



# 자료구조 10주차

## 연결리스트

MMC 연구실

석사 과정 강민제

## 조교 소개

- 강민제
- 컴퓨터공학과 대학원 석사과정
- MMC연구실 (A1406)
- rkdalswp29@gmail.com



## 실습 수업 진행 방식

- 확인 문제 풀이
- 확인 문제를 해결한 학생은 검사 받고 퇴실



## 과제 설명

- 자료구조 수업은 Eclipse를 사용하여 코드를 작성합니다.
- 확인 문제 및 과제를 전부 해결하여 제출해주세요.
- 과제 제출 시 **프로젝트 폴더를 압축**해서 제출합니다.
- 과제의 채점은 프로젝트의 실행 결과를 기준으로 점수를 매깁니다.

# 실습 예제 (연결 리스트)

Package Name : list\_test

Class Name : Test

```
public class Test {  
    public static void main(String[] args) {  
        class ListNode {  
            String name;  
            ListNode link;  
        }  
        ListNode L = new ListNode();  
        L.name = "Kim";  
        L.link = null;  
  
        L.link = new ListNode();  
        L.link.name = "Lee";  
  
        L.link.link = new ListNode();  
        L.link.link.name = "Park";  
  
        ListNode current = L;  
        while (current != null) {  
            System.out.println(current.name);  
            current = current.link;  
        }  
    }  
}
```

```
Console X  
<terminated> Test [Jav  
Kim  
Lee  
Park
```

## ◆ 스트링 리스트에 대한 연결 표현을 위한 노드 구조

```
class ListNode {  
    String name;  
    ListNode link;  
}
```

- name 필드 - String값을 가짐
- link 필드 - 다른 ListNode 객체를 가리키는 참조값을 가짐
- 리스트는 link 필드들을 통해 연결되는 ListNode 클래스의 객체들이 됨
- link 값이 null이면 그 리스트의 마지막 노드라는 것을 나타냄
- ListNode 변수가 null이면 공백리스트를 나타냄

## 확인문제 1 (연결 리스트)

Package Name : list

Class Name : ListNode

```
package list;

public class ListNode {
    private String name;
    private ListNode link;

    public ListNode() {
        link = null;
    }

    public ListNode(String name) {
        this.name = name;
        link = null;
    }

    public ListNode(String name, ListNode link) {
        this.name = name;
        this.link = link;
    }

    public void setName(String name) { this.name = name; }

    public void setLink(ListNode link) { this.link = link; }

    public String getName() { return name; }

    public ListNode getLink() { return link; }
}
```

list/ListNode.txt 파일 참고

# 확인문제 1 (연결 리스트)

Package Name : list

Class Name : LinkedList

```
package list;

public class LinkedList {
    private ListNode first;

    public LinkedList() {
        first = null;
    }

    public void addFirst(String name) {
        // 연결리스트의 맨 앞에 노드를 추가
    }

    public void insert(String name, ListNode target) {
        // target 노드 뒤에 새로운 노드를 추가
    }

    public ListNode searchNode(String name) {
        // 전달 인자로 받은 name을 연결리스트에서 탐색하여 해당 노드 반환
    }

    public void delete(ListNode p) {
        // 전달 받은 노드 p 뒤의 노드를 삭제
    }

    public void print() {
        // 연결리스트의 모든 노드의 name을 출력
    }
}
```

list/LinkedList.txt 파일 참고

## 확인문제 1 (연결 리스트)

Package Name : list

Class Name : LinkedListMain

```
package list;

public class LinkedListMain {

    public static void main(String[] args) {
        LinkedList list1 = new LinkedList();
        list1.insert("Kim", null);
        list1.addFirst("Park");
        list1.addFirst("Joo");
        list1.insert("Kang", list1.searchNode("Joo")); // Joo 뒤에 Kang 삽입
        list1.print();
        System.out.println("-----");
        list1.delete(list1.searchNode("Park")); // Park 뒤에 노드(Kim) 삭제
        list1.print();
    }
}
```

