

자료구조 (Data Structure) 실습 과제

# 과제 1: Trees

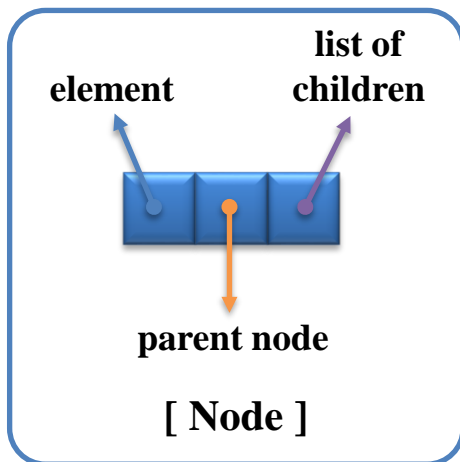
2024학년도 1학기

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



# 실습 1. 다음의 Tree Node 구조 및 Java 코드를 참조하여 MyNode 클래스를 완성하십시오

- A *node* is represented by an object storing
  - Element
  - Link to the parent node
  - Link to the list of child nodes



```
// ※ 참조: java.util.ArrayList 클래스를 사용함
import java.util.ArrayList;

public class MyNode {
    private Object element;
    private MyNode parent;
    private ArrayList children;

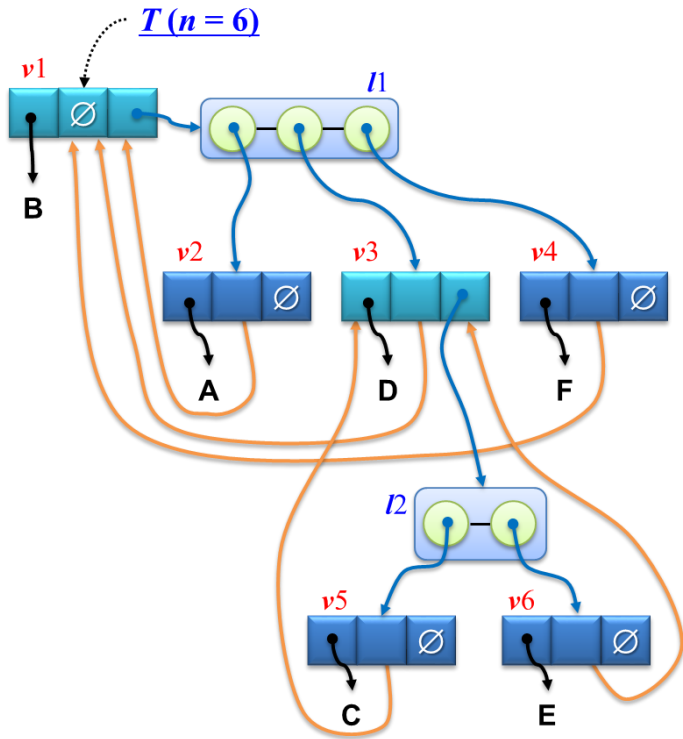
    // implement constructors
    MyNode() {
        element = null;
        parent = null;
        children = null;
    }
    MyNode(Object e) { element = e; ... }

    // implement the following methods
    public Object element() { ... }
    public MyNode parent() { ... }
    public ArrayList children() { ... }
    public int degree() { ... }
    public void setElement(Object e) { ... }
    public void setParent(MyNode p) { ... }
    public void setChildren(ArrayList c) { ... }
}
```

## 실습 2. 앞에서 구현한 MyNode 클래스를 이용하여

아래 예와 같이 **MyTree** 클래스를 **Java**로 구현하시오

[ MyTree로 생성한 Tree 구조의 예 ]



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

```
public class MyTree {
    private MyNode root;
```

```
// implement constructors
```

```
MyTree() { root = null; }
```

```
MyTree (Object e) {
```

```
root = new MyNode (e);
```

```
root.setChildren(new ArrayList());
```

