어 조회 결과를 반환합니다.
82      */
83     public static String kor2Eng(String word) {
84         // 시간 복잡도 O(n)의 처리 알고리즘 구현
85         for (int i = 0; i < kor.length; i++) {
86             // 만약 `word` (메서드의 매개변수)가 배열에 있다면 변환 결과 리턴
87             if (kor[i].equals(word)) {
88                 return eng[i];
89             }
90         }
91
92         // 이 부분까지 도달했다면 단어 목록에 대상 단어가 존재하지 않음을 의미함
93         // 단어가 존재하지 않으므로 null 반환
94         return null;
95     }
96 }

```

9.2 실행 예제

9.2.1 예제 1

```
java DicApp.java
한영 단어 검색 프로그램입니다.
한글 단어? 희망
희망은 hope
한글 단어? 아가
아가는 저의 사전에 없습니다.
한글 단어? 미래
미래는 future
한글 단어? 종료
프로그램을 종료합니다.
```

```
[shapelayer@ShapeLayer-MacBookAir Prob9 % java DicApp.java
한영 단어 검색 프로그램입니다.
한글 단어? 희망
희망은 hope
한글 단어? 아가
아가는 저의 사전에 없습니다.
한글 단어? 미래
미래는 future
한글 단어? 종료
프로그램을 종료합니다.]
```

9.2.2 예제 2

```
java DicApp.java
한영 단어 검색 프로그램입니다.
한글 단어? 사랑
사랑은 love
한글 단어? 아기
아기는 baby
한글 단어? 돈
돈은 money
한글 단어? 미래
미래는 future
한글 단어? 희망
희망은 hope
한글 단어? 종료
프로그램을 종료합니다.
```

10 과제 10

10.1 소스코드

```
1  /**
2   * @author @ShapeLayer
3   */
4
5  // 코드 Javadoc 문서화
6  // 참조: https://docs.oracle.com/en/java/javase/17/docs/specs/javadoc/doc-comment-spec.html
7
8  public class Main {
9      public static void main(String [] args) {
10         // ColorTV 객체 생성
11         ColorTV myTV = new ColorTV(32, 1024);
12         myTV.printProperty();
```

```

13     }
14 }
15
16 /**
17  * TV 클래스
18  *
19  * TV 클래스는 생산된 TV에 대한 정보를 담습니다.
20  */
21 class TV {
22     /**
23      * TV의 크기(인치) 입니다.
24      */
25     private int size;
26
27     /**
28      * TV 객체를 생성합니다.
29      * @param size 크기(인치)
30      */
31     public TV (int size) { this.size = size; }
32
33     /**
34      * TV의 크기를 반환합니다.
35      * @return {@code int} TV의 크기(인치)
36      */
37     protected int getSize() { return size; }
38 }
39
40 /**
41  * ColorTV 클래스
42  *
43  * TV 클래스를 상속함
44  * ColorTV 클래스는 생산된 컬러TV에 대한 정보를 담습니다.
45  */
46 class ColorTV extends TV {
47     /**
48      * TV가 표현할 수 있는 색상의 정보입니다.
49      */
50     private int colors;
51
52     /**
53      * ColorTV 객체를 생성합니다.
54      * @param size {@code int} 크기(인치)
55      * @param colors {@code int} 표현 가능한 색상 수
56      */
57     public ColorTV (int size, int colors) {
58         super(size);
59         this.colors = colors;
60     }
61
62     /**
63      * TV가 표현 가능한 색상의 개수를 반환합니다.
64      * @return {@code int} 표현 가능한 색상 수
65      */
66     protected int getColors() {
67         return this.colors;
68     }
69
70     /**
71      * TV의 속성을 표준 출력합니다.
72      */
73     public void printProperty() {
74         System.out.println(String.format("%d인치 %d컬러", this.getSize(), this.getColors()));

```



