



알고리즘 3주차

이진트리 2

MMC 연구실

박사 과정 문희찬

조교 소개

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



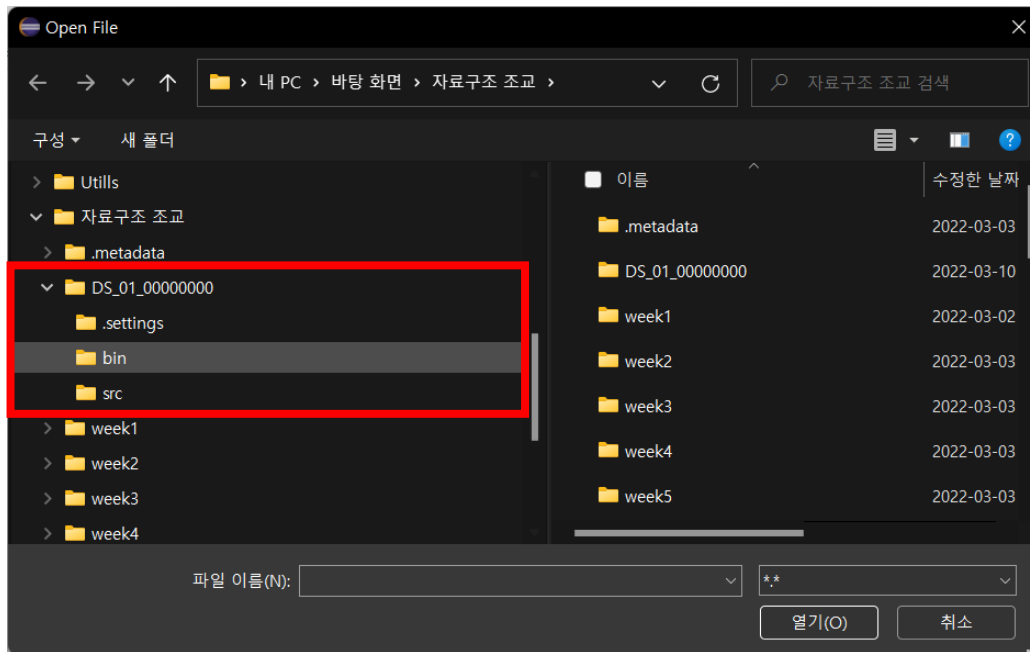
실습 수업 진행 방식

- 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_03_00000000

- *.java파일만 제출하면 안됩니다.

- 제출양식을 반드시 지켜주세요!

확인문제 (binarytree)

Package Name : binarytree

Class Name : BinaryTree

```
private void theInorder(BTNode t) {  
    // 재귀 알고리즘을 사용하여 중위 순회 구현  
}  
  
private void thePreorder(BTNode t) {  
    // 재귀 알고리즘을 사용하여 전위 순회 구현  
}  
  
private void thePostorder(BTNode t) {  
    // 재귀 알고리즘을 사용하여 후위 순회 구현  
}
```

확인문제 (binarytree)

Package Name : binarytree
Class Name : BinaryTreeTest

```
public static void main(String args[]) {  
    BinaryTree btree;  
    BinaryTree ltree;  
    BinaryTree rtree;  
    BinaryTree current;  
  
    rtree = new BinaryTree(new BinaryTree(), "H", new BinaryTree());  
    btree = new BinaryTree(new BinaryTree(), "G", rtree);  
    ltree = btree;  
    btree = new BinaryTree(ltree, "E", new BinaryTree());  
    ltree = new BinaryTree(new BinaryTree(), "D", new BinaryTree());  
    rtree = btree;  
    btree = new BinaryTree(ltree, "B", rtree);  
    ltree = new BinaryTree(new BinaryTree(), "F", new BinaryTree());  
    current = new BinaryTree(ltree, "C", new BinaryTree());  
    ltree = btree;  
    rtree = current;  
    btree = new BinaryTree(ltree, "A", rtree);  
  
    System.out.println(btree.rootData());  
    System.out.println(btree.leftSubTree().rootData());  
    System.out.println(btree.rightSubTree().rootData());  
    System.out.println(btree.leftSubTree().leftSubTree().rootData());  
    System.out.println(btree.rightSubTree().leftSubTree().rootData());  
  
    btree.inorder();  
    System.out.println();  
  
    btree.preorder();  
    System.out.println();  
  
    btree.postorder();  
    System.out.println();  
}
```

<terminated> BinaryTreeTest (2)

