

학과: 소프트웨어학부 컴퓨터과학전공

학번: 2312282

이름: 임다희

<실습 1> CountryTest

-코드 및 실행결과

```
1 1. *작성자: 임다희(2312282)
2 2. *작성일: 2024-11-21
3 3. *Lab 10-1 HashMap 활용 예제*/
4
5 import java.util.*; //자료구조 사용을 위한 패키지 import
6
7 public class ContryTest {
8
9     public static void main(String[] args) {
10
11         Scanner sc = new Scanner(System.in);
12         Map<String, String> map = new HashMap<String, String>();
13         // 국가 이름을 키, 수도를 값으로 하여 저장하는 해쉬맵을 생성한다.
14
15         map.put("USA", "Washington");
16         map.put("Japan", "Tokyo");
17         map.put("China", "Beijing");
18         map.put("UK", "London");
19         map.put("Korea", "Seoul");
20         // 해쉬맵에 5가지 데이터를 입력한다.
21
22         Iterator<String> it = map.keySet().iterator();
23         // 맵의 모든 요소를 방문하기 위해 반복자를 사용한다.
24
25         while (it.hasNext()) {
26             String key = it.next();
27             System.out.println("Key: " + key + ", Value: " + map.get(key));
28             // 맵의 모든 요소를 방문하며 각각의 key값과 key에 대응하는 value값을 출력한다.
29         }
30
31         System.out.print("국가 이름을 입력하십시오: ");
32         String name = sc.next();
33         // 사용자에게 key값(국가 이름)을 입력받는다.
34         System.out.println(name + "의 수도: " + map.get(name));
35         // 사용자가 입력한 key값에 대응하는 value 값(수도)을 찾아 출력한다.
36     }
37 }
```

```
1 1. *작성자: 임다희(2312282)
2 2. *작성일: 2024-11-21
3 3. *Lab 10-1 HashMap 활용 예제*/
4
5 Console Problems
6 <terminated> ContryTest [Java Application] C:\W
7 Key: USA, Value: Washington
8 Key: Japan, Value: Tokyo
9 Key: China, Value: Beijing
10 Key: UK, Value: London
11 Key: Korea, Value: Seoul
12 국가 이름을 입력하십시오: UK
13 UK의 수도: London
```

