

# 자료구조

06

과목명 : 자료구조

분반 : 06

담당교수 : 박정희

학번 : 202002546

이름 : 임우진

## 목차

1. 요구사항 및 풀이과정
2. 후위 표현식 변환 코드 캡처 및 설명
3. 실행 화면 캡처

## 1. 요구사항 및 풀이과정

배열로 구현된 스택을 이용하여 중위 표현식(Infix expression)을 후위 표현식(postfix expression)으로 변환하는 메소드를 작성하고 테스트하는 것.

ArrayStack클래스를 생성하고 배열로 구현한 스택을 활용하는데 필요한 메서드들을 구현해야 한다.

input.txt의 중위 표현식들을 한 줄 씩 불러와서 피연산자가 아닌 연산자(Operation)을 스택에 조건에 따라 push, pop해야한다.

위 조건에 맞게 push, pop하기 위해

연산자 우선순위를 반환하는 getPriority함수와

ArrayStack을 받아와 getPriority()와 ArrayStack의 메서드들을 이용해서 중위 표현식을 후위 연산식으로 변환시켜주는 함수인 Change()를 구현해야 한다.

## 2. 후위 표현식 변환 코드 캡처 및 설명

먼저 연산자간 우선순위를 반환해주는 `getPriority()`이다.

```
public static int getPriority(char symbol){
    if (symbol == '+' || symbol == '-') {
        return 1;
    } else if (symbol == '*' || symbol == '/') {
        return 2;
    } else {
        return -1;
    }
}
```

+와-의 우선순위 < \*와/의 우선순위

그리하여 각각 1과 2의 우선순위를 부여한다.

(높은 수일수록 우선순위 높음)

그리고 배열을 변환시켜줄 `Change()`이다.

이 함수는 String을 받아 String으로 반환하는 함수다.

```
public static String Change(String unchanged_fn) {
    int n = unchanged_fn.length();
    char[] changed_fn = new char[n];
    ArrayStack stack = new ArrayStack(n);
    int top = -1;

    for (int i = 0; i < n; i++) {
        char c = unchanged_fn.charAt(i);
        if (Character.isLetterOrDigit(c)) { //알파벳, 숫자가 들어오면 true, 그 외엔 false
            changed_fn[++top] = c; //c삽입
        } else if (c == '(') { // 왼쪽 괄호인 경우
            stack.push(c); //스택에 삽입
        } else if (c == ')') { // 오른쪽 괄호인 경우
            while (!stack.isEmpty() && stack.peek() != '(') { //스택이 비거나 왼쪽 괄호를 만날때까지
                changed_fn[++top] = stack.pop(); //스택의 요소들을 pop하여 changed_fn에 삽입
            }
            if (!stack.isEmpty() && stack.peek() == '(') { //스택이 비거나 왼쪽 괄호가 있다면
                stack.pop(); // '(' 제거
            }
        } else { // 연산자인 경우
            while (!stack.isEmpty() && getPriority(c) <= getPriority(stack.peek())) { //비어있거나 c의 우선순위가 top의
                changed_fn[++top] = stack.pop(); //스택의 요소들을 pop하여 changed_fn에 삽입
            }
            stack.push(c); //스택에 삽입
        }
    }
}
```

```
while (!stack.isEmpty()) { // 스택에 남아있는 연산자 출력
    changed_fn[++top] = stack.pop();
}
return new String(changed_fn, offset: 0, count: top+1); //새로운 문자열을 반환
}
```

### 3. 실행화면 캡처

```

C:\Users\8146q\jdk\openjdk-19.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2022.3.2\lib\idea_rt.jar=53398:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2022.3.2\bin" -Didea.config.path=C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2022.3.2\conf -Didea.copyright.profiles=C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2022.3.2\conf\copyright -Didea.log=C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2022.3.2\log -Didea.platform.prefix=IntelliJ -jar C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2022.3.2\bin\idea_2022.3.2.jar
A B + C D - /
A B C * + D -
5 2 + 8 3 - * 4 /

```