

[511643] 자료구조 (2019-2 학기)**숙제 #02 (실습 #05 보고서)**

이름	20175105
학번	곽영주
소속 학과/대학	빅데이터
분반	03 (담당교수: 김태운)

<주의사항>

- 개별 과제 입니다. (팀으로 진행하는 과제가 아니며, 모든 학생이 보고서를 제출해야 함)
- **각각의 문제 바로 아래에 답을 작성 후 제출해 주세요.**
 - 소스코드/스크립트 등을 작성 한 경우, 해당 파일의 이름도 적어주세요.
- 스마트캠퍼스 제출 데드라인: **2020. 04. 15. ~ 2020. 04. 21 (화) 23:59**
 - 데드라인을 지나서 제출하면 24 시간 단위로 20%씩 감점(5 일 경과 시 0 점)
 - 주말/휴일/학교행사 등 모든 날짜 카운트 함
 - 부정행위 적발 시, 원본(보여준 사람)과 복사본(베낀 사람) 모두 0 점 처리함
 - 예외 없음
- 스마트캠퍼스에 아래의 파일을 제출 해 주세요
 - 보고서(**PDF 파일로 변환 후 제출**)
 - 보고서 파일명에 이름과 학번을 입력 해 주세요.
 - 소스코드, 스크립트, Makefile 등을 작성해야 하는 경우, 모든 파일 제출 (미 제출시 감점)

<개요>

이번 과제는 연결 리스트(단순 연결 리스트, 이중 연결 리스트, 환형 연결 리스트)에 대한 내용입니다.

<실습 과제>

[Q 0] 요약 [배점: 10]

이번 과제에서 배운 내용 또는 과제 완성을 위해서 무엇을 했는지 2~3 문장으로 요약하세요.

답변: 교재와 구글링을 통해서 연결 리스트의 다양한 소스코드를 참고하면서 연결 리스트를 깊게 공부할 수 있었습니다. 그리고 교수님께 질문을 통해 얻은 접근방식으로 과제를 완성할 수 있었습니다.

[Q 1] 단순 연결 리스트 [15 점]

강의노트에 있는 SList 를 그대로 구현하세요. 강의노트와 동일하게 main 메소드를 구현하고, SList 를 테스트 하세요. 실행 결과도 강의노트와 동일해야 합니다. main 메소드를 실행하고 터미널 출력 결과를 캡처한 뒤 본 문서에 첨부하세요. 소스코드도 첨부파일로 제출하세요.

답변:

소스코드 : [Q1] 단순 연결 리스트 소스코드.txt

```
<terminated> SListMain [Java Application] C:\Program Files\Java\jdk-9.0.4\bin\java.exe
pear    apple    orange    cherry    : s의 길이 = 4

체리가    3번째에 있다.
키위가    -1번째에 있다.

pear    orange    cherry    : s의 길이 = 3

orange    cherry    : s의 길이 = 2

100      200      300      400      500      : t의 길이 = 5
```

[Q 2] 이중 연결 리스트 [15 점]

강의노트에 있는 DList 를 그대로 구현하세요. 강의노트와 동일하게 main 메소드를 구현하고, DList 를 테스트 하세요. 실행 결과도 강의노트와 동일해야 합니다. main 메소드를 실행하고 터미널 출력 결과를 캡처한 뒤 본 문서에 첨부하세요. 소스코드도 첨부파일로 제출하세요.

답변:

소스코드 : [Q2] 이중 연결 리스트 소스코드.txt

```
<terminated> DListMain [Java Application]
apple    pear    orange  cherry

apple    pear    orange

apple    pear    orange  grape

pear     orange  grape
orange   grape
grape
리스트 비어있음
```

[Q 3] 환형 연결 리스트 [15 점]

강의노트에 있는 CList 를 그대로 구현하세요. 강의노트와 동일하게 main 메소드를 구현하고, CList 를 테스트 하세요. 실행 결과도 강의노트와 동일해야 합니다. main 메소드를 실행하고 터미널 출력 결과를 캡처한 뒤 본 문서에 첨부하세요. 소스코드도 첨부파일로 제출하세요.

답변:

소스코드 : [Q3] 환형 연결 리스트 소스코드.txt

```
<terminated> CListMain [Java Application] C:\Program File
apple  orange  cherry  pear    : s의 길이 = 4
orange  cherry  pear    : s의 길이 = 3
```

[Q 4] 단순 연결 리스트, SList [15 점]

SList 에 `public void setAndOrderNodes(int[] arr)` 메소드를 아래와 같이 동작하도록

구현하세요. 가정: SList 의 Node 는 int 형 item 을 저장하고 있습니다.

- 1 IF(arr != null) 리스트에 저장된 모든 노드 삭제; ELSE return;
- 2 인자로 넘어온 arr 배열의 원소를 하나씩 SList 에 노드로 추가. 단, 노드들이 (노드에 저장된 숫자를 기준으로) 오름차순으로 정렬되도록 노드를 추가. 중복된 숫자가 있을 경우 어느 것이 먼저 오든 관계 없음. **인자로 넘어오는 arr 배열을 미리 정렬하면 안됩니다.**

[Task 1: SList slist1] setAndOrderNodes([7,11,4,1,8]) 호출 시, 1-4-7-8-11 와 같이 노드가 연결됩니다. (배열을 미리 정렬하면 안됩니다)

[Task 2: SList slist2] setAndOrderNodes([5,2,8,7,13]) 호출 시, 2-5-7-8-13 와 같이 노드가 연결됩니다. (배열을 미리 정렬하면 안됩니다)

[Task 3: slist1.join(slist2)] 두개의 리스트를 결합(join) 할 경우, 그 결과가 정렬된 상태가 되도록 하는 `public void join (SList slist2)` 메소드를 구현하세요. slist1.join(slist2) 호출 시, slist1 에 join 된 결과를 저장합니다.

