

Data Structure

Fall 2019

M 16:00-18:00 W 11:00-13:00

<http://smart.hallym.ac.kr>

Instructor:

Jin Kim

010-6267-8189(033-248-2318)

jinkim@hallym.ac.kr

Office Hours:

Lab(Array 2)

Fall 2019

<http://smart.hallym.ac.kr>

Instructor: Jin Kim
010-6267-8189(033-248-2318)

jinkim@hallym.ac.kr

Office Hours:

Reference Site

참고사이트

- ◆ <https://www.geeksforgeeks.org/>

Java

Collections(컬렉션)

목록성 데이터를 처리하는 자료구조

- ◆ **collection**: an object that stores data; a.k.a. "data structure"
 - ◆ the objects stored are called **elements**
 - ◆ some collections maintain an ordering; some allow duplicates
 - ◆ typical operations: *add*, *remove*, *clear*, *contains* (search), *size*
 - ◆ examples found in the Java class libraries:
(covered in this course!)
 - ArrayList, LinkedList, HashMap, TreeSet, PriorityQueue
 - ◆ all collections are in the `java.util` package
`import java.util.*;`

Lab

- ◆ 각 프로그램들을 실제 구현해보고, 스마트 캠퍼스에 업로드하라.
- ◆ 두 개의 빵틀 Arrays, ArrayList를 실습해본다.

Java Arrays

- ◆ <https://docs.oracle.com/javase/7/docs/api/java/util/Arrays.html>
- ◆ 여러 static (정적) 메소드가 존재한다.
- ◆ Java.util.Arrays라는 뺱틀(클래스)을 가져와(import) 사용할 수 있다.

ArraysTest.java

Array.binarySearch()

// 배열 intArr에서 intKey값이 저장되어 있는 방의 번호를 이진탐색으로
찾아라

/ Java program to demonstrate

// Array.binarySearch() method

import java.util.Arrays;

public class Main {

public static void main(String[] args) {

// Get the Array

int intArr[] = { 10, 20, 15, 22, 35 };

Arrays.sort(intArr); //정렬

int intKey = 22;

System.out.println(intKey

+ " found at index = "

+ Arrays.binarySearch(intArr, intKey));

}

}

클래스(빵틀)

메소드

compare(array1, array2)

import java.util.Arrays;

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

```
        // Get the Array  
        int intArr[] = { 10, 20, 15, 22, 35 };
```

```
        // Get the second Array  
        int intArr1[] = { 10, 15, 22 };
```

```
        // To compare both arrays  
        System.out.println("Integer Arrays on comparison: "  
            + Arrays.compare(intArr, intArr1));
```

```
    }  
}
```

클래스(빡틀)

메소드

equals(array1, array2)

// Java program to demonstrate

// Array.equals() method

import java.util.**Arrays**;

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

// Get the Arrays

int intArr[] = { 10, 20, 15, 22, 35 };

// Get the second Arrays

int intArr1[] = { 10, 15, 22 };

// To compare both arrays

System.out.println("Integer Arrays on comparison: "

+ Arrays.**equals**(intArr, intArr1));

}

}

클래스(빵틀)

메소드

sort(array1)

// Java program to demonstrate

// Array.sort() method

import java.util.**Arrays**;

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

// Get the Array

int intArr[] = { 10, 20, 15, 22, 35 };

// To sort the array using normal sort-

Arrays.**sort**(intArr);

System.out.println("Integer Array: "
 + Arrays.toString(intArr));

}

}

클래스(빵틀)

메소드

Sort the given matrix

Given a $n \times n$ matrix. The problem is to sort the given matrix in strict order. Here strict order means that matrix is sorted in a way such that all elements in a row are sorted in increasing order and for row 'i', where $1 \leq i \leq n-1$, first element of row 'i' is greater than or equal to the last element of row 'i-1'.

```
Input : mat[][] = { {5, 4, 7},  
                    {1, 3, 8},  
                    {2, 9, 6} }
```

```
Output : 1 2 3  
         4 5 6  
         7 8 9
```