<실습 2> LottoTest

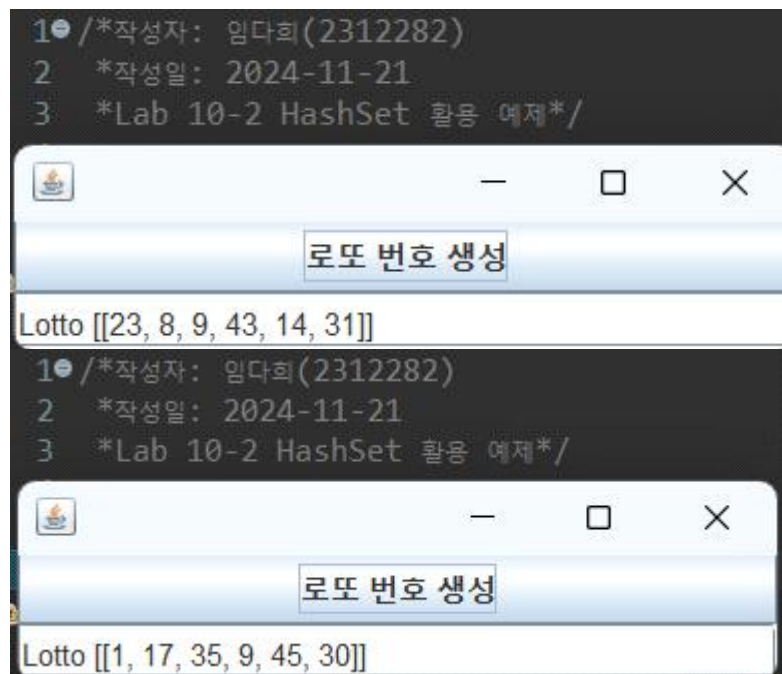
-코드 및 실행결과

```
10 /*작성자: 임다희(2312282)
11  *작성일: 2024-11-21
12  *Lab 10-2 HashSet 활용 예제*/
13
14
15 import java.util.*;
16 import java.util.List;
17 import java.awt.*;
18 import javax.swing.*;
19 //자료구조 사용, GUI 및 이벤트 처리를 위한 패키지 import
20
21 class Lotto {
22     HashSet<Integer> set = new HashSet<Integer>();
23     // 중복 없이 로또 번호 숫자 6개를 저장하는 HashSet을 만든다.
24
25     public void setNumbers() {
26
27         set.clear(); //가존에 저장되어있던 로또 번호를 초기화한다.
28         int len = 0; //확정된 로또 번호 개수를 저장하는 변수 len.
29
30         while (len < 6) { //len이 6 미만일 때(별한 로또 번호가 6개 미만일 때) 다음 내용을 실행한다.
31             int num = (int) (Math.random() * 45 + 1);
32             //1~45의 난수를 생성한다.
33             if (!set.contains(num)) {
34                 //set이 방금 생성된 난수를 포함하고 있는지 확인한다.
35                 //포함하지 않을 경우 난수를 set에 추가하고,
36                 //len에 1을 더해 로또 번호가 하나 추가되었음을 표시한다.
37                 set.add(num);
38                 len++;
39             } //set이 생성한 난수를 포함할 경우에는 해당 난수가 set에 추가되지 않고,
40             //별한 로또번호의 개수도 증가하지 않는다.
41         }
42     }
43
44     // Iterator, List 사용하고 (" ")로 연결해서 로또번호 문자열로 반환?
45     public String setStringNumbers() {
46         Iterator<Integer> it = set.iterator();
47         // set의 모든 요소를 방문하기 위해 반복자를 사용한다.
48         List<String> numList = new ArrayList();
49         //set에 저장된 Integer 형태의 로또 번호를 String 문자열로 바꾸어 저장하는 리스트를 만든다.
50
51         while (it.hasNext()) {
52             Integer num = it.next();
53             numList.add(Integer.toString(num));
54             //set의 모든 요소를 방문하며 Integer 형태의 원소를 String 문자열로 바꾸어 numList에 추가한다.
55         }
56         return ("Lotto [" + numList + "]");
57         //numList를 리턴하여 " "로 연결된 하나의 문자열 형태로 로또 번호를 나타낸다.
58     }
59 }
60 }
```

```

51 public class LottoTest extends JFrame {
52     JTextField tf; //로또 번호를 표시할 텍스트 필드
53     JButton btn; //누르면 로또 번호 추첨이 실행되는 버튼
54
55     public LottoTest() {
56         Lotto lotto = new Lotto(); //Lotto 클래스의 객체 lotto를 생성한다.
57
58         btn = new JButton("로또 번호 생성");
59         btn.addActionListener(e -> {
60             //버튼이 눌리는 이벤트가 발생하면 다음과 같이 실행된다.
61             lotto.setNumbers(); //로또 번호 생성 메소드를 호출한다.
62             tf.setText(lotto.setStringNumbers()); //결과 문자열을 tf에 설정한다.
63         });
64
65         tf = new JTextField(30);
66
67         //각 요소를 frame에 추가한다.
68         add(btn, "North");
69         add(tf, "Center");
70         pack();
71
72         setVisible(true);
73         setDefaultCloseOperation(EXIT_ON_CLOSE);
74     }
75
76     public static void main(String[] args) {
77         new LottoTest();
78     }
79 }
80
81 }
82

