데이터융합SW과
김규석 교수

JAVA

기본 프로그래밍 07

Objective

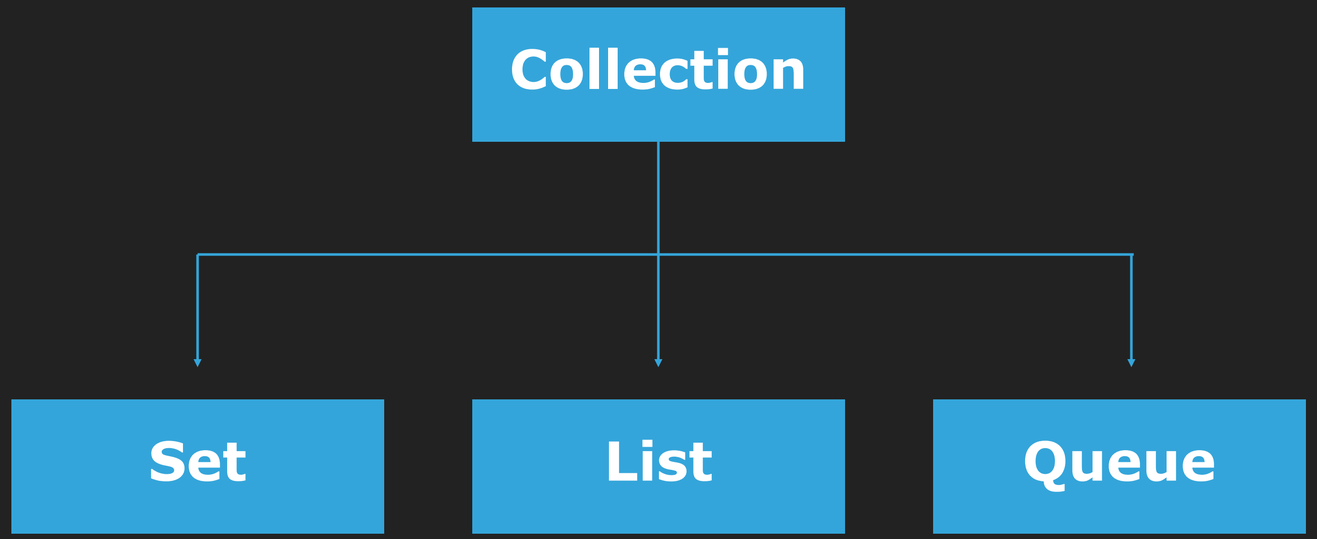
Collection

- List
- Set
- Queue
- Stack

Collection(Cont'd)

Java Collections Framework

- A set of classes and interfaces that implement commonly reusable collection data structures



Set

Hashset

- A collection of items and the items are unique

```
 9⊖    public static void main(String[] args) {
10        HashSet<String> name = new HashSet<String>();
11        name.add("kim");
12        System.out.println(name);
13        name.add("lee");
14        System.out.println(name);
15        name.add("kim");
16        System.out.println(name);
17    }
```

Problems | Javadoc | Declaration | Console

<terminated> Main [Java Application] C:₩Users₩CTC₩.p2₩poo

```
[kim]
[lee, kim]
[lee, kim]
```

List(Cont'd)

ArrayList

- A resizable array

```
8⊖    public static void main(String[] args) {
9         ArrayList<String> name = new ArrayList<String>();
10        name.add("kim");
11        name.add("lee");
12        name.add("john");
13
14        System.out.println(name);
15        System.out.println(name.size());
16        System.out.println(name.get(2));
17
18        name.remove(1);
19        System.out.println(name);
20
21        name.set(1, "park");
22        System.out.println(name);
23        System.out.println(name.size());
24
25        name.add("lee");
26        System.out.println(name);
27
28        Collections.sort(name);
29        System.out.println(name);
30
31        Collections.reverse(name);
32        System.out.println(name);
33    }
```

Problems | Javadoc | Declaration | Console

<terminated> Main [Java Application] C:₩Users₩CTC₩.p2₩po

```
[kim, lee, john]
3
john
[kim, john]
[kim, park]
2
[kim, park, lee]
[kim, lee, park]
[park, lee, kim]
```

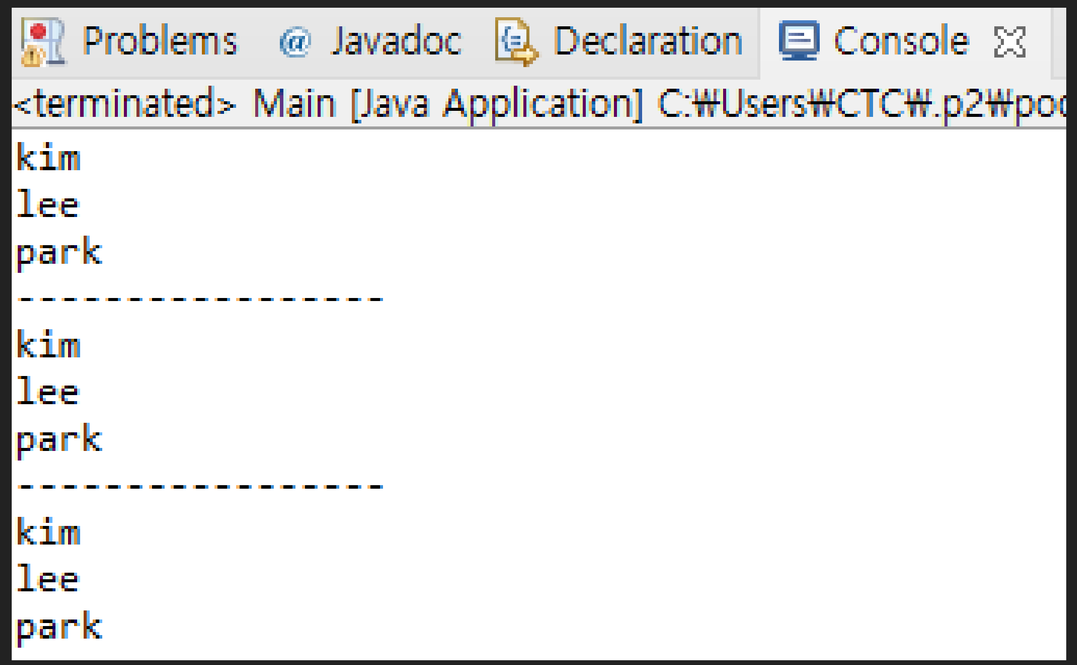
Q1* : What's the difference between "Hashset" and "ArrayList"?

