

2024년 상반기 K-디지털 트레이닝

컬렉션 자료구조

[KB] IT's Your Life



다음 Board 클래스에 롬복을 사용하여 기본 데이터 클래스로 작성하세요.

```
package ch15.sec02.exam01;

public class Board {
  private String subject;
  private String content;
  private String writer;
}
```

☑ Board.java

```
package ch15.sec02.exam01;

@Data
@NoArgsConstructor
@AllArgsConstructor
public class Board {
   private String subject;
   private String content;
   private String writer;
}
```

☑ 다음 조건을 만족하는 프로그램을 작성하세요.

- o Board 클래스를 저장하기위한 ArrayList 인스턴스 생성
- o Board 클래스의 인스턴스 5개를 생성하여 ArrayList에 추가
- o ArrayList에 저장된 요소의 개수 출력
- o 3번째 데이터를 추출하여 출력
- o 일반 for 문으로 순회하면서 리스트의 각 요소를 출력
- o 인덱스 2에 해당하는 요소 제거
- o 인덱스 2에 해당하는 요소 제거
- o 향상된 for 문으로 순회하면서 리스트의 각 요소를 출력

```
package ch15.sec02.exam01;

public class ArrayListExample {
   public static void main(String[] args) {
   }
}
```

```
package ch15.sec02.exam01;
import java.util.ArrayList;
import java.util.List;
public class ArrayListExample {
 public static void main(String[] args) {
   //ArrayList 컬렉션 생성
   List<Board> list = new ArrayList< >();
   //객체 추가
   list.add(new Board("제목1", "내용1", "글쓴이1"));
   list.add(new Board("제목2", "내용2", "글쓴이2"));
   list.add(new Board("제목3", "내용3", "글쓴이3"));
   list.add(new Board("제목4", "내용4", "글쓴이4"));
   list.add(new Board("제목5", "내용5", "글쓴이5"));
   //저장된 총 객체 수 얻기
   int size = list.size();
   System.out.println("총 객체 수: " + size);
   System.out.println();
```

```
//특정 인덱스의 객체 가져오기
Board board = list.get(2);
System.out.println(board.getSubject() + "\t" + board.getContent() +
          "\t" + board.getWriter());
System.out.println();
//모든 객체를 하나씩 가져오기
for(int i=0; i<list.size(); i++) {</pre>
 Board b = list.get(i);
 System.out.println(b.getSubject() + "\t" + b.getContent() +
          "\t" + b.getWriter());
System.out.println();
//객체 삭제
list.remove(2);
list.remove(2);
```

```
//향상된 for문으로 모든 객체를 하나씩 가져오기
for(Board b : list) {
 System.out.println(b.getSubject() + "\t" + b.getContent() +
       "\t" + b.getWriter());
    총 객체 수: 5
           내용3
    제목3
                 글쓴이3
    제목1
           내용1
                 글쓴이1
    제목2
           내용2
                글쓴이2
    제목3
           내용3
               글쓴이3
    제목4
          내용4 글쓴이4
    제목5
           내용5
                 글쓴이5
    제목1
           내용1
                 글쓴이1
    제목2
           내용2
                글쓴이2
    제목5
           내용5
                 글쓴이5
```

♥ 앞의 조건을 만족하는 프로그램을 작성하세요. 단 리스트는 LinkedList를 사용한다.

```
package ch15.sec02.exam03;

public class LinkedListExample {
   public static void main(String[] args) {
   }
}
```

```
package ch15.sec02.exam03;

import java.util.LinkedList;
import java.util.List;

public class LinkedListExample {
  public static void main(String[] args) {
    //ArrayList 컬렉션 생성
    List<Board> list = new LinkedList < >();

  // 나머지는 모두 동일
  }
}
```

◎ 다음 프로그램의 결과를 적어보고, 실제 실행을 통해 결과를 확인하세요.

```
package ch15.sec03.exam01;
import java.util.*;
public class HashSetExample {
 public static void main(String[] args) {
   Set<String> set = new HashSet<String>();
   //객체 저장
   set.add("Java");
   set.add("JDBC");
   set.add("Servlet/JSP");
   set.add("Java");
   set.add("iBATIS");
   //저장된 객체 수 출력
   int size = set.size();
   System.out.println("총 객체 수: " + size);
                     총 객체 수: 4
```

○ 다음 Member 클래스를 Set에 저장하기위한 클래스로 수정하세요.

```
package ch15.sec03.exam02;

public class Member {
   public String name;
   public int age;
}
```

Member.java

```
package ch15.sec03.exam02;

@Data
@NoArgsConstructor
@AllArgsConstructor
public class Member {
   public String name;
   public int age;
}
```