```



<실습 3> MovieTest

-코드 및 실행결과

```
10 /*작성자: 임다희(2312282)
11  *작성일: 2024-11-21
12  *Lab 10-3 영화 정보 프로그램*/
13
14 import java.util.*; //자료구조 사용을 위한 패키지 import
15
16 class Movie { //영화 클래스를 정의한다.
17
18     private String title, director, genre; //영화의 제목, 감독, 장르를 나타내는 String 필드
19     private int year; //영화의 개봉 연도를 나타내는 int 필드
20
21     public Movie(String title, String director, String genre, int year) {
22         this.title = title;
23         this.director = director;
24         this.genre = genre;
25         this.year = year;
26     } //Movie 클래스의 생성자. 제목, 감독, 장르, 연도 4개의 값을 받아 Movie 객체를 생성한다.
27
28     @Override
29     public String toString() { //Movie 객체의 필드 정보를 문자열로 출력하는 메소드 toString
30         return "[제목: " + title + "/ 감독: " + director + "/ 장르: " + genre + "/ 개봉년도: " + year + "]\n";
31     }
32
33     public String getTitle() { //제목의 getter
34         return title;
35     }
36
37     public void setTitle(String title) { //제목의 setter
38         this.title = title;
39     }
40
41     public String getDirector() { //감독의 getter
42         return director;
43     }
44
45     public void setDirector(String director) { //감독의 setter
46         this.director = director;
47     }
48
49     public String getGenre() { //장르의 getter
50         return genre;
51     }
52
53     public void setGenre(String genre) { //장르의 setter
54         this.genre = genre;
55     }
56
57     public int getYear() { //연도의 getter
58         return year;
59     }
60
61     public void setYear(int year) { //연도의 setter
62         this.year = year;
63     }
64 }
65
66 public class MovieTest {
67     public static void main(String[] args) {
68
69         ArrayList<Movie> list = new ArrayList<Movie>();
70         //영화 객체들을 저장하는 ArrayList
71
72         Scanner sc = new Scanner(System.in);
73         boolean run = true; //while문의 계속 실행되도록 하는 boolean 타입 변수
74         while (run) {
75             System.out.print(
76                 "=====영화 정보 관리===== \n" + "1.입력/2.출력/3.검색/4.수정/5.삭제/6.종료" + "\n" + "메뉴입력>>");
77             int num = sc.nextInt(); //1~6번까지의 메뉴 옵션 번호를 사용자에게 입력받는다.
78
79             switch (num) { //입력받은 번호에 따라 다음과 같이 실행한다.
80
81                 case 1 -> { //1번(영화 정보 입력)의 경우
82                     System.out.print("제목: ");
83                     sc.nextLine();
84                     String title = sc.nextLine();
85                     System.out.print("감독: ");
86                     String director = sc.nextLine();
87                     System.out.print("장르: ");
88                     String genre = sc.nextLine();
89                     System.out.print("연도: ");
90                     int year = sc.nextInt();
91                     //제목, 감독, 장르, 연도 정보를 사용자에게 입력받아 새로운 Movie 객체를 만든다.
92                     Movie mov = new Movie(title, director, genre, year);
93                     list.add(mov); //생성된 Movie 객체를 리스트에 추가한다.
94                 }
95             }
96         }
97     }
98 }
```



```

87     case 2 -> { //2번(영화 정보 출력)의 경우
88         Iterator<Movie> it = list.iterator();
89         // 리스트의 모든 요소를 방문하기 위해 반복자를 사용한다.
90         while (it.hasNext()) {
91             Movie mov = it.next();
92             System.out.println(mov);
93             //리스트의 모든 요소를 차례로 방문하며 출력한다.
94         }
95     }
96
97     case 3 -> { //3번(영화 정보 검색)의 경우
98         System.out.print("검색할 제목 입력: ");
99         sc.nextLine(); //사용자에게 검색할 영화 제목을 입력받는다.
100         String searchTitle = sc.nextLine();
101         Iterator<Movie> it = list.iterator();
102         // 리스트의 모든 요소를 방문하기 위해 반복자를 사용한다.
103         while (it.hasNext()) {
104             Movie mov = it.next();
105             if (mov.getTitle().equals(searchTitle)) {
106                 System.out.println(mov);
107                 //리스트의 요소 중 제목이 입력받은 제목과 동일한 요소가 있으면
108                 //해당 Movie 객체의 정보를 출력한다.
109             }
110         }
111     }
112
113     case 4 -> { //4번(영화 정보 수정)의 경우
114         System.out.print("수정할 제목 입력: ");
115         sc.nextLine(); //사용자에게 수정할 영화 제목을 입력받는다.
116         String Title = sc.nextLine();
117         Iterator<Movie> it = list.iterator();
118         // 리스트의 모든 요소를 방문하기 위해 반복자를 사용한다.
119         while (it.hasNext()) {
120             Movie mov = it.next();
121
122             if (mov.getTitle().equals(Title)) {
123                 //리스트 요소 중 제목이 입력받은 제목과 동일한 요소가 있으면 해당 Movie 객체의 정보를 출력한다.
124                 System.out.println(mov);
125
126                 //영화의 제목, 감독, 장르, 개봉 연도 값을 다시 입력받는다.
127                 //입력받은 특정 필드 값이 Movie 객체가 기존에 가지고 있었던 값과 다를 경우
128                 //setter를 통해 해당 필드 값을 재지정한다.
129                 System.out.print("제목: ");
130                 String title = sc.nextLine();
131                 if (mov.getTitle() != title)
132                     mov.setTitle(title);
133
134                 System.out.print("감독: ");
135                 String director = sc.nextLine();
136                 if (mov.getDirector() != director)
137                     mov.setDirector(director);
138
139                 System.out.print("장르: ");
140                 String genre = sc.nextLine();
141                 if (mov.getGenre() != genre)
142                     mov.setGenre(genre);
143
144                 System.out.print("연도: ");
145                 int year = sc.nextInt();
146                 if (mov.getYear() != year)
147                     mov.setYear(year);
148
149                 System.out.println(mov + "로 수정되었습니다.");
150                 //수정된 Movie 객체의 필드 정보를 출력한다.
151             }
152         }
153     }
154
155     case 5 -> { //5번(영화 삭제)의 경우
156         System.out.print("삭제할 제목 입력: ");
157         sc.nextLine(); //사용자에게 삭제할 영화 제목을 입력받는다.
158         String Title = sc.nextLine();
159         Iterator<Movie> it = list.iterator();
160         // 리스트의 모든 요소를 방문하기 위해 반복자를 사용한다.
161         while (it.hasNext()) {
162             Movie mov = it.next();
163             if (mov.getTitle().equals(Title)) {
164                 //리스트 요소 중 제목이 입력받은 제목과 동일한 요소가 있으면 해당 Movie 객체를 리스트에서 삭제한다.
165                 int i = list.indexOf(mov);
166                 list.remove(i);
167                 System.out.println(mov + "이 삭제되었습니다.");
168                 break; //삭제가 이루어지면 리스트 요소 방문을 멈춘다.
169                 //삭제한 Movie 객체 정보를 출력한다.
170             }
171         }
172     }