SortMatrix.java

```
// Java implementation to
// sort the given matrix
import java.io.*;
import java.util.*;
```

```
class SortMatrix {

    static int SIZE = 10;

    // function to sort the given matrix
    static void sortMat(int mat[][], int n){
        // temporary matrix of size n^2
        int temp[] = new int[n * n];
        int k = 0;

        // copy the elements of matrix
        // one by one into temp[]
        for (int i = 0; i < n; i++)
            for (int j = 0; j < n; j++)
                temp[k++] = mat[i][j];

        // sort temp[]
        Arrays.sort(temp);

        // copy the elements of temp[]
        // one by one in mat[][]
        k = 0;
        for (int i = 0; i < n; i++)
            for (int j = 0; j < n; j++)
                mat[i][j] = temp[k++];
    }
```

```
    // function to print the given matrix
    static void printMat(int mat[][], int n)
    {
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < n; j++)
                System.out.print( mat[i][j] +
                " ");

            System.out.println();
        }
    }

    // Driver program to test above
    public static void main(String args[]){
        int mat[][] = { { 5, 4, 7 },
                        { 1, 3, 8 },
                        { 2, 9, 6 } };

        int n = 3;

        System.out.println("Original
        Matrix:");
        printMat(mat, n);
        sortMat(mat, n);
        System.out.println("Matrix After
        Sorting:");
        printMat(mat, n);
    }
}
```

toString(array)

array의 내용을 문자열로 리턴

// Java program to demonstrate
// Arrays.toString() method

```
import java.util.Arrays;

public class Main {
    public static void main(String[] args) {
        // Get the Array
        int intArr[] = { 10, 20, 15, 22, 35 };

        // To print the elements in one line
        System.out.println("Integer Array: "
            + Arrays.toString(intArr));
    }
}
```

Move all zeroes to end of array

Given an array of random numbers, Push all the zero's of a given array to the end of the array. For example, if the given arrays is {1, 9, 8, 4, 0, 0, 2, 7, 0, 6, 0}, it should be changed to {1, 9, 8, 4, 2, 7, 6, 0, 0, 0, 0}. The order of all other elements should be same.

Example

Input : arr[] = {1, 2, 0, 4, 3, 0, 5, 0};

Output : arr[] = {1, 2, 4, 3, 5, 0, 0, 0};

Input : arr[] = {1, 2, 0, 0, 0, 3, 6};

Output : arr[] = {1, 2, 3, 6, 0, 0, 0};

MoveZero.java

```
import java.io.*;

class MoveZero {
    // Function which pushes all zeros to end of an array.
    static void pushZerosToEnd(int arr[], int n){
        // 채워라
    }

    /*Driver function to check for above functions*/
    public static void main (String[] args)
    {
        int arr[] = {1, 9, 8, 4, 0, 0, 2, 7, 0, 6, 0, 9};
        int n = arr.length;
        pushZerosToEnd(arr, n);
        System.out.println("Array after pushing zeros to the back: ");
        for (int i=0; i<n; i++)
            System.out.print(arr[i]+" ");
    }
}
```


Java ArrayList

- ◆ <https://docs.oracle.com/javase/8/docs/api/java/util/ArrayList.html>
여러 static (정적) 메소드가 존재한다
- ◆ `Import Java.util.ArrayList;`

ArrayListTest.java

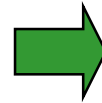
- ◆ ArrayListExample.java를 테스트해보라.
- ◆ 열 개의 문자열을 삽입하라.
- ◆ 다섯 번째 원소를 출력하라.
- ◆ 다섯 번째 원소를 제거하라.
- ◆ 원소의 개수를 출력하라
- ◆ 원소의 개수 -3 번째 원소를 출력하라.
- ◆ 모든 원소를 출력하라.
- ◆ 원소 “Kim”이 존재하는지 확인하라.

- ◆ 주어진 프로그램을 사용하여 다음 페이지의 행렬을 3원소쌍으로 표현하고 전치행렬을 만들어라.

Efficient representation of sparse matrix matrix(희소 행렬의 효율적인 표현)

Row(행) Col(열) val(값)

	0	1	2	3	4	5
0	15	0	0	22	0	-15
1	0	11	3	0	0	0
2	0	0	0	-6	0	0
3	0	0	0	0	0	0
4	91	0	0	0	0	0
5	0	0	28	0	0	0
6	0	0	3	0	9	0
7	0	0	2	0	0	1



Ex: sparse matrix(희소 행렬)

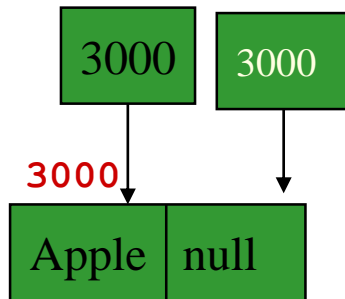
Ordered list representation
(순서 리스트 표현)

