
프로그래밍언어 개론 보고서

[06] List Max, Sum

제출일	2017/07/03
학 번	201000287
소 속	일어일문학과
이 름	유다훈

1 과제문제

1 주어진 코드에서 함수 구현

- int max(Node node)
 - 최대값을 반환하는 메소드
 - Recursion 하게 작성
- int sum(Node node)
 - 노드의 총합을 반환하는 메소드
 - Recursion 하게 작성

2 과제 해결 방법

- 문자열을 받아 트리를 구성하고, 각 노드를 방문하여 총합과 최대값을 구하는 본 프로그램을 이해한 후 작성한다.
- int max(Node node)

```
public static int max(Node node) {
    //최대값을 리턴하도록 작성, recursion으로 짜시오.
    //value와 next 값 중 큰 값을 리턴
    int maxValue=Integer.MIN_VALUE;

    if(node instanceof ListNode) {
        ListNode listNode = (ListNode)node;
        if(listNode.value != null)
            maxValue = Math.max(maxValue, max(listNode.value));
    }
    if(node instanceof IntNode) { //type check;
        IntNode intNode = (IntNode)node;

        maxValue = Math.max(maxValue, intNode.value);
    }
    if(node.getNext() != null) {
        maxValue = Math.max(maxValue, max(node.getNext()));
    }

    return maxValue;
}
```

- 최대값을 저장할 변수를 저장해주고, Java 에서 저장할 수 있는 가장 최소값을 저장해주어 최대값을 비교할 시에 잘 작동할 수 있도록 초기화를 해준다.
- 전달받은 노드의 타입검사를 해준다.
- 만일 ListNode 타입의 노드라면, 최대값 maxvalue 와 ListNode 가 가지고 있는 노드의 값을 재귀적으로 구해준 후, Math.max()함수로 비교를 하여 최대값을 저장한다.
- 만일 IntNode타입의 노드라면, 현재 노드의 값과 maxvalue 값을 비교하여 최대값을 구한다.
- 해당 노드의 다음 노드가 존재한다면, 현재 maxvalue 값과 다음 노드의 값을 비교하여 최대값을 저장해준다.
- int sum(Node node)

```
public static int sum(Node node) {
    //노드 value의 총합을 반환
    //value와 next의 총 합을 리턴하면됨
    //sum_result 값 초기화
    int sum_result = 0;

    // node의 타입 확인
    if (node instanceof ListNode) {
        // listnode 일 경우
        ListNode listNode = (ListNode) node;
        // listnode안의 값이 있을 경우
        if(listNode.value != null) {
            sum_result += sum(listNode.value);
            // sum_result 값 갱신
        }
    }
    if(node instanceof IntNode) {
        // node 타입 확인
        // intnode일 경우
        // sum_result 은 해당 노드의 value.
        IntNode intNode = (IntNode)node;
        sum_result = intNode.value;
    }
    // 다음 노드의 값이 있을 경우
    if(node.getNext() != null) {
        sum_result += sum(node.getNext());
        // recursion으로 계속 실행
    }
    return sum_result;
    // sum_result 값 return
}
```

- 노드들의 총합을 저장할 수 있는 변수 sum_result 를 선언한다.
- 전달받은 노드가 ListNode 타입의 노드일 경우, ListNode 의 노드값을 체크해준다.

- 노드값이 있다면, sum_result 에 ListNode 가 가진 Node 를
파라미터로, 재귀적으로 sum()메소드를 실행해준다.
- 전달받은 노드가 IntNode 타입의 노드일 경우, intNode 가 가진
값을 sum_result 에 저장해준다.
- 지금 검사를 하고있는 노드의 다음값이 있다면,
- 다음값을 파라미터로하여 sum()메소드를 재귀적으로 호출하여
그 값을 sum_result 에 더하여 저장해준다.

3 과제를 해결하면서 느낀 점

- 1 작성해놓고 보니 간단한 구조의 메소드인데, ListNode 타입과 IntNode 타입을
구별해야하는 과정이 어렵게 느껴졌다.
- 2 여러 곳에서 함수를 재귀적으로 호출하다보니 이해하는데 난해했다.
- 3 어떠한 클래스에서 생성된 객체인지 확인하고자 할 때, instanceof 라는
keyword 를 쓰면 간단하게 알 수 있는 방법을 알게 되었다.
- 4 Math 클래스의 max()함수를 쓰게되어서 조금이나마 코드를 줄일 수 있었다.
- 5 jar 파일을 생성하고, 그것을 실행시키는 연습문제가 있었는데, Mac 의
Terminal에서는 실행이 되질 않아 Mac에서는 어떻게 실행하는지 궁금하다.