알고리즘 7주차

Graph 1

MMC 연구실

박사 과정 문희찬

조교 소개

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

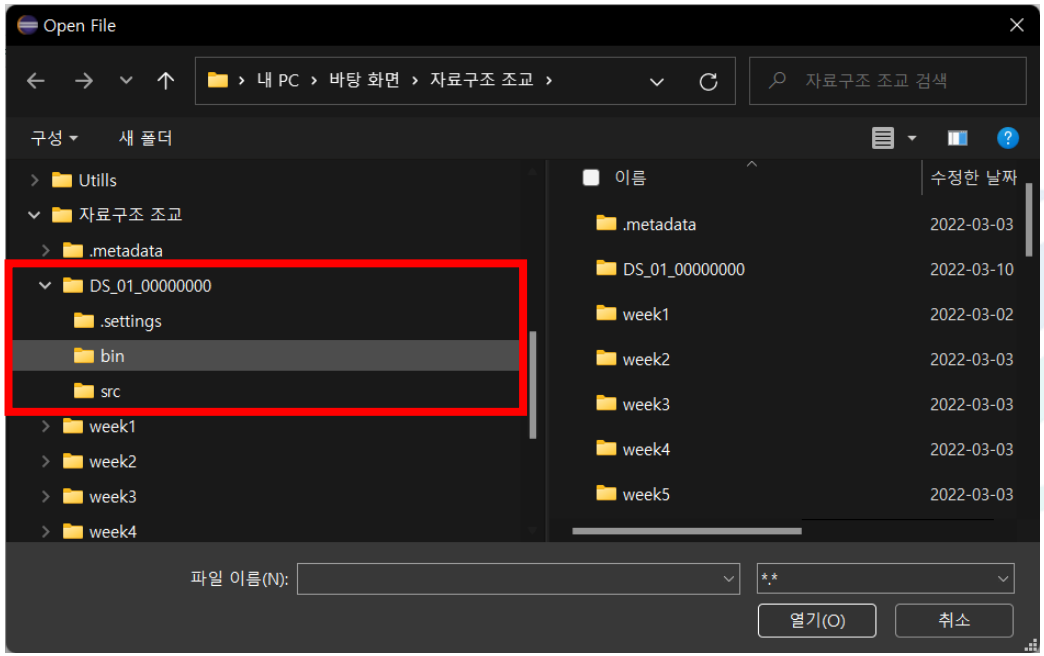
실습 수업 진행 방식

- 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 압축해서 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 : graph
Class Name : Graph

```java
public class Graph {
    int n; // Number of vertices
    int e; // Number of edges
    int[][] weight;

    public Graph(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 : graph
Class Name : GraphTest

```
public static void main(String[] args) {
    // TODO Auto-generated method stub
    Graph gr = new Graph(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.adjacency(i);
        System.out.print(i + ": ");
        for(int e: adj) {
            System.out.print(e + " ");
        }
        System.out.println();
    }

    gr.bfs(1);
    System.out.println();
    gr.dfs(1);
    System.out.println();
}
```

```
<terminated> GraphTest [Java Application]

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

실습 과제

1. dfsComponent 구현

2. 연결리스트를 이용하여 Graph 구현

과제 1

Package Name : graph
Class Name : Graph

- 연결 요소를 찾는 알고리즘

```
dfsComponent(G, n)     // G=(V,E), n은 G의 정점 수
  for (i←0; i <n; i ← i +1) do {
       visited[i] ← false;
  }
  for (i ←0; i <n; i ← i +1) do {
    // 모든 정점 0, 1, …, n-1에 대해 연결 요소 검사
       if (visited[i] = false) then {
            print("new component");
            DFS(i);     // 정점 i가 포함된 연결 요소를 탐색
       }
  }
end dfsComponent()
```

