

# 2019년 프로그래밍 언어 개론

- HW 05 -

제출일자	2019.04.12.
이름	장수훈
학번	201402414
분반	01

## 1. 과제 해결 방법

### - public static int max(Node node)

최댓값을 리턴 하는 함수이다. 주어진 jar 파일을 value 와 next 값 중 큰 값을 리턴 한다. 먼저 instanceof 연산자를 이용하여 참조변수가 참조하고 있는 인스턴스의 실제 타입을 알아보고 형 변환이 가능한지 확인한다. 참조변수인 node가 ListNode 클래스로 형 변환이 가능하면 다시 조건을 건다. 재귀를 이용하여 노드의 다음 노드가 이전노드보다 작을 경우 그 노드를 max로 잡고 아닐 경우 다음 노드를 최대로 잡고 리턴을 한다. 이와 마찬가지로 IntNode로 형 변환이 가능한지 보고 전과같이 비교를 한다. 차이점이라고 하면 ListNode의 value값은 node이고 IntNode의 value값은 int형이라는 것이다. 예외 입력은 들어오지 않는다고 가정하였기에 그냥 -201402414를 리턴 하였다(내 학번..).

### - public static int sum(Node node)

max함수와 구조적으로 상당히 유사하다. 노드의 다음 값이 null이 될 때까지 노드의 value값을 더한다. 노드의 value의 총 합을 반환하고 value와 next의 총 합을 리턴 한다.

### - public static void main(String... args)

jar 파일에 포함되어 있는 함수로 반환값는 노드인 TreeFactory.createTree를 사용하여 노드에 입력값을 저장하고 위에 구현한 max 와 sum을 사용하여 출력을 한다.

## 2. 느낀점

Nod, ListNode, IntNode와 제일 어려워 보이는 TreeFactory까지 전부 구현되었는걸 import해서 무난하게 했던 것 같다. 과제수행시간은 3~4시간 걸렸다.

## 3. 결과 수행 화면

```
public static void main(String... args) {
    System.out.println("test1");
    Node node = TreeFactory.createTree("( ( 3 ( ( 10 ) ) 6 ) 4 1 ( ) -2 ( ) )");
    // 이하 결과를 출력하도록 작성
    System.out.println("최대값 : " + max(node));
    System.out.println("총 합 : " + sum(node));

    System.out.println("test2");
    Node node1 = TreeFactory.createTree("( ( 20 ( ( -14 ) ) 0 ) 2 41 ( ) 4 ( ) )");
    System.out.println("최대값 : " + max(node1));
    System.out.println("총 합 : " + sum(node1));

    System.out.println("test3");
    Node node2 = TreeFactory.createTree("( ( 3 2 ) -378 ( ) )");
    System.out.println("최대값 : " + max(node2));
    System.out.println("총 합 : " + sum(node2));
}
```

Console

<terminated> Test1  
test1  
최대값 : 10  
총 합 : 22  
test2  
최대값 : 41  
총 합 : 53  
test3  
최대값 : 3  
총 합 : -373

1. pdf에 있던 main함수 예제1 test

2. 내 학번 test

3. pdf에 있던 문제쪽 문자열 test