



# 알고리즘 8주차

## Graph 2

MMC 연구실

박사 과정 문희찬

## 조교 소개

- 문희찬
- 컴퓨터공학과 대학원 석사과정
- MMC연구실 (A1406)
- HCMoon@hallym.ac.kr



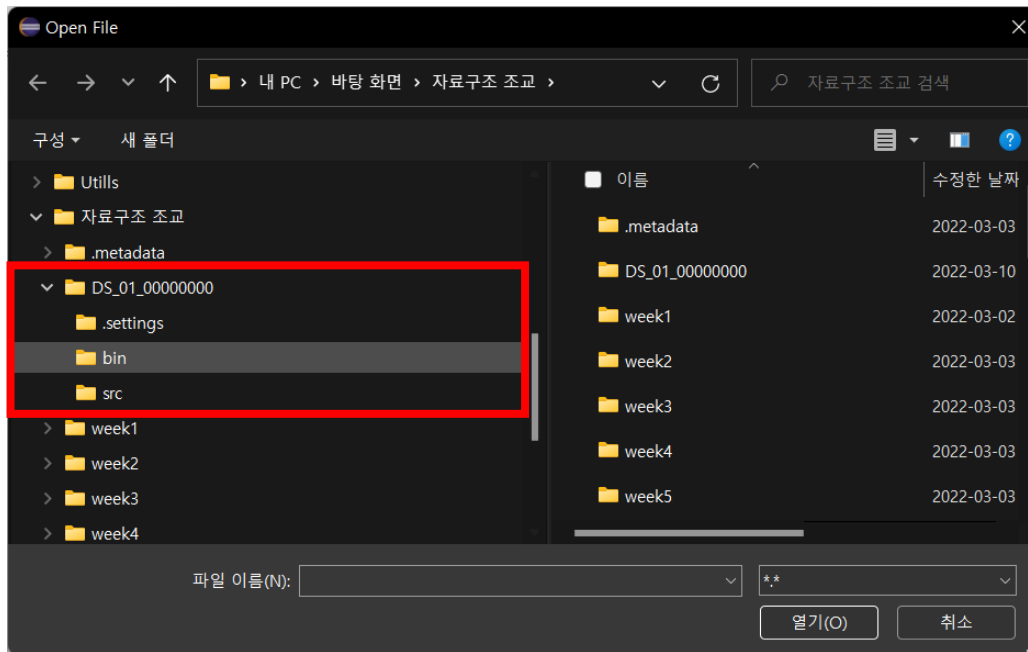
## 실습 수업 진행 방식

- 쉬는 시간 없이 1시간 30분 수업 (화장실 자유롭게 다녀오세요)
- 출석체크 : 수업 시작, 수업 끝날 때 체크
- 수업 시작 30분 뒤부터, 확인 문제를 해결한 학생은 검사 받고 퇴실
- 과제 진행 중 모르는 부분은 메일로 질문

## 과제 설명

- 알고리즘 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.
- 컨닝 금지, 모르는 것이 있으면 저에게 질문해주세요.  
(메일 주소 확인)

# 과제 제출 방법



- 프로젝트 폴더를 압축하여 제출

- 프로젝트이름 : AL\_(주차)\_(학번)

예) AL\_07\_00000000

- \*.java파일만 제출하면 안됩니다.

- 제출양식을 반드시 지켜주세요!

## 확인문제

Package Name : digraph

Class Name : DiGraph

- 7주차 과제를 수정하여 directed graph 구현



# 확인문제

Package Name : digraph


Class Name : DiGraph

```
public class DiGraph {  
    int n; // Number of vertices  
    int e; // Number of edges  
    int[][] weight;  
  
    public DiGraph(int noOfVertices) {  
        n = noOfVertices;  
        e = 0;  
        weight = new int[n][n];  
    }  
  
    public void insertEdge(int i, int j) {  
    }  
  
    public void removeEdge(int i, int j) {  
    }  
  
    public int[] adjacency(int u) {  
    }  
  
    public void bfs(int v) {  
    }  
  
    public void dfs(int v) {  
    }  
}
```