```
75 }
76 }
```

10.2 실행 예제

10.2.1 예제 1

```
java Prob10.java
32인치 1024컬러
```

```
[shapelayer@ShapeLayer-MacBookAir java % java Prob10.java
32인치 1024컬러
```

11 과제 11

11.1 소스코드

```
1  /**
2   * @author @ShapeLayer
3   */
4
5  // 코드 Javadoc 문서화
6  // 참조: https://docs.oracle.com/en/java/javase/17/docs/specs/javadoc/doc-comment-spec.html
7
8  public class Main {
9      public static void main(String [] args) {
10         IPTV iptv = new IPTV("192.1.1.2", 32, 2048);
11         iptv.printProperty();
12     }
13 }
14
15 /**
16  * TV 클래스
17  *
18  * TV 클래스는 생산된 TV에 대한 정보를 담습니다.
19  */
20 class TV {
21     /**
22      * TV의 크기(인치)입니다.
23      */
24     private int size;
25
26     /**
27      * TV 객체를 생성합니다.
28      * @param size 크기(인치)
29      */
30     public TV (int size) { this.size = size; }
31
32     /**
33      * TV의 크기를 반환합니다.
34      * @return {@code int} TV의 크기(인치)
35      */
36     protected int getSize() { return size; }
37 }
38
39 /**
40  * ColorTV 클래스
41  *
42  * TV 클래스를 상속함
```

```

43  * ColorTV 클래스는 생산된 컬러TV에 대한 정보를 담습니다.
44  */
45  class ColorTV extends TV {
46      /**
47       * TV가 표현할 수 있는 색상의 정보입니다.
48       */
49      private int colors;
50
51      /**
52       * ColorTV 객체를 생성합니다.
53       * @param size {@code int} 크기(인치)
54       * @param colors {@code int} 표현 가능한 색상 수
55       */
56      public ColorTV (int size, int colors) {
57          super(size);
58          this.colors = colors;
59      }
60
61      /**
62       * TV가 표현 가능한 색상의 개수를 반환합니다.
63       * @return {@code int} 표현 가능한 색상 수
64       */
65      protected int getColors() {
66          return this.colors;
67      }
68
69      /**
70       * TV의 속성을 표준 출력합니다.
71       */
72      public void printProperty() {
73          System.out.println(String.format("%d 인치 %d 컬러", this.getSize(), this.getColors()));
74      }
75  }
76
77  /**
78   * IPTV 클래스
79   *
80   * ColorTV를 상속함
81   * IPTV 클래스는 생산된 IPTV에 대한 정보를 담습니다.
82   */
83  class IPTV extends ColorTV {
84      /**
85       * TV의 IP 주소
86       */
87      private String ip;
88
89      /**
90       * IPTV 객체를 생성합니다.
91       * @param ip {@code String} IP 주소
92       * @param size {@code int} 크기(인치)
93       * @param colors {@code int} 표현 가능한 색상 수
94       */
95      IPTV(String ip, int size, int colors) {
96          super(size, colors);
97          this.ip = ip;
98      }
99
100     /**
101      * TV에 할당된 IP 주소를 반환합니다.
102      * @return {@code String} IP 주소
103      */
104     protected String getIP() { return this.ip; }

```

```

105  /**
106   * TV의 속성을 표준 출력합니다.
107   */
108  public void printProperty() {
109      System.out.println("나의 IPTV는 " + this.getIP() + " 주소의 " + this.getSize() + "인치 " + this.getColors() +
110      ↪   컬러");
111  }
112  }

```

11.2 실행 예제

11.2.1 예제 1

java Prob11.java
나의 IPTV는 192.1.1.2 주소의 32인치 2048컬러

```

shapelayer@ShapeLayer-MacBookAir java % java Prob11.java
나의 IPTV는 192.1.1.2 주소의 32인치 2048컬러

```

12 과제 12

12.1 소스코드