과제 1

Package Name : graph
Class Name : Graph

```
public void dfsComponent() {

}
```

과제 1

Package Name : graph
Class Name : GraphTest

```
public static void main(String[] args) {
    Graph gr = new Graph(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.adjacency(i);
        System.out.print(i + ": ");
        for(int e: adj) {
            System.out.print(e + " ");
        }
        System.out.println();
    }

    gr.bfs(1);
    System.out.println();
    gr.dfs(1);
    System.out.println();

    gr = new Graph(9);
    gr.insertEdge(0, 1);
    gr.insertEdge(0, 2);
    gr.insertEdge(2, 6);
    gr.insertEdge(1, 7);
    gr.insertEdge(3, 4);
    gr.insertEdge(5, 8);

    gr.dfsComponent();
```

```
<terminated> GraphTest [Java Application]

0: 1 2
1: 0 2 3 5
2: 0 1 3 4
3: 1 2 4 5
4: 2 3 5
5: 1 3 4
BFS
1, 0, 2, 3, 5, 4,
DFS
1, 5, 4, 3, 2, 0,
New Component
0, 2, 6, 1, 7, New Component
3, 4, New Component
5, 8,
```

과제 2

Package Name : graph
Class Name : GraphList

```
public class GraphList {
    int n; // Number of vertices
    int e; // Number of edges
    Node[] header;

    public GraphList(int noOfVertices) {
        n = noOfVertices;
        e = 0;
        header = new Node[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) {

    }
}
```

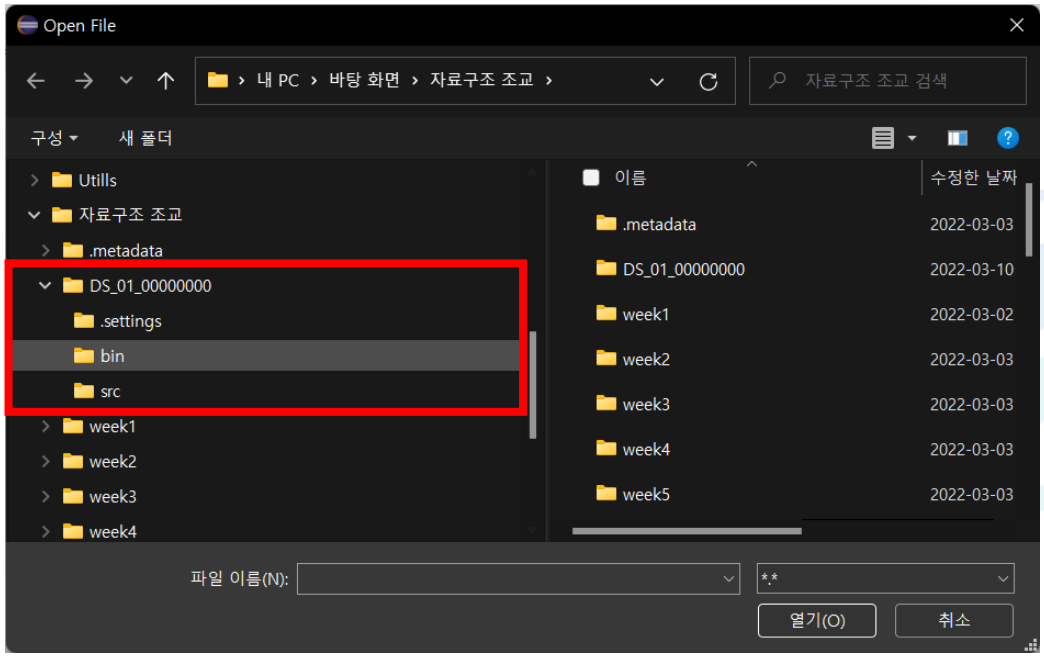
과제 2

Package Name : graph
Class Name : GraphTest

```
GraphList grList = new GraphList(6);
grList.insertEdge(1, 5);
grList.insertEdge(4, 5);
grList.insertEdge(3, 5);
grList.insertEdge(3, 4);
grList.insertEdge(2, 4);
grList.insertEdge(2, 3);
grList.insertEdge(1, 3);
grList.insertEdge(1, 2);
grList.insertEdge(0, 2);
grList.insertEdge(0, 1);

System.out.println();

for(int i = 0; i < 6; i++) {
    adj = grList.adjacency(i);
    System.out.print(i + ": ");
    for(int e: adj) {
        System.out.print(e + " ");
    }
    System.out.println();
}

grList.bfs(1);
System.out.println();
grList.dfs(1);
System.out.println();
```

