

## Database Systems

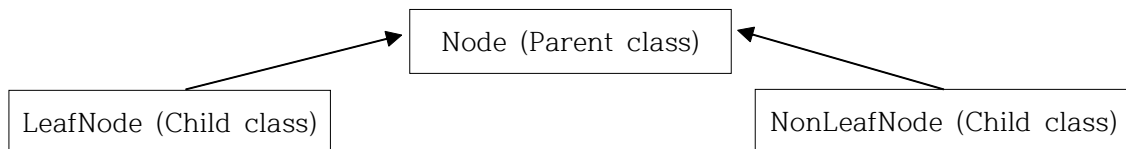
### [B+ tree implementation Assignment]

2019031776 윤건우

#### Summary

구현 언어 : JAVA

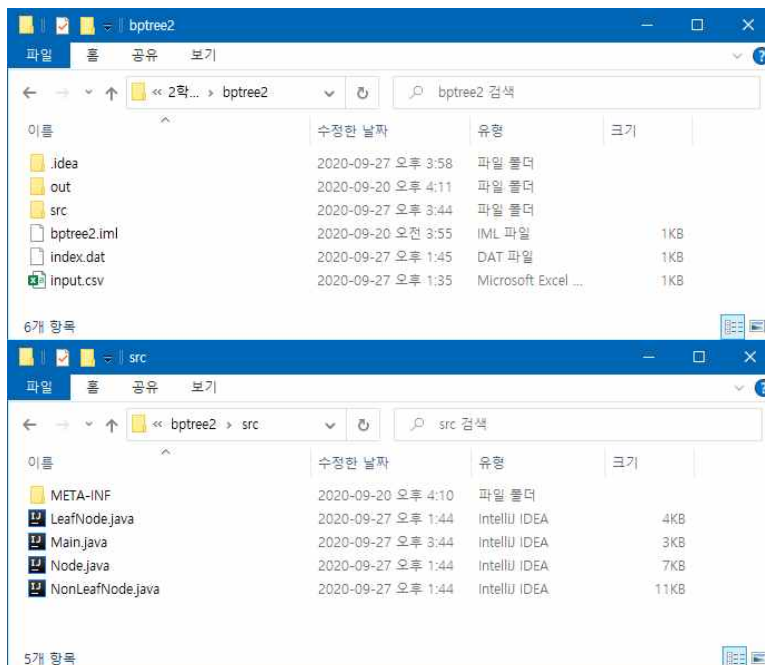
B+ Tree 저장 클래스 :



기능 구현 클래스 : Main class

Main에서 만든 Node 로부터 시작하여 각 종류에 따른 노드들에 대한 기능들은 LeafNode 클래스와 NonLeafNode 클래스에 각각 함수들이 구현되어 있고 공통적인 기능인 Insert, Delete, Search에 대한 함수들은 Node 클래스에서 recursion 함수를 이용하여 해당 노드가 어떤 노드인지에 따라 적절한 자식 클래스의 함수가 실행되도록 함.

IntelliJ IDEA 2020.2 버전에서 컴파일함.



## FILE I/O 형식

index.dat

저장 형식 : 첫 줄에 각 노드의 크기 정보 (ex. 3)

두 번째 줄부터 root 노드부터 시작해 순차적으로 순회하며 각 노드 정보를 저장.

- 찍어쓰기로 정보 구분
- NonLeafNode일 경우 - 0 들어있는 key 개수 firstKey secondKey ... lastKey
- LeafNode일 경우 - 1 들어있는 key 개수 firstKey firstValue ... lastKey lastValue

### class Node

< 변수 >

protected int size;

- 노드의 최대 size를 저장하는 변수

protected int keyNum;

- 노드에 현재 들어있는 key의 개수를 저장하는 변수

protected int[] keys;

- 노드에 현재 들어있는 key들을 저장하는 변수

< 함수 >

public void write(PrintWriter printWriter) throws IOException

- 현재 B+ Tree에 저장된 정보들을 printWriter를 통해 파일에 출력하는 재귀함수
- 현재 노드가 Leaf일 경우 this.writeLeaf(printWriter) 를 실행
- 현재 노드가 NonLeaf일 경우 this.writeNonLeaf(printWriter) 를 실행

public boolean singleSearch(int key)

- 있으면 true를, 없으면 false를 return
- 현재 노드가 Leaf일 경우 LeafNode의 find() 함수를 실행
- 현재 노드가 NonLeaf일 경우 NonLeafNode의 find() 함수를 실행
- 발견 되지 않았다면 NOT FOUND 출력

public boolean insertSearch(int key)

- 기능은 singleSearch와 같으나 NOT FOUND 등의 정보를 콘솔에 출력하지 않고 insert 함수에서 추가하고자 하는 key가 이미 존재하는지 존재하지 않는지를 알려주기 위한 함수