```

1  /**
2   * @author @ShapeLayer
3   */
4
5  // 코드 Javadoc 문서화
6  // 참조: https://docs.oracle.com/en/java/javase/17/docs/specs/javadoc/doc-comment-spec.html
7
8  public class Main {
9      public static void main(String[] args) {
10         // 문제 제시 코드
11         ColorPoint zeroPoint = new ColorPoint(); // (0,0) 위치의 BLACK 색점
12         System.out.println(zeroPoint.toString() + "입니다.");
13         ColorPoint cp = new ColorPoint(10, 10); // (10,10) 위치의 BLACK 색점
14         cp.setXY(5, 5);
15         cp.setColor("RED");
16         System.out.println(cp.toString() + "입니다.");
17     }
18 }
19
20 /**
21  * Point 클래스
22  *
23  * Point 클래스는 2차원 상의 한 점을 표현하는데 활용되는 클래스입니다.
24  */
25 class Point {
26     /**
27      * 점의 x, y좌표입니다.
28      */
29     private int x, y;
30
31     /**
32      * Point 객체를 생성합니다.
33      * @param x x좌표
34      * @param y y좌표
35      */

```

```

36 public Point(int x, int y) { this.x = x; this.y = y; }
37
38 /**
39  * x좌표를 반환합니다.
40  * @return {@code int} 점의 x좌표
41  */
42 public int getX() { return x; }
43
44 /**
45  * y좌표를 반환합니다.
46  * @return {@code int} 점의 y좌표
47  */
48 public int getY() { return y; }
49
50 /**
51  * 점의 x좌표와 y좌표를 이동시킵니다. (= setter)
52  * @param x 새 x좌표
53  * @param y 새 y좌표
54  */
55 protected void move(int x, int y) { this.x = x; this.y = y; }
56 }
57
58 /**
59  * ColorPoint 클래스
60  *
61  * Point 클래스를 상속함
62  * ColorPoint 클래스는 색상을 갖고 있는 한 점을 표현하는데 활용되는 클래스입니다.
63  */
64 class ColorPoint extends Point {
65     /**
66      * 점의 색상, 기본값은 {@code "BLACK"}입니다.
67      */
68     private String color = "BLACK";
69
70     /**
71      * ColorPoint 객체를 생성합니다.
72      * 생성된 객체의 x, y 좌표는 (0, 0)입니다.
73      */
74     public ColorPoint() { this(0, 0); }
75     /**
76      * ColorPoint 객체를 생성합니다.
77      * @param x x좌표
78      * @param y y좌표
79      */
80     public ColorPoint(int x, int y) { super(x, y); }
81
82     /**
83      * 객체의 x좌표와 y좌표를 설정합니다. (= setter)
84      * @param x x좌표
85      * @param y y좌표
86      */
87     public void setXY(int x, int y) { this.move(x, y); }
88
89     /**
90      * 객체의 색상을 설정합니다. (= setter)
91      * @param color 색상
92      */
93     public void setColor(String color) { this.color = color; }
94
95     /**
96      * 객체의 정보를 String형으로 변환해 반환합니다.
97      * @return {@code String} 객체의 정보

```

```

98     */
99     public String toString() { return String.format("%s색의 (%d, %d)의 점", this.color, this.getX(), this.getY());
100 ↵ }
    }

```

12.2 실행 예제

12.2.1 예제 1

```

java Prob12.java
BLACK색의 (0, 0)의 점입니다.
RED색의 (5, 5)의 점입니다.

```

```

shapelayer@ShapeLayer-MacBookAir java % java Prob12.java
BLACK색의 (0, 0)의 점입니다.
RED색의 (5, 5)의 점입니다.

```

13 과제 13

13.1 소스코드

