

문의 메일 : genie02166@duksung.ac.kr

Part1. 컬렉션 프레임워크

1) List 컬렉션

- ArrayList 클래스
- Arrays.asList() 메소드
- Vector 클래스
- LinkedList 클래스

2) Set 컬렉션

- HashSet 클래스 1
- HashSet 클래스 2

3) Queue

- Queue 인터페이스의 메소드
- Queue를 이용한 메시지 큐

1) List 컬렉션

- ArrayList

ArrayListExample.java

```
1 package week5;
2 import java.util.*;
3
4 public class ArrayListExample {
5     public static void main(String[] args) {
6         List<String> list = new ArrayList<String>();
7         list.add("Java");
8         list.add("JDBC");
9         list.add("Servlet/JSP");
10        list.add(2, "Database");
11        list.add("iBATIS");
12
13        int size = list.size();
14        System.out.println("총 객체수: " + size);
15        System.out.println();
16
17        String skill = list.get(2);
18        System.out.println("2: " + skill);
19        System.out.println();
20
21        for(int i=0; i<list.size(); i++) {
22            String str = list.get(i);
23            System.out.println(i + ":" + str);
24        }
25        System.out.println();
26
27        list.remove(2);
28        list.remove(2);
29        list.remove("iBATIS");
30
31        for(int i=0; i<list.size(); i++) {
32            String str = list.get(i);
33            System.out.println(i + ":" + str);
34        }
35    }
36 }
```

실행 결과

Console Problem

<terminated> ArrayList

총 객체수: 5

2: Database

0: Java

1: JDBC

2: Database

3: Servlet/JSP

4: iBATIS

0: Java

1: JDBC

1) List 컬렉션

- Arrays.asList()

ArraysAsListExample.java

```
1 package week5;
2
3 import java.util.Arrays;
4
5
6 public class ArraysAsListExample {
7     public static void main(String[] args) {
8         List<String> list1 = Arrays.asList("홍길동", "정원호", "김덕성");
9         for(String name: list1) {
10             System.out.println(name);
11         }
12
13         List<Integer> list2 = Arrays.asList(1, 2, 3);
14         for(int value : list2) {
15             System.out.println(value);
16         }
17     }
18 }
```

실행 결과

Console

<terminated> An

홍길동
정원호
김덕성
1
2
3

1) List 컬렉션 - Vector

Board.java VectorExample.java

```
1 package week5;
2
3 public class Board {
4
5     String subject;
6     String content;
7     String writer;
8
9     public Board(String subject, String content, String writer) {
10         this.subject = subject;
11         this.content = content;
12         this.writer = writer;
13     }
14 }
```

실행 결과

Console Problems

<terminated> VectorExample [

제목1	내용1	글쓴이1
제목2	내용2	글쓴이2
제목4	내용4	글쓴이4

Board.java VectorExample.java

```
1 package week5;
2
3 import java.util.*;
4
5 public class VectorExample {
6     public static void main(String[] args) {
7         List<Board> list = new Vector<Board>();
8
9         list.add(new Board("제목1", "내용1", "글쓴이1"));
10        list.add(new Board("제목2", "내용2", "글쓴이2"));
11        list.add(new Board("제목3", "내용3", "글쓴이3"));
12        list.add(new Board("제목4", "내용4", "글쓴이4"));
13        list.add(new Board("제목5", "내용5", "글쓴이5"));
14
15        list.remove(2);
16        list.remove(3);
17
18        for(int i=0; i<list.size(); i++) {
19            Board board = list.get(i);
20            System.out.println(board.subject + "\t" + board.content + "\t" + board.writer);
21        }
22    }
23 }
```

1) List 컬렉션

- LinkedList

LinkedListExample.java

```
1 package week5;
2
3 import java.util.*;
4
5 public class LinkedListExample {
6     public static void main(String[] args) {
7         List<String> list1 = new ArrayList<String>();
8         List<String> list2 = new LinkedList<String>();
9
10        long startTime;
11        long endTime;
12
13        startTime = System.nanoTime();
14        for(int i=0; i<10000; i++) {
15            list1.add(0, String.valueOf(i));
16        }
17        endTime = System.nanoTime();
18        System.out.println("ArrayList 걸린시간: " + (endTime-startTime) + " ns");
19
20        startTime = System.nanoTime();
21        for(int i=0; i<10000; i++) {
22            list2.add(0, String.valueOf(i));
23        }
24        endTime = System.nanoTime();
25        System.out.println("LinkedList 걸린시간: " + (endTime-startTime) + " ns");
26    }
27 }
```

실행 결과

Console Problems @ Javadoc

<terminated> LinkedListExample [Java A
ArrayList 걸린시간: 6276500 ns
LinkedList 걸린시간: 2259100 ns