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

과제 제출 방법



- 프로젝트 폴더를 압축하여 제출

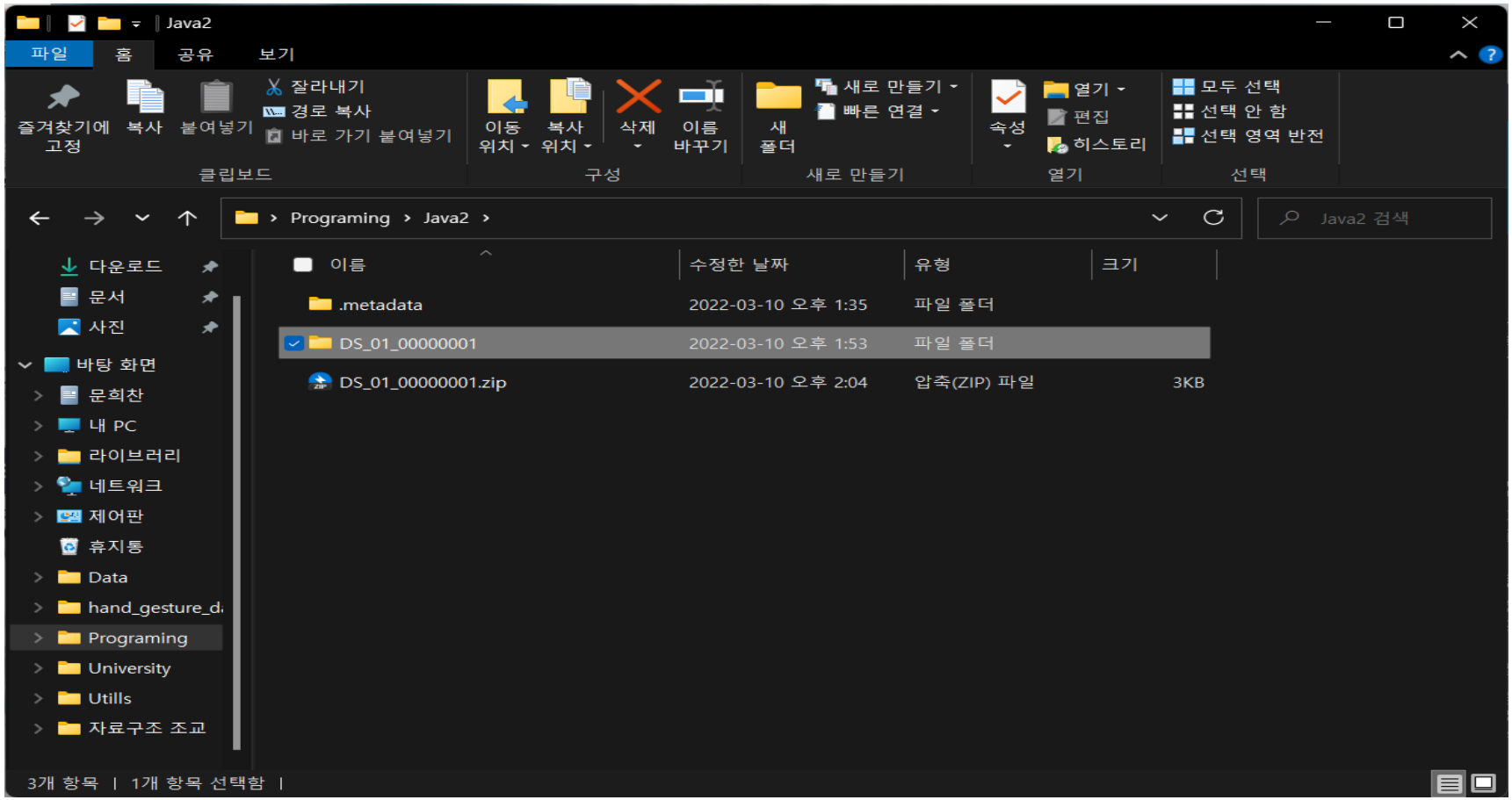
- 프로젝트이름 : AL_(주차)_(학번)

예) AL_07_00000000

- *.java파일만 제출하면 안됩니다.

제출양식을 반드시 지켜주세요

과제 제출 방법



- 반드시 **프로젝트 폴더를 압축**하여 제출