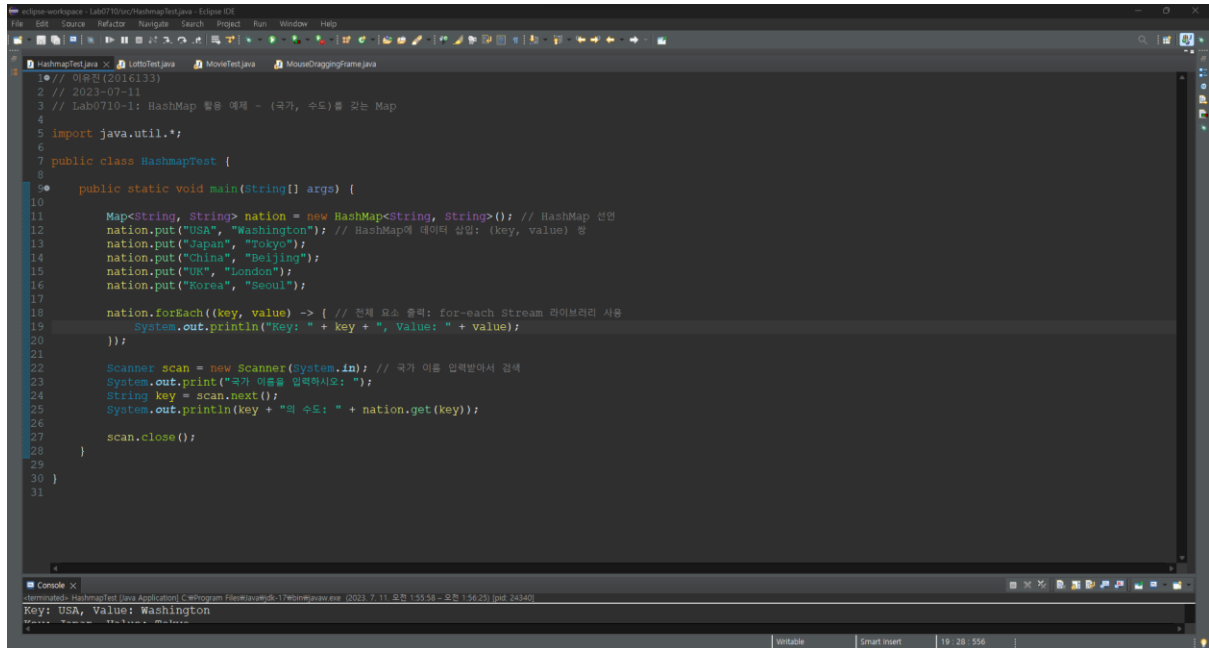


23-여름학기 자바프로그래밍(001) Lab0710
소프트웨어학부 컴퓨터과학전공 2016133 이유진

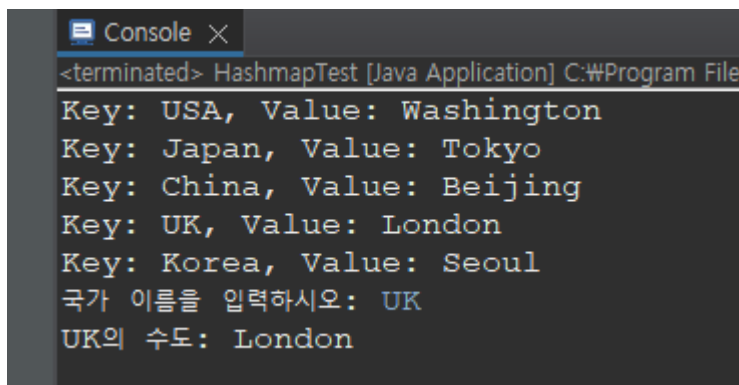
1. HashmapTest.java



```
1 // 이유진 (2016133)
2 // 2023-07-11
3 // Lab0710-1: HashMap 활용 예제 - (국가, 수도)를 갖는 Map
4
5 import java.util.*;
6
7 public class HashmapTest {
8
9     public static void main(String[] args) {
10
11         Map<String, String> nation = new HashMap<String, String>(); // HashMap 선언
12         nation.put("USA", "Washington"); // HashMap에 데이터 삽입: (key, value) 형
13         nation.put("Japan", "Tokyo");
14         nation.put("China", "Beijing");
15         nation.put("UK", "London");
16         nation.put("Korea", "Seoul");
17
18         nation.forEach((key, value) -> { // 전체 요소 출력: for-each Stream 라이브러리 사용
19             System.out.println("Key: " + key + ", Value: " + value);
20         });
21
22         Scanner scan = new Scanner(System.in); // 국가 이름 입력받아서 검색
23         System.out.print("국가 이름을 입력하시오: ");
24         String key = scan.next();
25         System.out.println(key + "의 수도: " + nation.get(key));
26
27         scan.close();
28     }
29 }
30
31
```

Console

```
<terminated> HashmapTest [Java Application] C:\Program Files\Java\jdk-17\bin\java.exe (2023. 7. 11. 오전 1:55:58 - 오전 1:56:25) [pid: 24340]
Key: USA, Value: Washington
Key: Japan, Value: Tokyo
Key: China, Value: Beijing
Key: UK, Value: London
Key: Korea, Value: Seoul
국가 이름을 입력하시오: UK
UK의 수도: London
```



```
Console X
<terminated> HashmapTest [Java Application] C:\Program Files\Java\jdk-17\bin\java.exe (2023. 7. 11. 오전 1:55:58 - 오전 1:56:25) [pid: 24340]
Key: USA, Value: Washington
Key: Japan, Value: Tokyo
Key: China, Value: Beijing
Key: UK, Value: London
Key: Korea, Value: Seoul
국가 이름을 입력하시오: UK
UK의 수도: London
```

2. LottoTest.java

```
1 // 이유진 (2016133)
2 // 2023-07-11
3 // Lab0710-2: HashSet 활용 - 로또 번호 뽑기
4
5 import java.util.*;
6 import javax.swing.*;
7 import java.awt.*;
8
9 class Lotto {
10
11     HashSet<Integer> set = new HashSet<Integer>(); //HashSet 선언
12
13     public void make() { // 로또 번호 생성 메소드
14         while (set.size() < 6) { // set의 수자 6개까 차 때까지 반복한다
15             int n = (int) (Math.random() * 45 + 1); // 1-45 정수
16             set.add(n);
17         }
18     }
19
20     @Override
21     public String toString() { // 로또 번호 결과 문자열로 변환
22         return "Lotto [" + set + "]";
23     }
24 }
25
26 public class LottoTest extends JFrame {
27
28     public LottoTest() {
29
30         Lotto lotto = new Lotto();
31
32         JButton btn = new JButton("로또 번호 생성");
33         JTextField tf = new JTextField(30);
34         btn.addActionListener(e -> {
35             lotto.make(); // 버튼이 눌러지면 로또 번호 생성 메서드 호출
36             tf.setText(lotto.toString()); // 결과 문자열을 tf에 설정
37         });
38
