



# 1학기 고급자바 실습

## week 5-2

김민진(18)

김지희(18)

문의 메일 : genie02166@duksung.ac.kr

## Part2. 컬렉션 프레임워크

### 1) Map 컬렉션

- HashMap 클래스 1
- HashMap 클래스 2

### 2) 문제 풀어보기

- HashSet 관련 문제
- HashMap 관련 문제

## 실행 결과

```
*HashMapEx.java
1 package week5_2;
2 import java.util.HashMap;
3 import java.util.Iterator;
4 import java.util.Map;
5 import java.util.Set;
6
7 public class HashMapEx {
8     public static void main(String[] args) {
9         //Map 컬렉션 생성
10         Map<String, Integer> map = new HashMap<String, Integer>();
11
12         //객체 저장
13         map.put("신용권", 85);
14         map.put("홍길동", 90);
15         map.put("동장군", 80);
16         map.put("홍길동", 95);
17         System.out.println("총 Entry 수 : " + map.size());
18
19         //객체 찾기
20         System.out.println("\t홍길동 : " + map.get("홍길동"));
21         System.out.println();
22
23         //객체를 하나씩 처리
24         Set<String> keySet = map.keySet();
25         Iterator<String> keyIterator = keySet.iterator();
26         while(keyIterator.hasNext()) {
27             String key = keyIterator.next();
28             Integer value = map.get(key);
29             System.out.println("\t" + key + " : " + value);
30         }
31         System.out.println();
32     }
33 }
```

```
Problems @ Javadoc Dec
<terminated> HashMapEx [Java
총 Entry 수 : 3
    홍길동 : 95

    홍길동 : 95
    신용권 : 85
    동장군 : 80

총 Entry 수 : 2
    신용권 : 85
    동장군 : 80

총 Entry 수 : 0
```

```
33         //객체 삭제
34         map.remove("홍길동");
35         System.out.println("총 Entry 수 : " + map.size());
36
37         //객체를 하나씩 처리
38         Set<Map.Entry<String, Integer>> entrySet = map.entrySet();
39         Iterator<Map.Entry<String, Integer>> entryIterator = entrySet.iterator();
40
41         while(entryIterator.hasNext()) {
42             Map.Entry<String, Integer> entry = entryIterator.next();
43             String key = entry.getKey();
44             Integer value = entry.getValue();
45             System.out.println("\t" + key + " : " + value);
46         }
47         System.out.println();
48
49         //객체 전체 삭제
50         map.clear();
51         System.out.println("총 Entry 수 : " + map.size());
52     }
53 }
```

# 1) Map 컬렉션 - HashMap 1

# 1) Map 컬렉션

## - HashMap 2

실행 결과

Problems @ Javadoc Decla  
<terminated> HashMapEx2 [Java A  
총 Entry 수 : 1

```
HashMapEx.java *Student.java ✕
1 package week5_2;
2
3 public class Student {
4
5     public int sno;
6     public String name;
7
8     public Student(int sno, String name) {
9         this.sno = sno;
10        this.name = name;
11    }
12
13    public boolean equals(Object obj) {
14        if(obj instanceof Student) {
15            Student student = (Student) obj;
16            return (sno==student.sno) && (name.equals(student.name));
17        }else
18            return false;
19    }
20
21    public int hashCode() {
22        return sno+name.hashCode();
23    }
24 }
```

```
HashMapEx.java *Student.java *HashMapEx2.java ✕
1 package week5_2;
2 import java.util.*;
3
4 import java.util.Map;
5
6 public class HashMapEx2 {
7     public static void main(String[] args) {
8         Map<Student, Integer> map = new HashMap<Student, Integer>();
9
10        map.put(new Student(1, "홍길동"), 95);
11        map.put(new Student(1, "홍길동"), 95);
12
13        System.out.println("총 Entry 수 : " + map.size());
14    }
15 }
```

## 2) 문제 풀어보기

### - HashSet

**Q1.** HashSet에 Student 객체를 저장하려고 합니다. 학번이 같으면 동일한 Student라고 가정하고 중복 저장이 되지 않도록 하고 싶습니다. Student 클래스에서 재정의해야 하는 hashCode()와 equals() 메소드의 내용을 채워보세요. Student의 해시코드는 학번이라고 가정합니다.