```

1  /**
2   * @author @ShapeLayer
3   */
4
5  // 코드 Javadoc 문서화
6  // 참조: https://docs.oracle.com/en/java/javase/17/docs/specs/javadoc/doc-comment-spec.html
7
8  public class Main {
9      public static void main(String[] args) {
10         Point3D p = new Point3D(1,2,3); // 1,2,3은 각각 x, y, z 축의 값.
11         System.out.println(p.toString() + "입니다.");
12         p.moveUp(); // z축 1값 증가
13         System.out.println(p.toString() + "입니다.");
14         p.moveDown(); // z축 1값 감소
15         p.move(10, 10); // x, y 축의 해당 위치로 이동
16         System.out.println(p.toString() + "입니다.");
17         p.move(100, 200, 300); // x, y, z 축의 해당 위치로 이동
18         System.out.println(p.toString() + "입니다.");
19     }
20 }
21
22 /**
23  * Point 클래스
24  *
25  * Point 클래스는 2차원 상의 한 점을 표현하는데 활용되는 클래스입니다.
26  */
27 class Point {
28     /**
29      * 점의 x, y좌표입니다.
30      */
31     private int x, y;
32
33     /**
34      * Point 객체를 생성합니다.
35      * @param x x좌표
36      * @param y y좌표
37      */
38     public Point(int x, int y) { this.x = x; this.y = y; }

```

```

39
40 /**
41  * x좌표를 반환합니다.
42  * @return {@code int} 점의 x좌표
43  */
44 public int getX() { return x; }
45
46 /**
47  * y좌표를 반환합니다.
48  * @return {@code int} 점의 y좌표
49  */
50 public int getY() { return y; }
51
52 /**
53  * 점의 x좌표와 y좌표를 이동시킵니다. (= setter)
54  * @param x {@code 새 x좌표}
55  * @param y {@code 새 y좌표}
56  */
57 protected void move(int x, int y) { this.x = x; this.y = y; }
58 }
59
60 /**
61  * Point3D 클래스
62  *
63  * Point 클래스를 상속함
64  * Point3D 클래스는 3차원 상의 한 점을 표현하는데 활용되는 클래스입니다.
65  */
66 class Point3D extends Point {
67     /**
68      * 점의 z 좌표입니다.
69      */
70     private int z;
71
72     /**
73      * Point3D 객체를 생성합니다.
74      * @param x x좌표
75      * @param y y좌표
76      * @param z z좌표
77      */
78     public Point3D(int x, int y, int z) {
79         // 부모의 생성자 호출
80         super(x, y);
81         this.z = z;
82     }
83
84     /**
85      * 점의 x좌표, y좌표, z좌표를 이동시킵니다. (= setter)
86      * @param x 새 x좌표
87      * @param y 새 y좌표
88      * @param z 새 z좌표
89      */
90     public void move(int x, int y, int z) { super.move(x, y); setZ(z); }
91
92     /**
93      * z좌표값을 1 증가시킵니다.
94      */
95     public void moveUp() { this.setZ(this.getZ()+1); }
96     /**
97      * z좌표값을 1 감소시킵니다.
98      */
99     public void moveDown() { this.setZ(this.getZ()-1); }
100    /**

```

```

101     * z좌표를 반환합니다.
102     * @return {@code int} 점의 z좌표
103     */
104     public int getZ() { return this.z; }
105     /**
106     * {@code setter} z좌표를 설정합니다.
107     * @param z {@code int} 새 z좌표
108     */
109     public void setZ(int z) { this.z = z; }
110
111     /**
112     * 객체의 정보를 String형으로 변환해 반환합니다.
113     * @return {@code String} 객체의 정보
114     */
115     public String toString() { return String.format("(%d, %d, %d)의 점", this.getX(), this.getY(), this.getZ()); }
116 }

```

13.2 실행 예제

13.2.1 예제 1

```

java Prob13.java
(1, 2, 3)의 점입니다.
(1, 2, 4)의 점입니다.
(10, 10, 3)의 점입니다.
(100, 200, 300)의 점입니다.

```

```

[shapelayer@ShapeLayer-MacBookAir java % java Prob13.java
(1, 2, 3)의 점입니다.
(1, 2, 4)의 점입니다.
(10, 10, 3)의 점입니다.
(100, 200, 300)의 점입니다.

```