2) Set 컬렉션

- HashSet 1

```
1 package week5_1;
2 import java.util.*;
3
4 public class HashSetEx {
5     public static void main(String[] args) {
6         Set<String> set = new HashSet<String>();
7
8         set.add("Java");
9         set.add("JDBC");
10        set.add("Servlet/JSP");
11        set.add("Java");
12        set.add("iBATIS");
13
14        int size = set.size();
15        System.out.println("총 객체수: " + size);
16
17        Iterator<String> iterator = set.iterator();
18        while(iterator.hasNext()) {
19            String element = iterator.next();
20            System.out.println("\t" + element);
21        }
22
23        set.remove("JDBC");
24        set.remove("iBATIS");
25
26        System.out.println("총 객체수: " + set.size());
27
28        iterator = set.iterator();
29        while(iterator.hasNext()) {
30            String element = iterator.next();
31            System.out.println("\t" + element);
32        }
33
34        set.clear();
35        if(set.isEmpty())
36            System.out.println("비어있음.");
37    }
38 }
```

실행 결과

```
Problems @ Javadoc Declaration Console
<terminated> HashSetEx [Java Application] C:\WProgram
총 객체수: 4
    Java
    JDBC
    Servlet/JSP
    iBATIS
총 객체수: 2
    Java
    Servlet/JSP
비어있음.
```

2) Set 컬렉션 - HashSet 2

```
HashSetEx.java Member.java HashSetEx2.java Message.java Queue.java
1 package week5_1;
2
3 public class Member {
4
5     public String name;
6     public int age;
7     public Member(String name, int age) {
8         this.name = name;
9         this.age = age;
10    }
11
12    @Override
13    public boolean equals(Object obj) {
14        if(obj instanceof Member) {
15            Member member = (Member) obj;
16            return member.name.equals(name) && (member.age==age);
17        }else {
18            return false;
19        }
20    }
21
22    @Override
23    public int hashCode() {
24        return name.hashCode() + age;
25    }
26 }
```

```
HashSetEx.java Member.java HashSetEx2.java Message.java
1 package week5_1;
2 import java.util.*;
3
4 public class HashSetEx2 {
5     public static void main(String[] args) {
6         Set<Member> set = new HashSet<Member>();
7
8         set.add(new Member("홍길동", 30));
9         set.add(new Member("홍길동", 30));
10
11         System.out.println("총 객체수: " + set.size());
12     }
13 }
```

실행 결과

```
Problems @ Javadoc |
<terminated> HashSetEx2
총 객체수: 1
```


3) Queue

Queue 인터페이스에 정의되어 있는 메소드

리턴 타입	메소드	설명
boolean	offer(E e)	주어진 객체를 넣는다.
E	peek()	객체 하나를 가져온다. 객체를 큐에서 제거하지 않는다.
E	poll()	객체 하나를 가져온다. 객체를 큐에서 제거한다.

3) Queue

- Queue를 이용한 메시지 큐

```
Message.java QueueEx.java
1 package week5;
2
3 public class Message {
4
5     public String command;
6     public String to;
7
8     public Message(String command, String to) {
9         this.command = command;
10        this.to = to;
11    }
12 }
```

3) Queue

- Queue를 이용한 메시지 큐

```
Message.java QueueEx.java ✕
1 package week5;
2
3 import java.util.LinkedList;
4 import java.util.Queue;
5
6 public class QueueEx {
7     public static void main(String[] args) {
8         Queue<Message> messageQueue = new LinkedList<Message>();
9
10        messageQueue.offer(new Message("sendMail", "홍길동"));
11        messageQueue.offer(new Message("sendSMS", "홍두깨"));
12        messageQueue.offer(new Message("sendKakaotalk", "정원호"));
13
14        while(!messageQueue.isEmpty()) {
15            Message message = messageQueue.poll();
16            switch(message.command) {
17                case "sendMail":
18                    System.out.println(message.to + "님에게 메일을 보냅니다.");
19                    break;
20                case "sendSMS":
21                    System.out.println(message.to + "님에게 SMS을 보냅니다.");
22                    break;
23                case "sendKakaotalk":
24                    System.out.println(message.to + "님에게 Kakaotalk을 보냅니다.");
25                    break;
26            }
27        }
28    }
29 }
```

실행 결과

```
Console ✕ Problems @ Javado
<terminated> QueueEx [Java Applica
홍길동님에게 메일을 보냅니다.
홍두깨님에게 SMS을 보냅니다.
정원호님에게 Kakaotalk을 보냅니다.
```

출석 과제 1 (4/12 월 오후 11:55 마감)

Q1. List 컬렉션에 대한 설명 중 틀린 것은 무엇입니까?

- 1) 대표적인 구현 클래스로는 ArrayList, Vector, LinkedList가 있다.
- 2) 멀티 스레드 환경에서는 ArrayList보다는 Vector가 스레드에 안전하다.
- 3) ArrayList에서 객체를 삭제하면 삭제된 위치는 비어 있게 된다.
- 4) 중간 위치에 객체를 빈번히 삽입하거나 제거할 경우 LinkedList를 사용하는 것이 좋다.

출석 과제 2 (4/12 월 오후 11:55 마감)

Q2. Set 컬렉션에 대한 설명 중 틀린 것은 무엇입니까?

- 1) 대표적인 구현 클래스로는 HashSet, LinkedHashSet, TreeSet이 있다.
- 2) Set 컬렉션에서 객체를 하나씩 꺼내고 싶다면 Iterator를 이용한다.
- 3) HashSet은 hashCode()와 equals()를 이용해서 중복된 객체를 판별한다.
- 4) Set 컬렉션에는 null을 저장할 수 없다.