# 확인문제

Package Name : digraph  
Class Name : DiGraphTest

```
public static void main(String[] args) {  
    DiGraph gr = new DiGraph(6);  
    gr.insertEdge(0, 1);  
    gr.insertEdge(0, 2);  
    gr.insertEdge(1, 2);  
    gr.insertEdge(1, 3);  
    gr.insertEdge(2, 3);  
    gr.insertEdge(2, 4);  
    gr.insertEdge(3, 4);  
    gr.insertEdge(3, 5);  
    gr.insertEdge(4, 5);  
    gr.insertEdge(1, 5);  
  
    System.out.println();  
    int[] adj;  
    for(int i = 0; i < 6; i++) {  
        adj = gr.successor(i);  
        System.out.print(i + ": ");  
        for(int e: adj) {  
            System.out.print(e + " ");  
        }  
        System.out.println();  
    }  
  
    gr.bfs(0);  
    System.out.println();  
    gr.dfs(0);  
    System.out.println();  
}
```



```
0: 1 2  
1: 2 3 5  
2: 3 4  
3: 4 5  
4: 5  
5:  
BFS  
0, 1, 2, 3, 5, 4,  
DFS  
0, 2, 4, 5, 3, 1,
```



# 실습 과제

## 1. Heap 삭제 메소드 구현



## 과제 1

Package Name : heap

Class Name : Heap

- 6주차 과제에 delete 메소드 추가

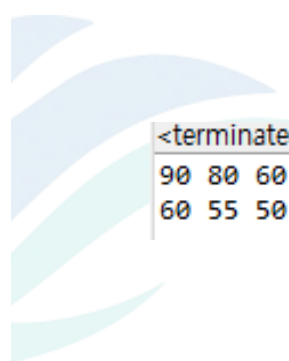
```
public int delete() {  
    if(count == 0) {  
        return -100;  
    } else {  
    }  
}
```



# 과제 1

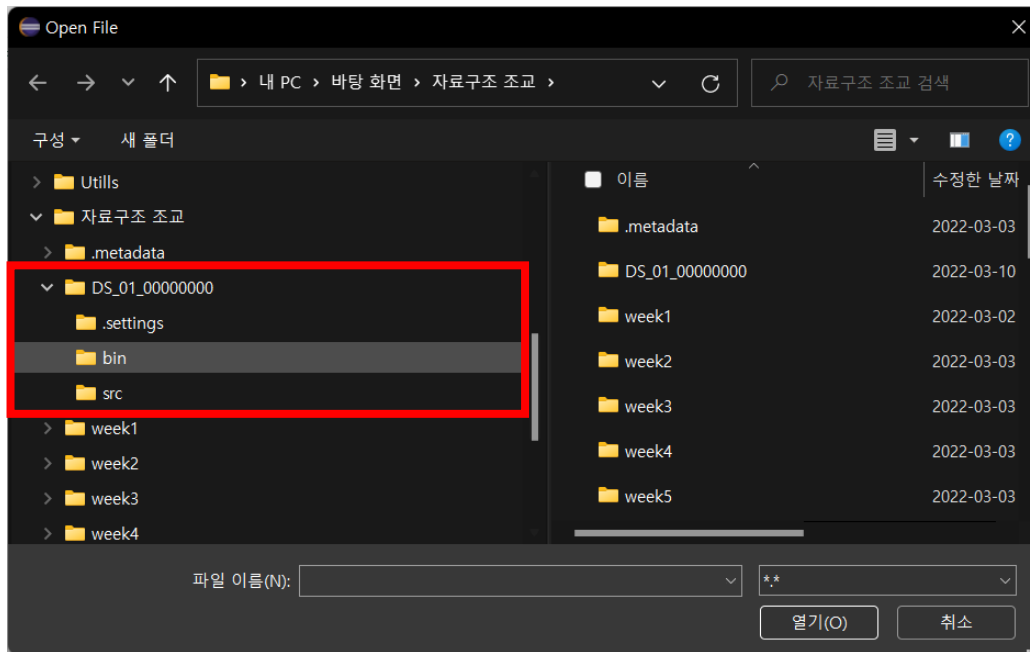
Package Name : heap  
Class Name : Heap

```
public static void main(String[] args) {  
    Heap h = new Heap();  
    int[] origArray = {0, 50, 55, 60, 30, 70, 90, 25, 80, 40, 45};  
    h = Heap.makeHeap(origArray);  
    h.printHeap();  
  
    h.delete();  
    h.delete();  
    h.delete();  
  
    h.printHeap();  
}
```



```
<terminated> TestHeap (1) [Java Application]  
90 80 60 55 70 50 25 30 40 45  
60 55 50 40 45 30 25
```