A
B
C
D
F

[Inorder]
DBGHEAFC

[Preorder]
ABDEGHCF

[Postorder]
DHGEBFCA

실습 과제

1. 전위, 중위 순회 반복문 구현
2. Level order 구현
3. 트리 복사 및 비교



과제 1

Package Name : binarytree

Class Name : BinaryTree

```
public void inorder() {
    System.out.println("[Inorder]");
    theInorder(root);
    System.out.println();

    // 아래 내용은 실습 과제입니다. (주석 해제 후 inorderIter 메소드에 코드 작성)
    /*
    System.out.println("[InorderIter]");
    inorderIter();
    System.out.println();
    */
}

public void preorder() {
    System.out.println("[Preorder]");
    thePreorder(root);
    System.out.println();

    // 아래 내용은 실습 과제입니다. (주석 해제 후 preorderIter 메소드에 코드 작성)
    /*
    System.out.println("[PreorderIter]");
    preorderIter();
    System.out.println();
    */
}
```

```
private void inorderIter() {
    Stack s = new Stack();
    BTNode p = root;

    // 스택과 반복문을 사용하여 중위 순회 구현
}

private void preorderIter() {
    Stack s = new Stack();
    BTNode p = root;

    // 스택과 반복문을 사용하여 전위 순회 구현
}
```

과제 2

Package Name : binarytree

Class Name : BinaryTree

```
public void levelorder() {  
    BTreeNode p;  
    Queue q = new Queue();  
    System.out.println("[Level order]");  
  
    // 큐와 반복문을 사용하여 레벨 순회 구현  
  
    System.out.println();  
}
```



과제 3

Package Name : binarytree

Class Name : BinaryTree

```
public BinaryTree copy() {
    BinaryTree newTree = new BinaryTree();
    newTree.root = theCopy(root);
    return newTree;
}

private BTreeNode theCopy(BTreeNode t) {
    // 재귀 알고리즘을 사용하여 트리 복사 구현
}

public boolean equals(BinaryTree tr) {
    return theEqual(this.root, tr.root);
}

private boolean theEqual(BTreeNode s, BTreeNode t) {
    // 재귀 알고리즘을 사용하여 트리 비교 구현
    // 같은 트리일 경우 True 반환
}
```



실행 결과

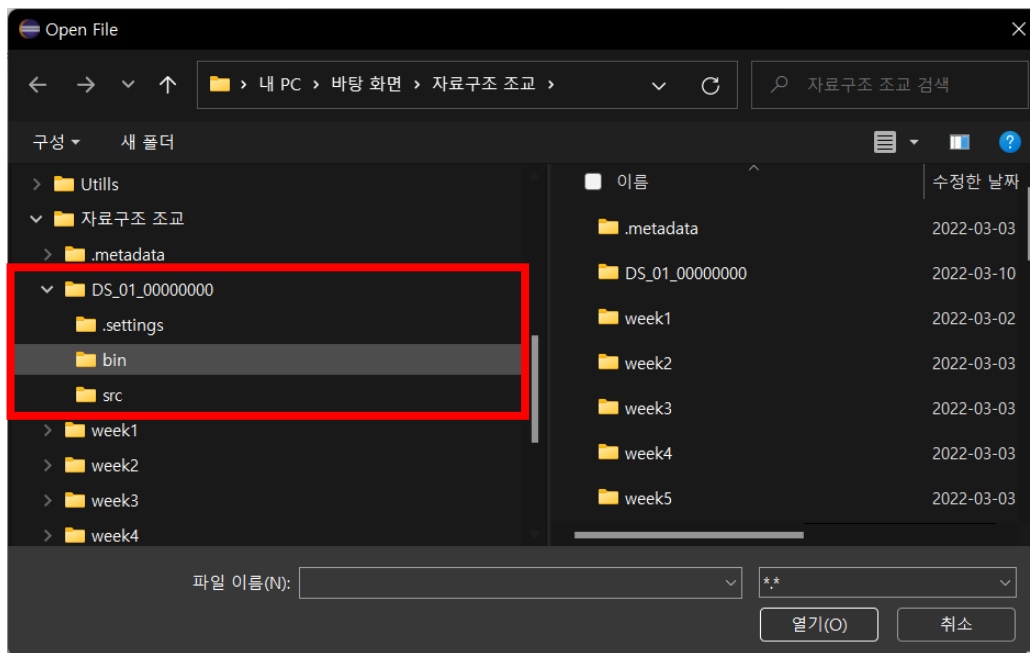
Package Name : binarytree
Class Name : BinaryTreeTest

```
<terminated> BinaryTreeTest (2) [Java Application]
A                                     [Level order]
B                                     ABCDEFGH
C
D                                     <Cloned tree>
F                                     [Inorder]
[Inorder]                           DBGHEAFC
DBGHEAFC                           [InorderIter]
[InorderIter]                       DBGHEAFC
DBGHEAFC

                                     [Case 1]
[Preorder]                           They are the same trees
ABDEGHCF                             [Case 2]
[PreorderIter]                       They are different trees
ABDEGHCF

[Postorder]
DHGEBFCA
```