39         add(btn, "North"); // 컨테이너에 붙이기
40         add(tf, "Center");
41         pack(); // 화면에 적절히 잘 차게
42
43         setVisible(true);
44         setDefaultCloseOperation(EXIT_ON_CLOSE);
45     }
46
47     public static void main(String[] args) {
48         new LottoTest();
49     }
50 }
51
```



3. MovieTest.java

```
1 // 이윤진 (2016133)
2 // 2023-07-11
3 // Lab0710-3: 컬렉션 활용 - 영화 정보 프로그램
4
5 import java.util.*;
6
7 class Movie { // 영화 클래스
8     private String title, director, genre;
9     private int year;
10
11     // 생성자
12     public Movie(String title, String director, String genre, int year) {
13         super();
14         this.title = title;
15         this.director = director;
16         this.genre = genre;
17         this.year = year;
18     }
19
20     // 접근자, 설정자
21     public String getTitle() {
22         return title;
23     }
24
25     public void setTitle(String title) {
26         this.title = title;
27     }
28
29     public String getDirector() {
30         return director;
31     }
32
33     public void setDirector(String director) {
34         this.director = director;
35     }
36
37     public String getGenre() {
38         return genre;
39     }
40
41     public void setGenre(String genre) {
42         this.genre = genre;
43     }
44
45     public int getYear() {
46         return year;
47     }
48
49     public void setYear(int year) {
50         this.year = year;
51     }
52
53     @Override
54     public String toString() {
55         return "[제목: " + title + " / 감독: " + director + " / 장르: " + genre + " / 개봉년도: " + year + " ]";
56     }
57 }
58
59 public class MovieTest {
60
61     public static void main(String[] args) {
62
63         Scanner scan = new Scanner(System.in);
64         ArrayList<Movie> movieList = new ArrayList<Movie>(); // 영화 컬렉션
65         Movie m;
66         String title, director, genre;
67         int year;
68
69         while (true) {
70             System.out.println("===== 영화 정보 관리 =====");
71             System.out.println("1.입력 / 2.출력 / 3.검색 / 4.수정 / 5.삭제 / 6.종료");
72             System.out.print("메뉴 입력>> ");
73             int menu = scan.nextInt();
74             scan.nextLine();
75
76             switch (menu) {
77                 case 1:
78                     System.out.print("제목: ");
79                     title = scan.nextLine();
80                     System.out.print("감독: ");
81                     director = scan.nextLine();
82                     System.out.print("장르: ");
83                     genre = scan.nextLine();
84                     System.out.print("개봉년도: ");
85                     year = scan.nextInt();
86                     scan.nextLine();
87                     m = new Movie(title, director, genre, year);
88                     movieList.add(m);