List

Iterator

- An object that can be used to loop through collections

```
16      public static void main(String[] args) {
17          List<String> list = new ArrayList<String>();
18          list.add("kim");
19          list.add("lee");
20          list.add("park");
21
22          Iterator<String> iterator = list.iterator();
23
24          while(iterator.hasNext()) {
25              System.out.println(iterator.next());
26          }
27          System.out.println("-----------------");
28
29          for (String name : list) {
30              System.out.println(name);
31          }
32
33          System.out.println("-----------------");
34          for (int i = 0; i < list.size(); i++) {
35              System.out.println(list.get(i));
36          }
37      }
```



Queue

Queue

- A data structure designed to have elements inserted at the end of the queue, and elements removed from the beginning of the queue

```
11⊖    public static void main(String[] args) {
12         Queue<String> name = new LinkedList<String>();
13         name.offer("Kim");
14         System.out.println(name);
15         name.offer("lee");
16         System.out.println(name);
17         System.out.println(name.poll());
18         System.out.println(name.poll());
19         System.out.println(name);
20     }
```

Q2* : Explain FIFO and FILO

Source : http://tutorials.jenkov.com/java-collections/queue.html#:~:text=The%20Java%20Queue%20interface%2C%20java,of%20the%20Java%20Collection%20interface.

Collection+

Stack

- A linear data structure that is used to store the collection of objects.
- Based on LIFO(Last-In-First-Out)

```
12    public static void main(String[] args) {
13        Stack<String> stack = new Stack<String>();
14        stack.add("kim");
15        System.out.println(stack);
16        stack.add("lee");
17        System.out.println(stack);
18        System.out.println(stack.pop());
19        System.out.println(stack);
20    }
```

Q3* : What's the difference between "Queue" and "Stack"?

Source : https://www.javatpoint.com/java-stack#:~:text=The%20stack%20is%20a%20linear,store%20the%20collection%20of%20objects.&text=The%20stack%20data%20structure%20has,the%20top%20of%20the%20stack.

P1

Split a String and Sort

- User inputs a string
- Print the words after splitting it against by a black(“ “)
- Sort them by in descending order and print them again

P2

Exception Handling

- User inputs a number and the number defines the size of an array
- User inputs numbers more than the defined size
- In this case, this program prints “error” and starts again from the beginning, Otherwise, assign the numbers into the array, sort and print them

```
5
1 3 4 5 7 8
error
4
1 22 6 12
1 6 12 22
```

P3

Fibonacci Numbers

- The formula for this is as below

  $F_0 = 0, F_1 = 1$

  $F_n = F_{n-1} + F_{n-2}$

  0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55 …

- User inputs a number

- Print the number of elements from the Fibonacci Numbers

- Add exception statements

7
0 1 1 2 3 5 8

P4

Comparing Strings

- User inputs the two words and assign them to the arrays

- Print the following information
  1. Size of the first word
  2. Size of the second word
  3. Number of the same characters

- Add exception statements

school
scholar
1 : 6
2 : 7
3 : 5

P5

Numbering

- User inputs a series of numbers and assign them to a queue
- User input a number from the series of numbers
- Print the number in order of output
- Add exception statements

```
1 21 13 34 15 16
13
3
```

P6

Numbering II

- User inputs a series of numbers and assign them to a stack
- User input a number from the series of numbers
- Print the number in order of output
- Add exception statements

```
1 21 13 34 15 16
13
4
```

Sort

Sort

```
15      public static void main(String[] args) {
16          // TODO Auto-generated method stub
17          String[] temp = {"lee", "kim", "john", "bill", "mike"};
18          Arrays.sort(temp);
19          for (int i = 0; i < temp.length; i++) {
20              System.out.println(temp[i]);
21          }
22          System.out.println("====================");
23          Arrays.sort(temp, Collections.reverseOrder());
24          for (int i = 0; i < temp.length; i++) {
25              System.out.println(temp[i]);
26          }
27      }
```

Problems @ Javadoc Declaration Console

<terminated> Main [Java Application] C:₩Users₩kopo₩.p2₩p

```
bill
john
kim
lee
mike
====================
mike
lee
kim
john
bill
```

P7

Grade Card

- User inputs more than 10 sets of a name and a score
- Print the list by name in ascending order
- Print the list by name in descending order
- Print the list by score in ascending order
- Print the list by score in descending order
- Add exception statements

P8

Multiples

- User inputs a series of numbers and a number
- Print the multiple numbers of N
- Add exception statements

```
1 30 2.3 42 23 41 50
5
30 50
```

P9

Base N Numbers

- User inputs the two numbers for M and N
- Print M in base N
- Add exception statements

```
7
111
```