과제 제출 방법



- 프로젝트 폴더를 압축하여 제출

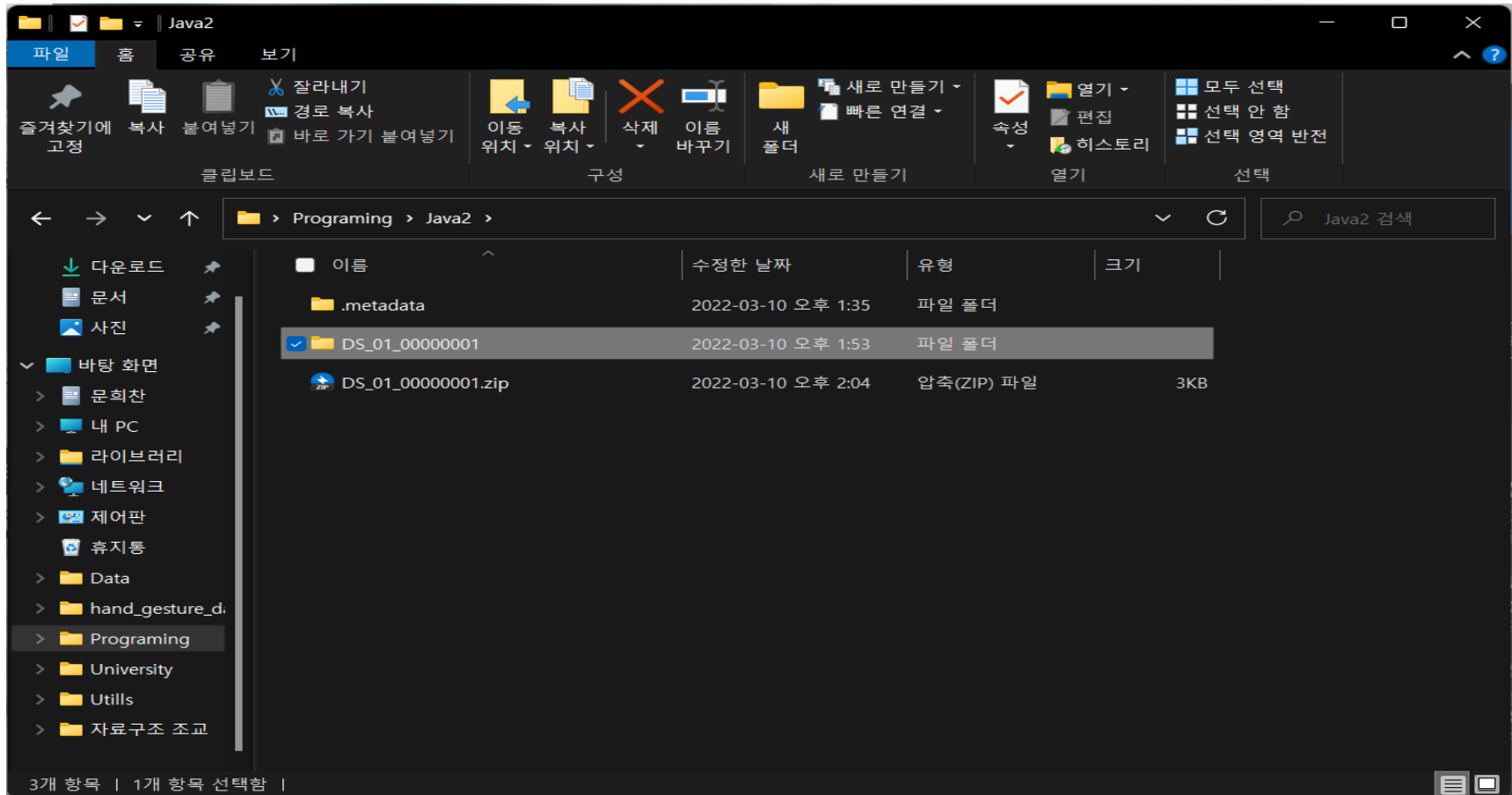
- 프로젝트이름 : AL_(주차)_(학번)

예) AL_01_00000000

- *.java파일만 제출하면 안됩니다.

제출양식을 반드시 지켜주세요

과제 제출 방법



- 반드시 **프로젝트 폴더를 압축**하여 제출