구현 결과를 테스트 하기 위한 main 메소드를 만들고, 위의 순서대로 구현결과를 테스트 하세요. SList 를 출력하기 위한 print 메소드를 만들고, [Task 1] 직후, [Task 2] 직후, 그리고 [Task 3] 직후에 print 메소드를 각각 호출하세요. 터미널 출력 결과를 캡처한 뒤 본 문서에 첨부하세요. 소스코드도 첨부파일로 제출하세요.

답변:

소스코드 : [Q4] 단순 정렬 소스코드.txt

```
<terminated> main4 [Java Application] C:\Program Files\Java\jdk-13.0.2\bin\javaw.exe (2020. 4.
[slist1 정렬]
1      4      7      8      11

[slist2 정렬]
2      5      7      8      13

[join 후 정렬]
1      2      4      5      7      7      8      8      11      13
```

[Q 5] 단순 연결 리스트, SList [15 점]

리스트를 역방향으로 바꾸는 `public void reverse()` 메소드를 구현하세요. SList 멤버함수인 `reverse()` 메소드를 호출하면, 저장된 리스트를 역방향으로 바꿉니다. SList 를 출력하기 위한 `print` 메소드를 구현하세요. 가정: SList 의 Node 는 int 형 item 을 저장하고 있습니다.

<사용 예>

[SList s1 리스트] 1-2-3-4-5

[s1.reverse() 결과] 5-4-3-2-1

[Task 1] 1-2-3-4-5 순으로 노드가 저장된 SList s1 을 생성하고 `print()` 메소드를 호출하세요.

[Task 2] `s1.reverse()` 호출하고 `print()` 메소드를 호출하세요.

터미널 출력 결과를 캡처한 뒤 본 문서에 첨부하세요. 소스코드도 첨부파일로 제출하세요.

답변:

소스코드 : [Q5] 단순 역방향 소스코드.txt

```
<terminated> main5 [Java Application] C:\WP
```

```
[역순 전]
```

```
1      2      3      4      5
```

```
[역순 후]
```

```
5      4      3      2      1
```

[Q 6] 이중 연결 리스트, DList [15 점]

두 개 리스트를 더하는 메소드 `public int add(DList d12)`를 구현하세요. DList 를 출력하기 위한 `print` 메소드를 구현하세요. 가정: DList 의 DNode 는 `int` 형 item 을 저장하고 있습니다.

<사용 예>

[d11 리스트] 1-2-3 (= 123 으로 취급)

[d12 리스트] 6-7-8-9 (= 6789 로 취급)

`d11.add(d12)` 또는 `d12.add(d11)` 호출 시, `int` 값인 6912 리턴

[Task 1] 1-2-3 순으로 노드가 저장된 DList d11 을 생성하고 `print()` 메소드를 호출하세요.

[Task 2] 6-7-8-9 순으로 노드가 저장된 DList d11 을 생성하고 `print()` 메소드를 호출하세요.

[Task 3] `int add1 = d11.add(d12); int add2 = d12.add(d11);` 호출 후, `add1` 과 `add2` 에 저장된 값을 터미널에 출력하세요.

터미널 출력 결과를 캡처한 뒤 본 문서에 첨부하세요. 소스코드도 첨부파일로 제출하세요.

답변:

소스코드 : [Q6] 이중 덧셈 소스코드.txt

```
<terminated> main6 [Java Applicat
1      2      3
6      7      8      9

add1 = 6912

add2 = 6912
```

[Q 7] 환형 연결 리스트, CList [15 점]

리스트를 n 번 rotate 하는 `public void rotateLeft(int n), public void rotateRight(int n)` 메소드를 구현하세요. CList 를 출력하기 위한 `print` 메소드도 구현하세요. Rotate 함수는 정해진 방향으로 n 번 리스트를 회전시키는 기능을 합니다. 가정: CList 의 Node 는 int 형 item 을 저장하고 있습니다.

<사용 예>

[CList c11 리스트] 1-2-3-4 ➔ [c11.rotateLeft(1) 결과] 2-3-4-1

[CList c12 리스트] 1-2-3-4 ➔ [c12.rotateRight(1) 결과] 4-1-2-3

[Task 1] 1-2-3-4 순으로 노드가 저장된 CList c11, c12 리스트를 생성하고 `print` 메소드를 각각 호출하세요.

[Task 2] `c11.rotateLeft(1); c12.rotateRight(1);` 을 호출하세요.

[Task 3] c11 과 c12 리스트를 출력하세요.

터미널 출력 결과를 캡처한 뒤 본 문서에 첨부하세요. 소스코드도 첨부파일로 제출하세요.

답변:

소스코드 : [Q7] 환형 로테이션 소스코드.txt


```
<terminated> main7 [Java Applicati
[cl1 list]
1      2      3      4

[cl2 list]
1      2      3      4

[cl1 rotateLeft(1)]
2      3      4      1

[cl2 rotateRight(1)]
4      1      2      3
```

끝! 수고하셨습니다 ☺