

자료구조 (Data Structure) 실습 과제

Search Trees

2024학년도 1학기

Instructor: Prof. Young-guk Ha
Dept. of Computer Science & Engineering



실습 1. 지난 실습에서 구현한 MyBinTree 클래스를 상속받아서 MyBST 클래스를 구현하시오

- 구현 조건

- MyBinTree 클래스를 상속받아서 오른쪽 예제에 있는 MyBST 클래스를 완성하시오

- 단, 각 노드에는 엔트리 대신 Integer 타입의 키값만 저장됨

- 메소드 설명

- nextNode(v): Inorder 순서상 v 노드의 다음번 노드를 반환
- find(k): k 값을 갖는 키 객체를 탐색
- findAll(k): 동일한 k 값을 갖는 키 객체를 모두 탐색
- insert(k): 새로운 키 객체 k 를 추가
- remove(k): 키 객체값이 k 인 노드를 삭제

```
// Class MyBST
```

```
import java.util.ArrayList;
```

```
public class MyBST extends MyBinTree {
```

```
    // Constructor
```

```
    MyBST() { super(); }
```

```
    MyBST(Object e) { super(e); }
```

```
    // Implement the following methods
```

```
    private MyBinNode nextNode(MyBinNode v) { ... }
```

```
    public Object find(Object k) { ... }
```

```
    public ArrayList findAll(Object k) { ... }
```

```
    public Object insert(Object k) { ... }
```

```
    public Object remove(Object k) { ... }
```

```
}
```

실습 2. 앞의 실습에서 구현한 MyBST를 이용하여 다음의 코드를 작성하시오

- 다음을 수행하는 main() 함수를 작성하시오

- 1) MyBST 클래스를 이용하여 오른쪽 그림의 구조와 같은 BST 객체를 생성하시오
(각각의 키 객체는 **insert** 메소드를 이용하여 추가할 것)
- 2) 다음 메소드를 순서대로 실행하고
실행 결과 및 반환값을 출력하시오
 - ① find(new Integer(8))
 - ② find(new Integer(3))
 - ③ insert(new Integer(3))
 - ④ insert(new Integer(7))
 - ⑤ insert(new Integer(9))
 - ⑥ remove(new Integer(1))
 - ⑦ remove(new Integer(4))
 - ⑧ remove(new Integer(6))
 - ⑨ find(new Integer(3))
 - ⑩ find(new Integer(6))
 - ⑪ findAll(new Integer(9))
- 3) 상속받은 inorder 메소드를 사용하여
최종 BST의 모든 키 값을 출력하시오