}

```
// implement the following methods
```

```
public int size() { ... }
```

```
public MyNode root() { ... }
```

```
public ArrayList children(MyNode v) { ... }
```

```
public boolean isExternal(MyNode v) { ... }
```

```
public MyNode addRoot(Object e) { ... }
```

```
public MyNode addNode(Object e) { ... }
```

```
public MyNode addChild(MyNode v, Object e) { ... }
```

```
public MyNode addChild(MyNode v, int i, Object e)
```

$$\{ \dots \}$$

```
public MyNode setChild(MyNode v, int i, Object e)
```

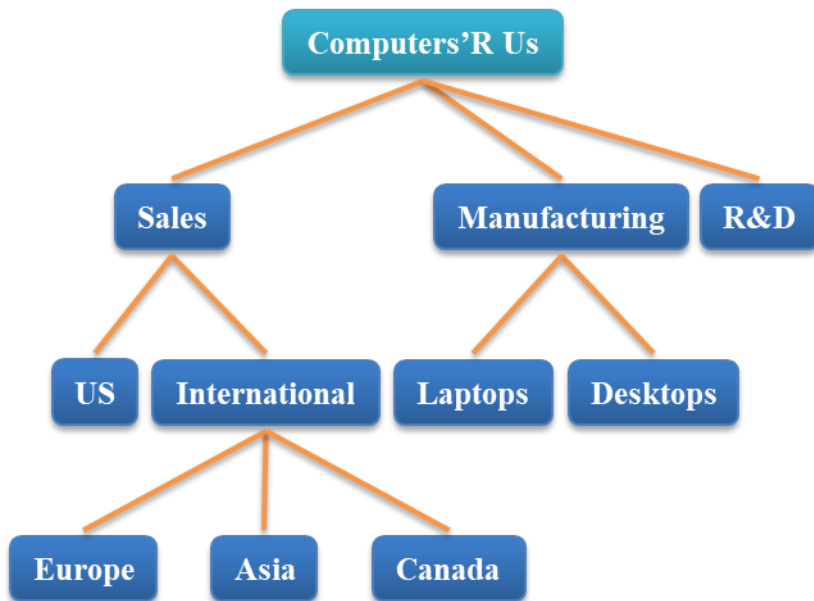
$$\{ \dots \}$$

```
public MyNode removeChild(MyNode v, int i) { ... }
```

}

### 실습 3. 앞에서 구현한 MyTree 클래스를 이용하여 다음의 실습 문제를 Java로 구현하시오

- 아래 그림과 같은 구조의 Tree를 생성하시오
  - addRoot 및 addNode를 사용하여 새로운 Node를 추가해 나가는 과정에서 생성되는 Node 객체의 Element 값을 Level 별로 출력하시오 (오른쪽 화면 출력에 참조)
  - 마지막에 최종 Tree의 크기(총 Node의 개수)를 출력하시오



