

# 자료구조

05

과목명 : 자료구조

분반 : 06

담당 교수 : 박정희

학과 : 컴퓨터융합학부

학번 : 202002546

이름 : 임우진

## 목차

1. 코드
2. 풀이 과정 및 코드 설명
3. 실행 화면 캡처
4. 요구 및 건의 사항

# 1. 코드 : MainJava, SinglyLinkedList

```
© MainJava.java × input.txt © SinglyLinkedList.java

1  import java.io.*;
   0개의 사용 위치
2  ▶ public class MainJava {
   0개의 사용 위치
3  ▶   public static void main(String[] args) throws Exception{
4       File file = new File( pathname: "C:\\Users\\8146q\\IdeaProjects\\DS_05_202002546\\src\\input.txt");
5       FileReader fr = new FileReader(file);
6       BufferedReader br = new BufferedReader(fr);
7
8       SinglyLinkedList list1 = new SinglyLinkedList(); //단순 연결 리스트 객체 생성
9       SinglyLinkedList list2 = new SinglyLinkedList();
10
11
12       String s1 = br.readLine(); //큰 정수 문자열로 받아서 정수로 바꾸고 list1 객체에 삽입 (list2도 동일)
13       int len1 = s1.length()-3;
14       int[] arr1 = new int[len1];
15       for(int i = 0; i<len1;i++){
16           arr1[i] = s1.charAt(i)-48;
17           list1.insertNumber(arr1[i]);
18       }
19       list1.printNumber();
20
21
22       String s2 = br.readLine();
23       int len2 = s2.length();
24       int[] arr2 = new int[len2];
25       for(int i = 0; i<len2;i++){
26           arr2[i] = s2.charAt(i)-48;
27           list2.insertNumber(arr2[i]);
28       }
29       list2.printNumber();
30
31
32       list1.add(list2);
33       list1.printNumber();
34
35   }
36 }
```

```
© MainJava.java × input.txt © SinglyLinkedList.java ×

9개의 사용 위치
33 Node first = null; //맨 앞 노드 초기화
34
2개의 사용 위치
35 void insertNumber(int x) { //삽입 메서드 : 새로운 노드 만들고 비어있는지 확인하고 삽입
36     Node newNode = new Node(x);
37     if (first == null) {
38         first = newNode;
39         return;
40     }
41     newNode.next = first;
42     first = newNode;
43
44 }
45 }
```

17개 사용 위치

```
46 @ void add(SinglyLinkedList sec) { //각 list에 저장되어있는 큰 두 정수를 더하고 list1에 저장하는 메서드
47     Node current1 = this.first; //list1 순회
48     Node current2 = sec.first; //list2 순회
49     Node result = null;
50     Node previous = null;
51     int carry = 0; //캐리값
52
53     while (current1 != null && current2 != null) {
54         int sum = current1.data + current2.data + carry;
55         carry = sum / 10;
56         sum = sum % 10;
57
58         Node node = new Node(sum);
59
60         if (result == null) {
61             result = node;
62             previous = node;
63         } else {
64             previous.next = node;
65             previous = node;
66         }
67
68         current1 = current1.next;
69         current2 = current2.next;
70     }
71
72
73     if (carry != 0) {
74         Node node = new Node(carry);
75         previous.next = node;
76         previous = node;
77     }
78     this.first = reverseList(result);
79 }
80
```

```

3개 사용 위치
81 void printNumber() { //출력함수/재귀함수를 호출해서 리스트내의 항목을 출력한다
82     if (first == null) {
83         System.out.println();
84         return;
85     }
86     printNumberRecursive(first);
87     System.out.println();
88 }
89
2개 사용 위치
90 void printNumberRecursive(Node current) { //맨앞 노드를 받아와 마지막 노드까지 출력한다
91     if (current == null) {
92         return;
93     }
94     printNumberRecursive(current.next);
95     System.out.print(current.data);
96 }
97
1개 사용 위치
98 Node reverseList(Node first) { //리스트를 뒤집어주는 역할
99     Node previous = null;
100    Node current = first;
101    Node next = null;
102    while (current != null) {
103        next = current.next;
104        current.next = previous;
105        previous = current;
106        current = next;
107    }
108    return previous;
109 }
110 }

```

보고서 작성 후 주석 추가했습니다.

## 2. 풀이 과정 및 코드 설명

SinglyLinkedList.java

### **Node class**

Node 구현에 필요한 필드와 생성자, 메서드를 포함한다.

### **insertNumber()메서드**

삽입 메서드로서 새로운 노드를 생성하고

리스트가 비어있는지 확인하고 리스트에 노드를 삽입한다.

### **add()메서드**

각 리스트에 저장되어 있는 매우 큰 두 정수를 더하고

list1에 저장한다.