## 2) 문제 풀어보기

### - HashSet

HashSetExample.java Student.java

```
1 package week5_2;
2
3 import java.util.HashSet;
4 import java.util.Iterator;
5 import java.util.Set;
6
7 public class HashSetExample {
8     public static void main(String[] args) {
9         Set<Student> set = new HashSet<Student>();
10
11         set.add(new Student(1, "홍길동"));
12         set.add(new Student(2, "정원호"));
13         set.add(new Student(1, "김자바"));
14
15         Iterator<Student> iterator = set.iterator();
16         while(iterator.hasNext()) {
17             Student student = iterator.next();
18             System.out.println(student.studentNum + ":" + student.name);
19         }
20     }
21 }
```

HashSetExample.java Student.java

```
1 package week5_2;
2
3 public class Student {
4     public int studentNum;
5     public String name;
6
7     public Student (int studentNum, String name) {
8         this.studentNum = studentNum;
9         this.name = name;
10    }
11
12    @Override
13    public int hashCode() {
14        // 코드 작성
15    }
16
17    @Override
18    public boolean equals(Object obj) {
19        // 코드 작성
20    }
21 }
```

실행 결과

1:홍길동

2:정원호

## 2) 문제 풀어보기

### - HashSet

```
HashSetExample.java Student.java ✖
1 package week5_2;
2
3 public class Student {
4     public int studentNum;
5     public String name;
6
7     public Student (int studentNum, String name) {
8         this.studentNum = studentNum;
9         this.name = name;
10    }
11
12    @Override
13    public int hashCode() {
14        // 코드 작성
15    }
16
17    @Override
18    public boolean equals(Object obj) {
19        // 코드 작성
20    }
21 }
```

return studentNum;

if(!(obj instanceof Student)) return false;  
Student student = (Student) obj;  
if(studentNum != student.studentNum) return false;  
return true;

## 2) 문제 풀어보기

### - HashMap

Q2. HashMap에 아이디(String)와 점수(Integer)가 저장되어 있습니다.

실행 결과와 같이 평균 점수를 출력하고, 최고 점수와 최고 점수를 받은 아이디를 출력해보세요.

```
MapExample.java
1 package week5_2;
2
3 import java.util.HashMap;
4 import java.util.Map;
5 import java.util.Set;
6
7 public class MapExample {
8     public static void main(String[] args) {
9         Map<String,Integer> map = new HashMap<String,Integer>();
10        map.put("blue", 96);
11        map.put("hong", 86);
12        map.put("white", 92);
13
14        String name = null;    // 최고 점수를 받은 아이디 저장
15        int maxScore = 0;      // 최고 점수 저장
16        int totalScore = 0;    // 점수 합계 저장
17
18
19        // 작성 위치
20
21    }
22 }
```

#### 실행 결과

평균점수: 91

최고점수: 96

최고점수를 받은 아이디: blue



## 2) 문제 풀어보기

### - HashMap

```
MapExample.java
1 package week5_2;
2
3 import java.util.HashMap;
4 import java.util.Map;
5 import java.util.Set;
6
7 public class MapExample {
8     public static void main(String[] args) {
9         Map<String,Integer> map = new HashMap<String,Integer>();
10        map.put("blue", 96);
11        map.put("hong", 86);
12        map.put("white", 92);
13
14        String name = null;    // 최고 점수를 받은 아이디 저장
15        int maxScore = 0;      // 최고 점수 저장
16        int totalScore = 0;    // 점수 합계 저장
17
18
19        // 작성 위치
20
21    }
22 }
```

실행 결과

평균점수: 91

최고점수: 96

최고점수를 받은 아이디: blue

```
Set<Map.Entry<String,Integer>> entrySet = map.entrySet();
for(Map.Entry<String,Integer> entry : entrySet) {
    if(entry.getValue() > maxScore) {
        name = entry.getKey();
        maxScore = entry.getValue();
    }
    totalScore += entry.getValue();
}
```

```
int avgScore = totalScore / map.size();
System.out.println("평균점수: " + avgScore);
```

```
System.out.println("최고점수: " + maxScore);
System.out.println("최고점수를 받은 아이디: " + name);
```

## 출석 과제 (4/16 금 오후 11:55 마감)

**Q. Map 컬렉션에 대한 설명 중 틀린 것은 무엇입니까?**

- 1) 대표적인 구현 클래스로는 HashMap, Hashtable, TreeMap, Properties가 있다.
- 2) HashMap과 Hashtable은 hashCode()와 equals()를 이용해서 중복 키를 판별한다.
- 3) 멀티 스레드 환경에서는 Hashtable보다는 HashMap이 스레드에 안전하다.
- 4) Properties는 키와 값이 모두 String 타입이다.