# 과제 제출 방법



- 프로젝트 폴더를 압축하여 제출

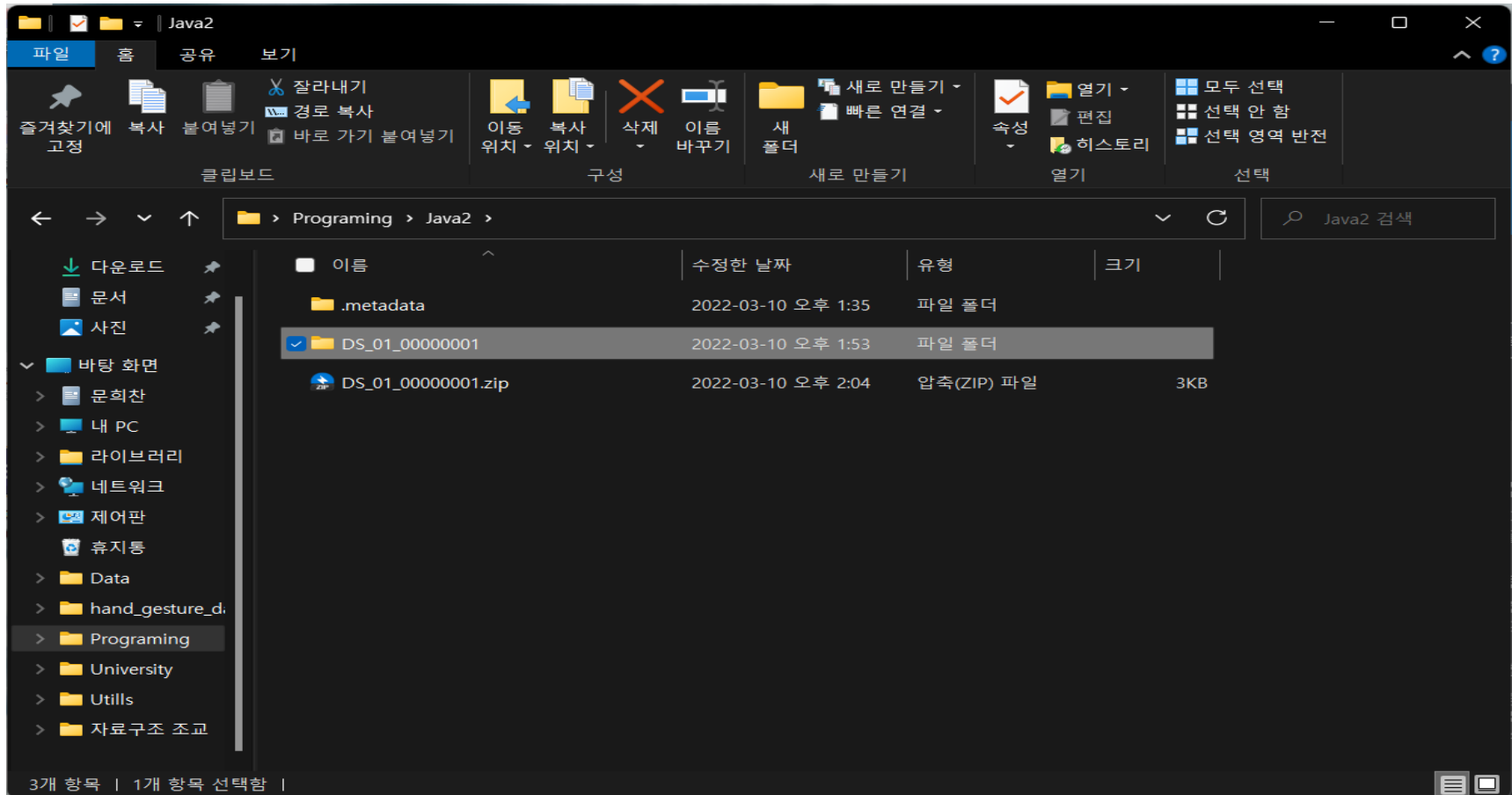
- 프로젝트이름 : AL\_(주차)\_(학번)

예) AL\_07\_00000000

- \*.java파일만 제출하면 안됩니다.

**제출양식을 반드시 지켜주세요**

# 과제 제출 방법



- 반드시 **프로젝트 폴더를 압축**하여 제출