각 리스트를 순회하면서 모든 리스트를 순회할 때까지

합과 캐리를 고려하여 각 자리에 값을 삽입한다.

참고로 previous변수는 add()에서 결과값으로 생성한 새 노드들을 연결

하기 위해 이전 노드를 추적하는 변수다.

reverseList메서드를 호출하여 합이 담긴 list1을 뒤집는다.

### **printNumber()메서드**

재귀함수를 호출하여 리스트내의 항목을 출력한다.

### **printNumberRecursive()메서드**

맨 앞 노드를 받아와서 재귀를 사용해서 마지막 노드까지 출력한다.

### **reverseList()메서드**

리스트를 뒤집어 역순으로 만든다.

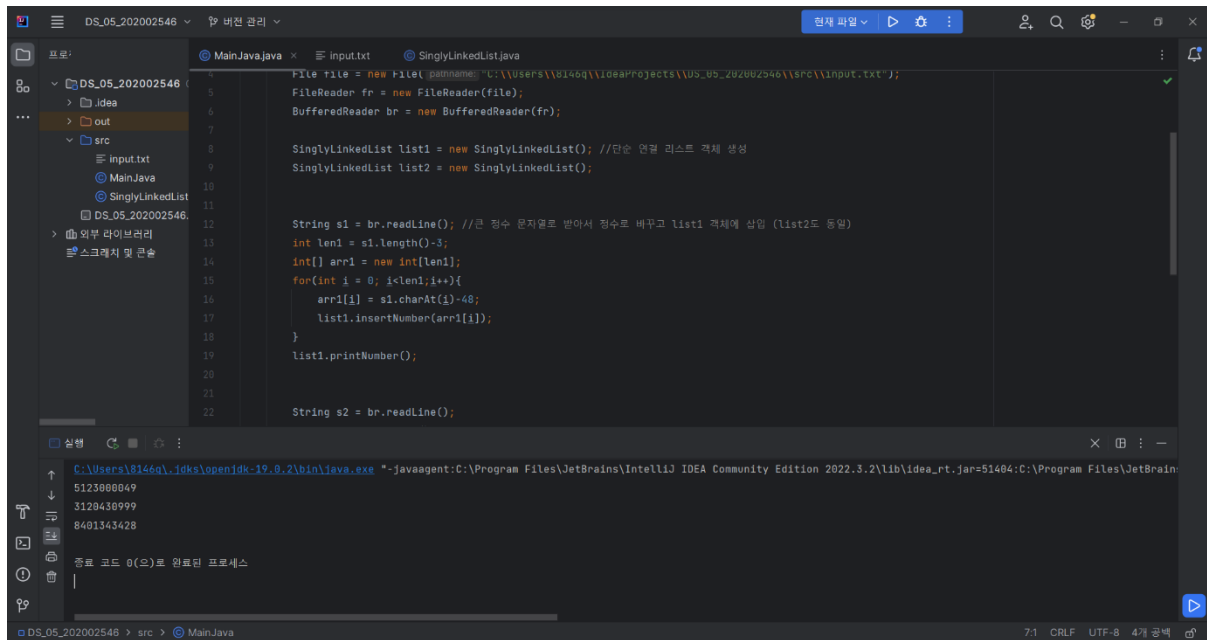
MainJava.java

단순 연결 리스트 객체 2개 생성

큰 정수 문자열로 받아서 정수로 바꾼 후 각 list에 삽입

list1, list2 출력 후 add()로 연산 후 list1 출력

### 3. 실행 화면 캡처



```
File file = new File(filepathname: "C:\\Users\\8146q\\IdeaProjects\\DS_05_202002546\\src\\input.txt");
FileReader fr = new FileReader(file);
BufferedReader br = new BufferedReader(fr);

SinglyLinkedList list1 = new SinglyLinkedList(); //단순 연결 리스트 객체 생성
SinglyLinkedList list2 = new SinglyLinkedList();

String s1 = br.readLine(); //문 문자열로 받아서 정수로 바꾸고 list1 객체에 삽입 (list2도 동일)
int len1 = s1.length()-3;
int[] arr1 = new int[len1];
for(int i = 0, i<len1;i++){
    arr1[i] = s1.charAt(i)-48;
    list1.insertNumber(arr1[i]);
}
list1.printNumber();

String s2 = br.readLine();
```

실행

C:\Users\8146q\Idea\openjdk-19.0.2\bin\java.exe -javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2022.3.2\lib\idea\_rt.jar=51404:C:\Program Files\JetBrain

5123000049  
3120430999  
8401343428

종료 코드 0(으)로 완료된 프로세스

5123000049  
3120430999  
8401343428



#### 4. 요구 및 건의 사항

없습니다.