

[511643] 자료구조**실습 #06 보고서**

이름	곽영주
학번	20175105
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. 22. ~ 2020. 04. 28. (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 **스택**을 구현하고 스택을 활용하는 내용입니다. 실습과제에 대한 내용은 강의자료에 상세하게 설명되어 있으니 강의자료를 확인해 주세요.

<실습 과제>

[Q 0] 요약 [배점: 10 점]

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

답변: 저번 주 List 과제를 하면서 이해하느라 시간이 오래 걸렸지만 이번 Stack 과제를 풀면서 확실히 List 를 이해하고 풀어서 그런지 복습하는 느낌으로 수월하게 풀 수 있었습니다. 그리고 궁금한 것을 교수님께 질문(이메일)을 했는데 잘 알려주셔서 많이 도움되었습니다.

[Q 1] 배열로 구현한 ArrayStack [15 점]

강의자료와 동일하게 ArrayStack 을 구현하세요. ArrayStack 을 테스트 하기 위한 main 함수도 강의자료와 동일하게 구현하고 실행하세요. 터미널 출력 결과를 캡처하여 본 문서에 첨부하세요. 소스코드도 첨부파일로 제출해야 합니다.

답변:

소스코드 : [Q1] 배열로 구현한 ArrayStack 소스코드.txt

```
<terminated> ArrayStackMain [Java Ap]
cherry
apple  orange  cherry  pear
cherry
apple  orange  cherry  grape
```

[Q 2] 단순 연결 리스트로 구현한 ListStack [15 점]

강의자료와 동일하게 ListStack 을 구현하세요. ListStack 을 테스트 하기 위한 main 함수도 강의자료와 동일하게 구현하고 실행하세요. 터미널 출력 결과를 캡처하여 본 문서에 첨부하세요. 소스코드도 첨부파일로 제출해야 합니다.

답변:

소스코드 : [Q2] 단순 연결 리스트로 구현한 ListStack 소스코드.txt

```
<terminated> ListStackMain [Java Appli
cherry
pear    cherry  orange  apple
cherry
grape   cherry  orange  apple
```

[Q 3] 괄호 짝 맞추기 [15 점]

ArrayStack 또는 ListStack 을 사용해서, 주어진 수식에서 괄호의 짝이 맞는지 여부를 확인하는 `public Boolean checkParentheses(String s)` 메소드를 작성하세요. 짝이 맞으면 `True` 를 리턴하고, 짝이 맞지 않으면 `False` 를 리턴합니다.

[Task 1] `String str1 = “(ant+bee)*crab+{[ant*food]+(offset)}/gas”` 라는 문자열이 있습니다. `checkParentheses` 을 호출하고, `str1` 과 리턴값을 모두 터미널에 출력하세요.

[Task 1] `String str2 = “(ant+bee)*crab+[ant*food]+(offset)}/gas”` 라는 문자열이 있습니다. `checkParentheses` 을 호출하고, `st2` 와 리턴값을 모두 터미널에 출력하세요.

터미널 출력 결과를 캡처하여 본 문서에 첨부하세요. 소스코드도 첨부파일로 제출해야 합니다.

소스코드 : [Q3] 괄호 짝 맞추기 소스코드.txt

답변 1 (Task 1):

```
<terminated> main3 [Java Application] C:\Program F
-- 괄호 짝이 맞으면 true, 안맞으면 false --

(ant+bee)*crab+{[ant*food]+(offset)}/gas
짝이 맞는가? true
```

답변 2 (Task 2):

```
<terminated> main3 [Java Application] C:\Program
-- 괄호 짝이 맞으면 true, 안맞으면 false --

(ant+bee)*crab+[ant*food]+(offset)}/gas
짝이 맞는가? false
```

[Q 4] 회문(Palindrome) 검사하기 [15 점]

ArrayStack 또는 ListStack 을 사용해서, 주어진 단어가 회문인지 여부를 확인하는 public Boolean checkPalindrome(String s) 메소드를 작성하세요. 회문이면 True 를 리턴하고, 아니면 False 를 리턴합니다.

[Task 1] String s1 = “redivider” 일 때, checkPalindrome 을 호출하고, s1 과 리턴값을 모두 터미널에 출력하세요

[Task 2] String s2 = “redder” 일 때, checkPalindrome 을 호출하고, s2 와 리턴값을 모두 터미널에 출력하세요

[Task 3] String s3 = “a” 일 때, checkPalindrome 을 호출하고, s3 와 리턴값을 모두 터미널에 출력하세요

[Task 4] String s4 = “abbd” 일 때, checkPalindrome 을 호출하고, s4 와 리턴값을 모두 터미널에 출력하세요

[Task 5] String s5 = “koxko” 일 때, checkPalindrome 을 호출하고, s5 와 리턴값을

모두 터미널에 출력하세요.