```

```

173         case 6 -> { //6번(프로그램 종료)의 경우
174             System.out.println("프로그램을 종료합니다."); //프로그램 종료 메시지를 출력한다.
175             run = false; // true일 때 while문이 계속 실행되도록 하는 변수 run 값을 false로 바꾼다.
176         }
177     }
178 }
179 }
180 }
181 }
182

```

- 1. 영화 정보 입력

```

1  /*작성자: 임다희(2312282)
2   *작성일: 2024-11-21
3   *Lab 10-3 영화 정보 프로그램*/
4

```

```

=====영화 정보 관리=====
1.입력/2.출력/3.검색/4.수정/5.삭제/6.종료
메뉴입력>>1
제목: 007 노 타임 투 다이
감독: 캐리 후쿠나가
장르: 액션
년도: 2021
=====영화 정보 관리=====
1.입력/2.출력/3.검색/4.수정/5.삭제/6.종료
메뉴입력>>1
제목: 보이스
감독: 김선, 김곡
장르: 범죄, 액션
년도: 2021
=====영화 정보 관리=====
1.입력/2.출력/3.검색/4.수정/5.삭제/6.종료
메뉴입력>>1
제목: 겨울왕국 2
감독: 크리스박
장르: 애니메이션, 판타지, 뮤지컬
년도: 2019
=====영화 정보 관리=====
1.입력/2.출력/3.검색/4.수정/5.삭제/6.종료
메뉴입력>>

```

- 2. 영화 정보 입력, 3. 영화 정보 검색

```
=====영화 정보 관리=====
1.입력/2.출력/3.검색/4.수정/5.삭제/6.종료
메뉴입력>>2
[제목: 007 노 타임 투 다이/ 감독: 캐리 후쿠나가/ 장르: 액션/ 개봉년도: 2021]
[제목: 보이스/ 감독: 김선, 김곡/ 장르: 범죄, 액션/ 개봉년도: 2021]
[제목: 겨울왕국 2/ 감독: 크리스박/ 장르: 애니메이션, 판타지, 뮤지컬/ 개봉년도: 2019]
=====영화 정보 관리=====
1.입력/2.출력/3.검색/4.수정/5.삭제/6.종료
메뉴입력>>3
검색할 제목 입력: 보이스
[제목: 보이스/ 감독: 김선, 김곡/ 장르: 범죄, 액션/ 개봉년도: 2021]
=====영화 정보 관리=====
1.입력/2.출력/3.검색/4.수정/5.삭제/6.종료
메뉴입력>>
```

- 4. 영화 정보 수정/5. 영화 삭제/6.종료

```
=====영화 정보 관리=====
1.입력/2.출력/3.검색/4.수정/5.삭제/6.종료
메뉴입력>>4
수정할 제목 입력: 겨울왕국 2
[제목: 겨울왕국 2/ 감독: 크리스박/ 장르: 애니메이션, 판타지, 뮤지컬/ 개봉년도: 2019]
제목: 겨울왕국2
감독: 크리스 박
장르: 애니메이션, 가족, 판타지, 뮤지컬
년도: 2019
[제목: 겨울왕국2/ 감독: 크리스 박/ 장르: 애니메이션, 가족, 판타지, 뮤지컬/ 개봉년도: 2019]로 수정되었습니다.
=====영화 정보 관리=====
1.입력/2.출력/3.검색/4.수정/5.삭제/6.종료
메뉴입력>>5
삭제할 제목 입력: 겨울왕국2
[제목: 겨울왕국2/ 감독: 크리스 박/ 장르: 애니메이션, 가족, 판타지, 뮤지컬/ 개봉년도: 2019]이 삭제되었습니다.
=====영화 정보 관리=====
1.입력/2.출력/3.검색/4.수정/5.삭제/6.종료
메뉴입력>>6
프로그램을 종료합니다.
```