○ 앞에서 작성한 Member 클래스를 다음과 같이 운영하고, 올바르게 동작하는지 확인하세요.

```
package ch15.sec03.exam02;
import java.util.*;
public class HashSetExample {
 public static void main(String[] args) {
   //HashSet 컬렉션 생성
   Set<Member> set = new HashSet<Member>();
   //Member 객체 저장
   set.add(new Member("홍길동", 30));
   set.add(new Member("홍길동", 30));
   //저장된 객체 수 출력
   System.out.println("총 객체 수 : " + set.size());
```

총 객체 수: 1

♡ 아래 조건(주석으로 표현)을 만족하도록 프로그램을 완성하세요.

```
package ch15.sec03.exam03;
import java.util.*;
public class HashSetExample {
 public static void main(String[] args) {
   //HashSet 컬렉션 생성
   Set<String> set = new HashSet<String>();
   //객체 추가
   set.add("Java");
   set.add("JDBC");
   set.add("JSP");
   set.add("Spring");
  // Iterator 패턴으로 순회하며 각 요소를 출력하세요.
  // 향상된 for 문으로 순회하며 각 요소를 출력하세요.
```

HashSetExample.java

```
package ch15.sec03.exam03;
import java.util.*;
public class HashSetExample {
 public static void main(String[] args) {
   //HashSet 컬렉션 생성
   Set<String> set = new HashSet<String>();
   //객체 추가
   set.add("Java");
   set.add("JDBC");
   set.add("JSP");
   set.add("Spring");
```

HashSetExample.java

```
//객체를 하나씩 가져와서 처리
Iterator<String> iterator = set.iterator();
while(iterator.hasNext()) {
 //객체를 하나 가져오기
 String element = iterator.next();
 System.out.println( element);
 if(element.equals("JSP")) {
   //가져온 객체를 컬렉션에서 제거
   iterator.remove();
System.out.println();
                                          Java
//객체 제거
                                          JSP
set.remove("JDBC");
                                          JDBC
                                          Spring
//객체를 하나씩 가져와서 처리
for(String element : set) {
                                          Java
 System.out.println(element);
                                          Spring
```

☑ 다음 프로그램을 완성하세요.

```
package ch15.sec04.exam01;
import java.util.HashMap;
import java.util.Iterator;
import java.util.Map;
import java.util.Map.Entry;
import java.util.Set;
public class HashMapExample {
 public static void main(String[] args) {
   //Map 컬렉션 생성
                        map =
   //객체 저장
   map.put("신용권", 85);
   map.put("홍길동", 90);
   map.put("동장군", 80);
   map.put("홍길동", 95);
   System.out.println("총 Entry 수: " + _____);
   System.out.println();
```

💟 다음 프로그램을 완성하세요.

```
//키로 값 얻기
String key = "홍길동";
int value = _____
System.out.println(key + ": " + value);
System.out.println();
//키 Set 컬렉션을 얻고, 반복해서 키와 값을 얻기
Set<String> keySet = _____
Iterator<String> keyIterator = keySet.iterator();
while (keyIterator.hasNext()) {
 String k = keyIterator.next();
 Integer v = _____
 System.out.println(k + " : " + v);
System.out.println();
```

💟 다음 프로그램을 완성하세요.

```
//엔트리 Set 컬렉션을 얻고, 반복해서 키와 값을 얻기
                                 entrySet = map.entrySet();
                                 entryIterator = entrySet.iterator();
while (entryIterator.hasNext()) {
 Entry<String, Integer> entry = entryIterator.next();
 String k = entry.getKey();
 Integer v = entry.getValue();
 System.out.println(k + " : " + v);
System.out.println();
//키로 엔트리 삭제
                 ("홍길동");
System.out.println("총 Entry 수: " + _____
System.out.println();
```

HashMapExample.java

```
package ch15.sec04.exam01;
import java.util.HashMap;
import java.util.Iterator;
import java.util.Map;
import java.util.Map.Entry;
import java.util.Set;
public class HashMapExample {
 public static void main(String[] args) {
   //Map 컬렉션 생성
   Map<String, Integer> map = new HashMap< >();
   //객체 저장
   map.put("신용권", 85);
   map.put("홍길동", 90);
   map.put("동장군", 80);
   map.put("홍길동", 95);
   System.out.println("총 Entry 수: " + map.size());
   System.out.println();
                                         총 Entry 수: 3
```

HashMapExample.java

```
//키로 값 얻기
String key = "홍길동";
int value = map.get(key);
System.out.println(key + ": " + value);
System.out.println();
//키 Set 컬렉션을 얻고, 반복해서 키와 값을 얻기
Set<String> keySet = map.keySet();
Iterator<String> keyIterator = keySet.iterator();
while (keyIterator.hasNext()) {
 String k = keyIterator.next();
 Integer v = map.get(k);
 System.out.println(k + " : " + v);
                                            홍길동: 95
System.out.println();
                                           홍길동 : 95
                                            신용권: 85
                                            동장군:80
```

