

[511643] 자료구조 (2019-2 학기)**실습 #06 보고서**

이름	지현한
학번	20165164
소속 학과/ 대학	소프트웨어융합대학 빅데이터전공
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: **2019. 10. 16. (수요일) 23:59 // 2 주**
 - 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: 실수형 자료형에서 문자열 자료형으로 바꾸는 방법을 몰라서 인터넷을 찾아봤습니다.

Double.parseDouble()

[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.

```
cherry
apple  orange  cherry  pear
cherry
apple  orange  cherry  grape
```

답변: Q1main

[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.

```
cherry
pear   cherry  orange  apple
cherry
grape  cherry  orange  apple
```

답변: Q2main

[Q 3] 괄호 짝 맞추기 [15 점]

ArrayStack 또는 ListStack 을 사용해서, 주어진 수식에서 괄호의 짝이 맞는지 여부를 확인하는 public boolean checkParentheses() 또는 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, str2 와 리턴값을 모두 터미널에 출력하세요.

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

답변 1 : Q3main

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

답변 2 : Q3main

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

[Q 4] 회문(Palindrome) 검사하기 [15 점]

ArrayStack 또는 ListStack 을 사용해서, 주어진 단어가 회문인지 여부를 확인하는 public Boolean checkPalindrome() 또는 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 4] String s4 = "koxko" 일 때, checkPalindrome 을 호출하고, s5 와 리턴값을 모두 터미널에 출력하세요.

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

답변 1 : Q4main `redivider: true`

답변 2 : Q4main `redder: true`

답변 3 : Q4main `a: true`

답변 4 : Q4main `abbd: false`

답변 5 : Q4main `koxko: false`

[Q 5] 중위표기법 수식을 후위표기법으로 변환 [15 점]

ArrayStack 또는 ListStack을 사용해서, 주어진 중위표기법 수식 표현을 후위표기법 수식 표현으로 변경하는 `public String getPostEq()` 또는 `public String getPostEq(String s)` 메소드를 작성하세요. 후위표기법으로 변환된 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.

답변 1 : Q5main `A*(B+C/D): ABCD/+*`

답변 2 : Q5main `(A+B)*(C-D): AB+CD-*`

답변 3 : Q5main `A+B+C*(B+C*D)/D: ABCBCD*+*D/++`

[Q 6] 후위표기법 수식 계산 [15 점]

ArrayStack 또는 ListStack을 사용해서, 주어진 후위표기법 수식을 계산하는 calc 메소드를 작성하세요. Public double calc 메소드는 계산 결과를 double 타입으로 리턴합니다. 'calc' 메소드는 인자를 받지 않을 수도 있고, ArrayStack 또는 ListStack 타입의 인자를 받을 수도 있습니다.

[Task 1]

- 후위표기 수식 : 81, 37, -211, +, 15, -, /
- 'calc' 메소드를 호출하고, 주어진 수식과 결과를 모두 터미널에 출력하세요.

[Task 2]

- 후위표기 수식 : 20.5, 10.2, +, *
- 'calc' 메소드를 호출하고, 주어진 수식과 결과를 모두 터미널에 출력하세요.

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

```
81 37 -211 + 15 - / : 2.3333333333333335
20.5 10.2 + 10 * : 307.0
```

답변: Q6main

끝! 수고하셨습니다 ㄹ