Linked list(연결리스트)

```
// 1. list2 = ("apple", "strawberry", "orange", "banana",  
"melon") 만들어라.  
//선두에 원소추가하라  
// 2. list2 = ("pineapple", "apple", "strawberry",  
"orange", "banana", "melon") 만들어라  
// 두번째 원소를 제거하라.  
// 3. list2 = ("pineapple", "strawberry", "orange",  
"banana", "melon")  
// 4. list2의 원소의 개수를 계산하여 출력하라.  
// 5. list2의 마지막에 "durian"을 추가하라  
//list2 = ("pineapple", "strawberry", "orange", "banana",  
"melon", "durian")
```

LinkedList Class

Head 혹은 listname tail

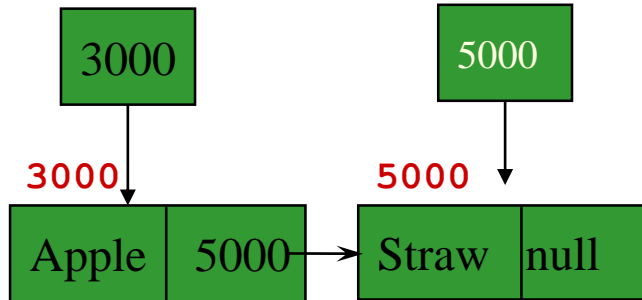


LinkedList Class

```
// L 대신 실제 프로그램에서 사용  
// 첫번째 원소의 주소를 가짐
```

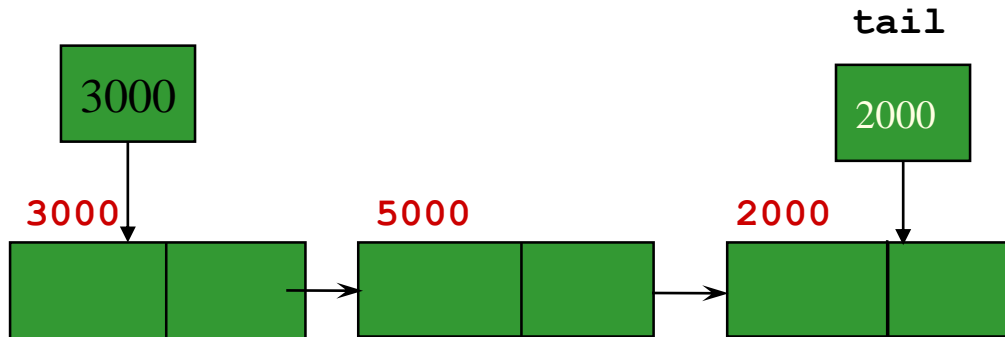
head

tail

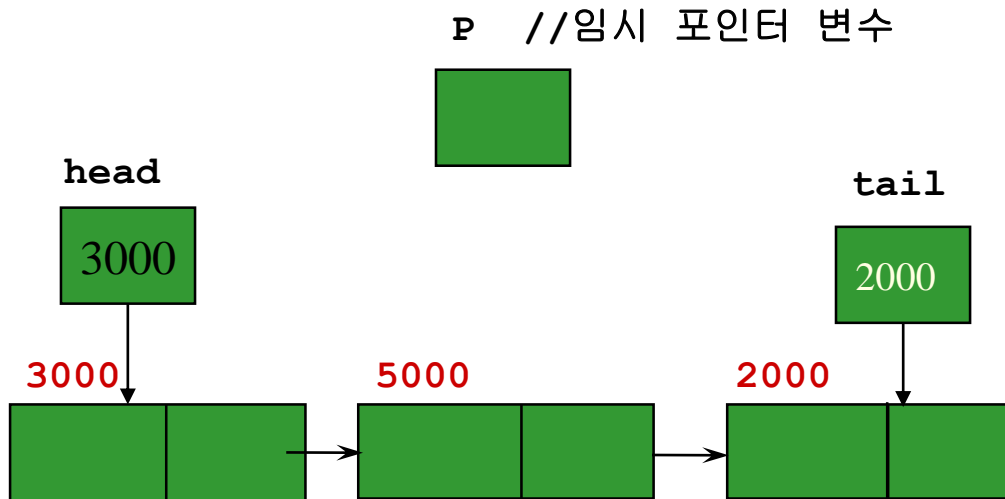


LinkedList Class

```
// l 대신 실제 프로그램에서 사용  
// 첫번째 원소의 주소를 가짐  
head
```



LinkedList Class



1. ArraysTest.java
2. ArrayListTest.java
3. SparseMatrixMain.java
4. LinkedListBasic.java

Zip all your programs(name.zip) and
upload to smart.hallym.ac.kr