HashMapExample.java

```
//엔트리 Set 컬렉션을 얻고, 반복해서 키와 값을 얻기
Set<Entry<String, Integer>> entrySet = map.entrySet();
Iterator<Entry<String, Integer>> entryIterator = entrySet.iterator();
while (entryIterator.hasNext()) {
 Entry<String, Integer> entry = entryIterator.next();
 String k = entry.getKey();
 Integer v = entry.getValue();
 System.out.println(k + " : " + v);
System.out.println();
//키로 엔트리 삭제
map.remove("홍길동");
System.out.println("총 Entry 수: " + map.size());
System.out.println();
                                            홍길동 : 95
                                            신용권: 85
                                            동장군:80
                                            총 Entry 수: 2
```

- ♥ 롬복을 이용하여 다음 클래스를 완성하세요.
 - o 전체 매개변수 생성자
 - o value에 대한 Getter

```
package ch15.sec06.exam01;

public class Coin {
   private int value;
}
```

Coin.java

```
package ch15.sec06.exam01;

public class Coin {
  private int value;

public Coin(int value) {
    this.value = value;
  }

public int getValue() {
    return value;
  }
}
```

- 스택을 이용하여 다음 순서대로 추가하고, 스택이 빌 때까지 하나씩 뽑으면서 해당 요소를 출력하세요.
 - o 추가 순서 Coin 100, 50, 500, 10

```
package ch15.sec06.exam01;

public class StackExample {
   public static void main(String[] args) {
   }
}
```

```
꺼내온 동전 : 10원
꺼내온 동전 : 500원
꺼내온 동전 : 50원
꺼내온 동전 : 100원
```

StackExample.java

```
package ch15.sec06.exam01;
import java.util.Stack;
public class StackExample {
 public static void main(String[] args) {
   //Stack 컬렉션 생성
   Stack<Coin> coinBox = new Stack<Coin>();
   //동전 넣기
   coinBox.push(new Coin(100));
   coinBox.push(new Coin(50));
   coinBox.push(new Coin(500));
                                                  꺼내온 동전: 10원
   coinBox.push(new Coin(10));
                                                  꺼내온 동전: 500원
                                                  꺼내온 동전: 50원
   //동전을 하나씩 꺼내기
                                                  꺼내온 동전: 100원
   while(!coinBox.isEmpty()) {
    Coin coin = coinBox.pop();
    System.out.println("꺼내온 동전 : " + coin.getValue() + "원");
```

- ♥ 롬복을 이용하여 다음 클래스를 완성하세요.
 - o 전체 매개변수 생성자

```
package ch15.sec06.exam02;

public class Message {
   public String command;
   public String to;
}
```

Message.java

```
package ch15.sec06.exam02;

@AllArgsConstructor
public class Message {
   public String command;
   public String to;
}
```

- ☑ 다음 조건을 만족하는 프로그램을 작성하세요.
 - o LinkedList를 큐로 이용
 - o Message 인스턴스를 다음 순(command, to)으로 추가
 - sendMail, 홍길동
 - sendSMS, 신용권
 - sendKakaotalk, 감자바
 - o 큐가 빌 때까지 큐에서 Message를 추출하며 command에 따라 다음과 같이 출력

```
package ch15.sec06.exam02;

import java.util.LinkedList;
import java.util.Queue;

public class QueueExample {
  public static void main(String[] args) {

  }

}

홍길동님에게 메일을 보냅니다.
  신용권님에게 SMS를 보냅니다.
  감자바님에게 카카오톡를 보냅니다.
```

QueueExample.java

```
package ch15.sec06.exam02;
import java.util.LinkedList;
import java.util.Queue;
public class QueueExample {
 public static void main(String[] args) {
   //Queue 컬렉션 생성
   Queue<Message> messageQueue = new LinkedList<>();
   //메시지 넣기
   messageQueue.offer(new Message("sendMail", "홍길동"));
   messageQueue.offer(new Message("sendSMS", "신용권"));
   messageQueue.offer(new Message("sendKakaotalk", "감자바"));
```

QueueExample.java

```
//메시지를 하나씩 꺼내어 처리
while(!messageQueue.isEmpty()) {
 Message message = messageQueue.poll();
 switch(message.command) {
   case "sendMail":
    System.out.println(message.to + "님에게 메일을 보냅니다.");
    break;
   case "sendSMS":
    System.out.println(message.to + "님에게 SMS를 보냅니다.");
    break;
   case "sendKakaotalk":
    System.out.println(message.to + "님에게 카카오톡를 보냅니다.");
    break;
```

```
홍길동님에게 메일을 보냅니다.
신용권님에게 SMS를 보냅니다.
감자바님에게 카카오톡를 보냅니다.
```