화면  
출력



[Level 0]  
Computers'R Us

[Level 1]  
Sales, Manufacturing, R&D

[Level 2]  
US, International, Laptops, Desktops

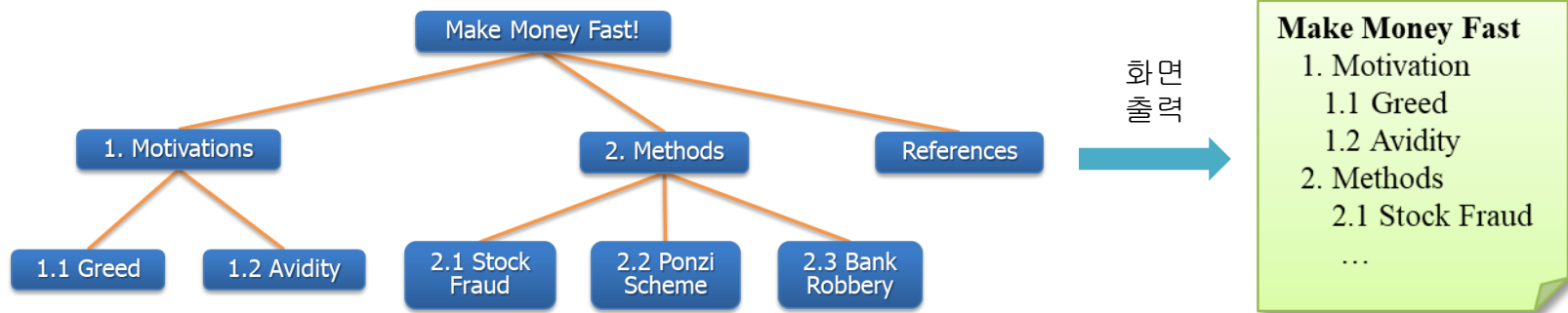
[Level 4]  
Europe, Asia, Canada

\* Tree size = Total 11 Nodes

## 실습 4. 앞에서 구현한 MyTree 클래스를 이용하여 다음의 문제를 Java로 구현하시오

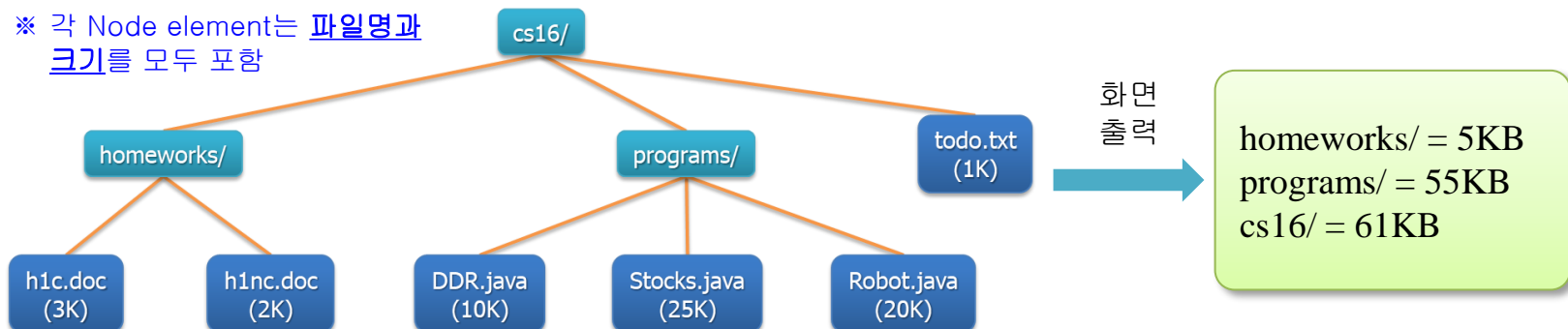
- 1) 아래 그림과 같은 구조의 Tree를 생성하고 Preorder Traversal을 이용하여 오른쪽 그림과 같이 출력하는 Java 프로그램을 작성하시오

※ 줄간 Indentation 주의



- 2) 아래 그림과 같은 구조의 Tree를 생성하고 Postorder Traversal을 이용하여 오른쪽 그림과 같이 출력하는 Java 프로그램을 작성하시오