터미널 출력 결과를 캡처하여 본 문서에 첨부하세요. 소스코드도 첨부파일로 제출해야 합니다.

소스코드 : [Q4] 회문(Palindrome) 검사하기 소스코드.txt

답변 1 (Task 1):

```
<terminated> main4 (1) [Java Applicati
    -- 회문 검사 --
    -- 회문이면 true, 아니면 false --

[1]redivider이(가) 회문인가? true
```

답변 2 (Task 2):

```
<terminated> main4 (1) [Java Applic
    -- 회문 검사 --
    -- 회문이면 true, 아니면 false --

[2]redder이(가) 회문인가? true
```

답변 3 (Task 3):

```
<terminated> main4 (1) [Java Applica
    -- 회문 검사 --
    -- 회문이면 true, 아니면 false --

[3]a이(가) 회문인가? true
```

답변 4 (Task 4):

```
<terminated> main4 (1) [Java Applica
    -- 회문 검사 --
-- 회문이면 true, 아니면 false --

[4]abbd이(가) 회문인가? false
```

답변 5 (Task 5):

```
<terminated> main4 (1) [Java Applica
    -- 회문 검사 --
-- 회문이면 true, 아니면 false --

[5]koxko이(가) 회문인가? false
```

[Q 5] 중위표기법 수식을 후위표기법으로 변환 [15 점]

ArrayStack 또는 ListStack 을 사용해서, 주어진 중위표기법 수식 표현을 후위표기법 수식 표현으로 변경하는 `public String getPostEq(String s)` 메소드를 작성하세요. `getPostEq` 메소드는 후위표기법으로 변환한 수식을 `String` 타입으로 리턴합니다.

[Task 1] `String s1 = "A*(B+C/D)"` 일 때, `getPostEq` 를 호출하고, `s1` 과 리턴값을 모두 터미널에 출력하세요.

[Task 2] `String s2 = "(A+B)*(C-D)"` 일 때, `getPostEq` 를 호출하고, `s2` 와 리턴값을 모두 터미널에 출력하세요.

[Task 3] `String s3 = "A+B+C*(B+C*D)/D"` 일 때, `getPostEq` 를 호출하고, `s3` 와 리턴값을 모두 터미널에 출력하세요.

터미널 출력 결과를 캡처하여 본 문서에 첨부하세요. 소스코드도 첨부파일로 제출해야 합니다.

소스코드 : [Q5] 중위표기법 수식을 후위표기법으로 변환 소스코드.txt

답변 1 (Task 1):

```
<terminated> main5 (1) [Java
```

```
-- 중위표기법 => 후위표기법 --
```

중위표기법: $A * (B + C / D)$

후위표기법: $ABCD / + *$

답변 2 (Task 2):

```
<terminated> main5 (1) [Java
```

```
-- 중위표기법 => 후위표기법 --
```

중위표기법: $(A + B) * (C - D)$

후위표기법: $AB + CD - *$

답변 3 (Task 3)

```
-- 중위표기법 => 후위표기법 --
```

중위표기법: $A + B + C * (B + C * D) / D$

후위표기법: $AB + CB CD * + * D / +$

[Q 6] 후위표기법 수식 계산 [15 점]

ArrayStack 또는 ListStack을 사용해서, 주어진 후위표기법 수식을 계산하는 public double calc (String[] s) 메소드를 작성하세요. 'calc' 메소드는 계산 결과를 double 타입으로 리턴합니다.

[Task 1]

- 후위표기 수식: String[] s1 = {"81", "37", "-211", "+", "15", "-", "/"}
- 'calc' 메소드를 호출하고, 주어진 수식과 결과를 모두 터미널에 출력하세요.

[Task 2]

- 후위표기 수식: String[] s2 = {"20.5", "10.2", "3.3", "+", "*"}
- 'calc' 메소드를 호출하고, 주어진 수식과 결과를 모두 터미널에 출력하세요.

터미널 출력 결과를 캡처하여 본 문서에 첨부하세요. 소스코드도 첨부파일로 제출해야 합니다.

소스코드 : [Q6] 후위표기법 수식 계산 소스코드.txt

답변 (Task 1):

```
<terminated> main6 (1) [Java Applet]
-- 후위표기법 수식 계산 --

81  37  -211  +  15  -  /
결과 = -0.42857142857142855
```

답변 (Task 2):

```
<terminated> main6 (1) [Java Applet]
-- 후위표기법 수식 계산 --

20.5  10.2  3.3  +  *
결과 = 276.75
```

끝! 수고하셨습니다 ☺