89                     System.out.println(m.toString());
90                 case 2:
91                     System.out.println("출력할 영화의 개수: " + movieList.size());
92                     for (Movie m : movieList) {
93                         System.out.println(m.toString());
94                     }
95                 case 3:
96                     System.out.print("검색할 제목: ");
97                     String keyword = scan.nextLine();
98                     ArrayList<Movie> result = new ArrayList<Movie>();
99                     for (Movie m : movieList) {
100                         if (m.getTitle().contains(keyword)) {
101                             result.add(m);
102                         }
103                     }
104                     System.out.println("검색 결과: " + result.size());
105                     for (Movie m : result) {
106                         System.out.println(m.toString());
107                     }
108                 case 4:
109                     System.out.print("수정할 제목: ");
110                     String keyword = scan.nextLine();
111                     ArrayList<Movie> result = new ArrayList<Movie>();
112                     for (Movie m : movieList) {
113                         if (m.getTitle().contains(keyword)) {
114                             result.add(m);
115                         }
116                     }
117                     System.out.println("수정할 영화의 개수: " + result.size());
118                     for (Movie m : result) {
119                         System.out.print("현재 제목: ");
120                         String oldTitle = scan.nextLine();
121                         System.out.print("수정할 제목: ");
122                         String newTitle = scan.nextLine();
123                         m.setTitle(newTitle);
124                         System.out.println(m.toString());
125                     }
126                 case 5:
127                     System.out.print("삭제할 제목: ");
128                     String keyword = scan.nextLine();
129                     ArrayList<Movie> result = new ArrayList<Movie>();
130                     for (Movie m : movieList) {
131                         if (m.getTitle().contains(keyword)) {
132                             result.add(m);
133                         }
134                     }
135                     System.out.println("삭제할 영화의 개수: " + result.size());
136                     for (Movie m : result) {
137                         System.out.println(m.toString());
138                     }
139                     movieList.removeAll(result);
140                     System.out.println("삭제 완료");
141                 case 6:
142                     System.out.println("프로그램 종료");
143                     return;
144             }
145         }