※ 각 Node element는 파일명과 크기를 모두 포함



자료구조 (Data Structure) 실습 과제

# 과제 2: Binary Trees

2024학년도 1학기

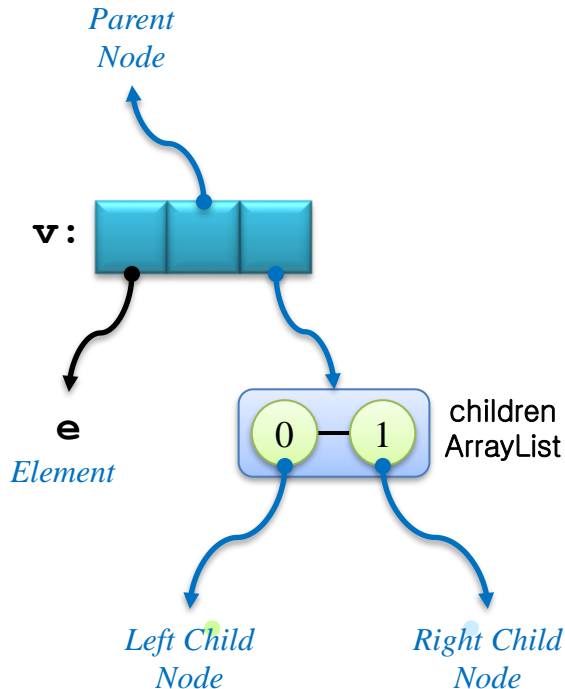
Instructor: Prof. Young-guk Ha  
Dept. of Computer Science & Engineering



# 실습 1. MyNode 클래스를 상속하여 MyBinNode 클래스를 완성하시오 (※ 과제 1번의 MyNode.java 사용)

## [ MyBinNode 구조 ]

MyBinNode **v** = new MyBinNode(**e**);  
실행시에 아래 구조를 생성함



```
public class MyBinNode extends MyNode {  
    // constructors  
    MyBinNode() {  
        super();  
    }  
    MyBinNode(Object e) {  
        super(e);  
    }  
  
    // implement the following methods  
    public MyBinNode left() { ... }  
    public MyBinNode right() { ... }  
    public void setLeft(MyBinNode v) { ... }  
    public void setRight(MyBinNode v) { ... }  
}
```

## 실습 2. MyTree 클래스를 상속하여 MyBinTree 클래스를 완성하시오 (※ 과제 1번의 MyTree.java 사용)

```
import java.util.ArrayList;

public class MyBinTree extends MyTree{
    // constructors
    MyBinTree() { super(); }
    MyBinTree(Object e) { super(e); }

    // implement the following methods
    public Boolean isEmpty() { ... }
    public boolean isRoot(MyBinNode v) { ... }
    public boolean isInternal(MyBinNode v) { ... }
    public boolean isExternal(MyBinNode v) { ... }
    public MyBinNode root() { ... }
    public MyBinNode parent(MyBinNode v) { ... }
    public MyBinNode left(MyBinNode v) { ... }
    public MyBinNode right(MyBinNode v) { ... }
    public boolean hasLeft(MyBinNode v) { ... }
    public boolean hasRight(MyBinNode v) { ... }
    public MyBinNode addRoot(Object e) { ... }
    public MyBinNode addNode(Object e) { ... }
    public MyBinNode insertLeft(MyBinNode v, Object e) { ... }
    public MyBinNode insertRight(MyBinNode v, Object e) { ... }
    public Object replace(MyBinNode v, Object e) { ... }
    public MyBinNode remove(MyBinNode v) throws TwoChildrenException { ... }
    public void attach(MyBinNode v, MyBinNode t1, MyBinNode t2)
        throws NotExternalException { ... }
}
```

```
// inherits from Class Exception
public class TwoChildrenException
    extends Exception {    }

// inherits from Class Exception
public class NotExternalException
    extends Exception {    }
```



### 실습 3. 앞에서 구현한 MyBinTree 클래스를 이용하여 다음의 실습 문제를 Java로 구현하시오

- 1) MyBinTree 및 MyBinNode 클래스를 사용하여 아래 그림과 같은 수식을 Binary Tree로 생성하시오
- 2) Inorder Traverse를 사용하여 Tree가 나타내는 수식을 아래 화면과 같이 출력하시오
- 3) Postorder Traverse를 사용하여 수식을 계산하고 그 결과를 아래 화면과 같이 출력하시오

❖ 상기 1), 2), 3)번 모두 하나의 Java 프로그램으로 구현할 것