Console X

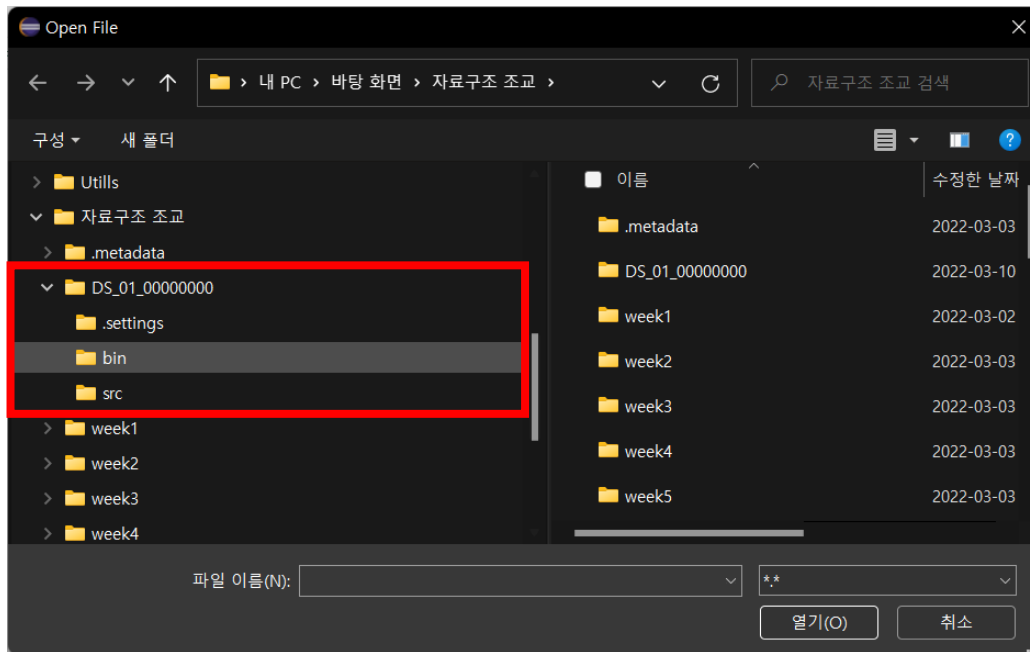
<terminated> LinkedListMain [Java Appl

Joo  
Kang  
Park  
Kim  
-----  
Joo  
Kang  
Park

list/LinkedListMain.txt 파일 참고



# 과제 제출 방법



- 프로젝트 폴더를 압축하여 제출

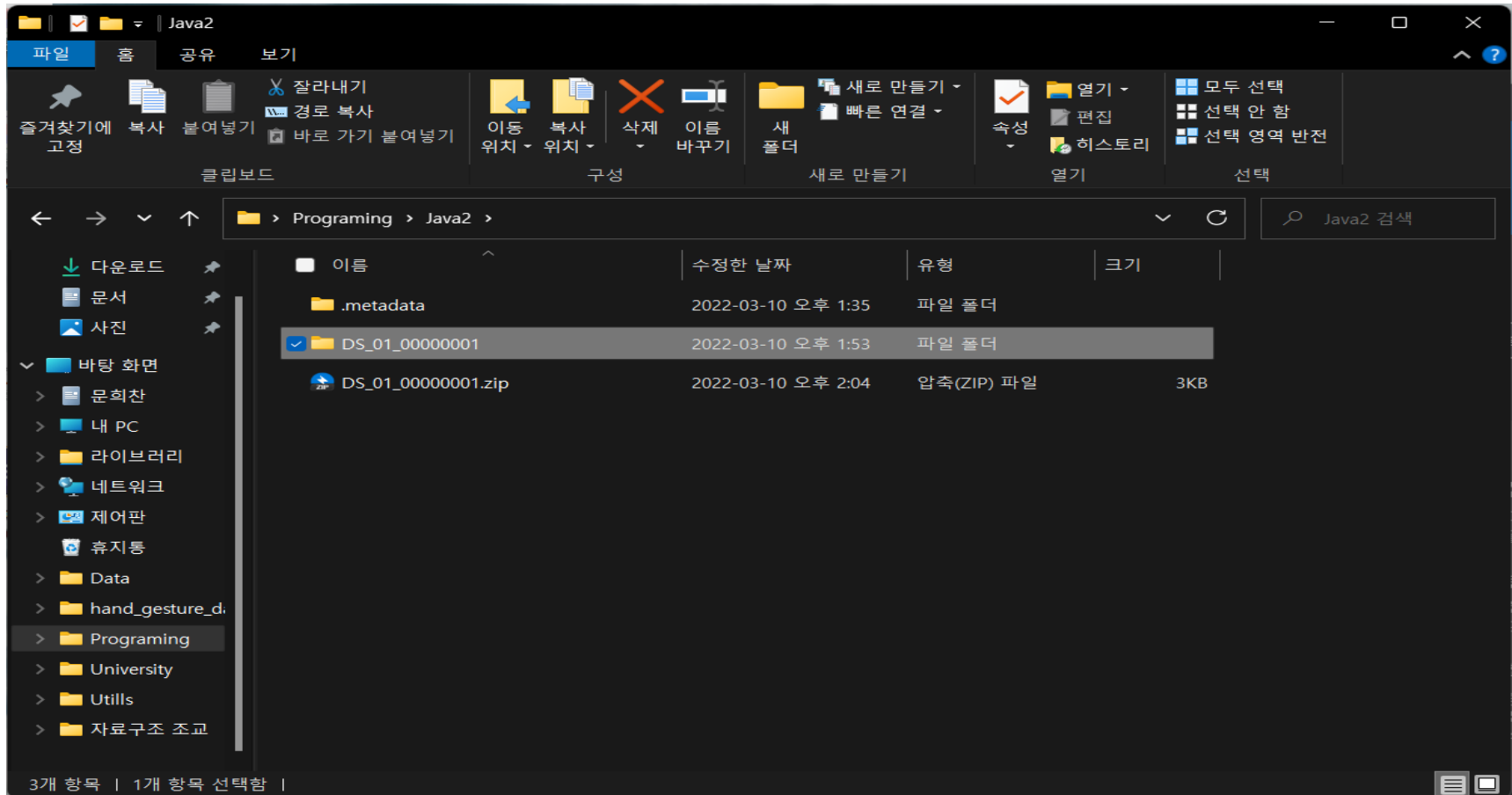
- 프로젝트이름 : DS\_(주차)\_(학번)

예) DS\_10\_00000000

- \*.java파일만 제출하면 안됩니다.

**제출양식을 반드시 지켜주세요**

# 과제 제출 방법



- 반드시 **프로젝트 폴더를 압축**하여 제출