146     }
147 }
```

```
File Edit Source Refactor Navigate Search Project Run Window Help
HashMapTest.java LottoTest.java MovieTest.java MovieCraggingFrame.java
79 title = scan.nextLine();
80 System.out.print("감독: ");
81 director = scan.nextLine();
82 System.out.print("장르: ");
83 genre = scan.nextLine();
84 System.out.print("년도: ");
85 year = scan.nextInt();
86
87 m = new Movie(title, director, genre, year); // Movie 객체 생성
88 movieList.add(m); // 컬렉션에 저장
89 continue;
90
91 case 2:
92 for (int i = 0; i < movieList.size(); i++)
93 System.out.println(movieList.get(i));
94 continue;
95
96 case 3:
97 System.out.print("검색할 제목 입력: ");
98 String search = scan.nextLine();
99
100 for (int i = 0; i < movieList.size(); i++) {
101 if (movieList.get(i).getTitle().equals(search)) // 해당 내용으로 검색된 검색: 있으면 출력
102 System.out.println("검색 결과 " + movieList.get(i));
103 }
104 continue;
105
106 case 4:
107 System.out.print("수정할 제목 입력: ");
108 String modify = scan.nextLine();
109
110 for (int i = 0; i < movieList.size(); i++) {
111 if (movieList.get(i).getTitle().equals(modify)) { // 해당 내용으로 검색된 검색: 있으면 수정
112 System.out.println(movieList.get(i));
113
114 System.out.print("제목: "); // 새로운 정보 받기
115 title = scan.nextLine();
116 System.out.print("장르: ");
117 director = scan.nextLine();
118 System.out.print("년도: ");
119 year = scan.nextInt();
120
121 m = new Movie(title, director, genre, year); // 새 Movie 객체 만들기
122 movieList.set(i, m); // 컬렉션에서 해당 인덱스의 객체 수정
123 System.out.println(movieList.get(i) + "로 수정되었습니다.");
124 }
125 }
126 continue;
127
128 case 5:
129 System.out.print("삭제할 제목 입력: ");
130 String delete = scan.nextLine();
131
132 for (int i = 0; i < movieList.size(); i++) {
133 if (movieList.get(i).getTitle().equals(delete)) // 해당 내용으로 검색된 검색: 있으면 삭제
134 System.out.println(movieList.get(i) + "이 삭제되었습니다.");
135 movieList.remove(i);
136 }
137 continue;
138
139 case 6:
140 System.out.println("프로그램을 종료합니다.");
141 break;
142 }
143 }
144 }
145 }
146 }
147 }
148 }
149 }
150 }
```

```
Console x
C:\Program Files\Java\jdk-17\bin\javaw.exe (2023.7.11. 오전 2:09:24 - 오전 2:10:21) (pid 29876)
===== 영화 정보 관리 =====
1.입력 / 2.출력 / 3.검색 / 4.수정 / 5.삭제 / 6.종료
메뉴 입력>> 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
메뉴 입력>> 1
제목: 보이스
감독: 김선, 김국
장르: 범죄, 액션
년도: 2021
===== 영화 정보 관리 =====
1.입력 / 2.출력 / 3.검색 / 4.수정 / 5.삭제 / 6.종료
메뉴 입력>> 2
[제목: 007 노 타임 투 다이/ 감독: 캐리 후쿠나가/ 장르: 액션/ 개봉년도: 2021 ]
[제목: 보이스/ 감독: 김선, 김국/ 장르: 범죄, 액션/ 개봉년도: 2021 ]
[제목: 겨울왕국 2/ 감독: 크리스토퍼 로글스/ 장르: 애니메이션, 판타지, 뮤지컬/ 개봉년도: 2019 ]
===== 영화 정보 관리 =====
1.입력 / 2.출력 / 3.검색 / 4.수정 / 5.삭제 / 6.종료
메뉴 입력>> 3
검색할 제목 입력: 보이스
검색 결과 [제목: 보이스/ 감독: 김선, 김국/ 장르: 범죄, 액션/ 개봉년도: 2021 ]
===== 영화 정보 관리 =====
1.입력 / 2.출력 / 3.검색 / 4.수정 / 5.삭제 / 6.종료
메뉴 입력>> 4
```

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