

# 2023.02.20

≡ SSAFY

## APS기본- 스택(Stack) & 큐(Queue)

### 스택(Stack)

#### 스택의 특성

- 물건을 쌓아 올리듯 자료를 쌓아 올린 형태의 자료구조이다.
- 스택에 저장된 자료는 선형 구조를 갖는다. ★
  - 선형구조 : 자료 간의 관계가 1대 1의 관계를 갖는다.
  - 비선형구조 : 자료 간의 관계가 1대 N의 관계를 갖는다. (예: 트리)
- 스택에 자료를 삽입하거나 스택에서 자료를 꺼낼 수 있다. (한쪽에서만 가능)
- 마지막에 삽입한 자료를 가장 먼저 꺼낸다. 후입선출(LIFO, Last In First Out)이라고 부른다. ★
  - 예를 들어 스택에 1, 2, 3 순으로 자료를 삽입한 후 꺼내면 역순으로 즉 3, 2, 1 순으로 꺼낼 수 있다.

#### 스택을 프로그램에서 구현하기 위해서 필요한 자료구조와 연산

- 자료구조 : 자료를 선형으로 저장할 저장소
  - C언어에서는 배열을 사용할 수 있다.
  - 저장소 자체를 스택이라 부르기도 한다.
  - 스택에서 마지막 삽입된 원소의 위치를 top이라 부른다.
- 연산
  - 삽입 : 저장소에 자료를 저장한다. 보통 **push** 라고 부름

- 삭제 : 저장소에서 자료를 꺼낸다. 꺼낸 자료는 삽입한 자료의 역순으로 꺼낸다. 보통 **pop**
- 스택이 공백인지 아닌지를 확인하는 연산 . **isEmpty**
- 스택의 top에 있는 item(원소)를 반환하는 연산 **peek** (확인)

top

- -1 : 스택이 비어있음. `int top = -1;` 초기화
- 0 ~ n - 1 : 스택이 차있음
- n - 1 : 스택이 꽉 참

```
push(X)
if(top == n - 1)
    System.out.println("stack overflow");
else{
    top++;
    arr[top] = X;
}

pop()
if(top == -1)
    System.out.println("stack is empty");
else
    return arr[top--];
```

## 스택 구현 고려 사항

- 1차원 배열을 사용하여 구현할 경우 구현이 용이하다는 장점이 있지만 스택의 크기를 변경하기가 어렵다는 단점이 있다.
- 이를 해결하기 위한 방법으로 저장소를 동적으로 할당하여 스택을 구현하는 방법이 있다. 동적 연결리스트를 이용하여 구현하는 방법을 의미한다. 구현이 복잡하다는 단점이 있지만 메모리를 효율적으로 사용한다는 장점을 가진다. 스택의 동적 구현은 생략한다.

## 스택 응용

## 괄호 검사

- 괄호의 종류 : 대괄호 `[ ]` , 중괄호 `{ }` , 소괄호 `( )`
- 조건
  - 왼쪽 괄호의 개수와 오른쪽 괄호의 개수가 같아야 한다.
  - 같은 괄호에서 왼쪽 괄호는 오른쪽 괄호보다 먼저 나와야 한다.
  - 괄호 사이에는 포함관계만 존재한다.

## 괄호를 조사하는 알고리즘 개요

- 문자열에 있는 괄호를 차례대로 조사하면서 왼쪽 괄호를 만나면 스택에 삽입하고, 오른쪽 괄호를 만나면 스택에서 top 괄호를 삭제한 후 오른쪽 괄호와 짝이 맞는지를 검사한다.
- 이 때, 스택이 비어있으면 조건1 또는 조건2에 위배되고 괄호의 짝이 맞지 않으면 조건3에 위배된다.
- 마지막 괄호까지를 조사한 후에도 스택에 괄호가 남아 있으면 조건 1에 위배된다.

## 스택의 응용 2 : function call

### Function call

- 프로그램에서의 함수 호출과 복귀에 따른 수행 순서를 관리
  - 가장 마지막에 호출된 함수가 가장 먼저 실행을 완료하고 복귀하는 후입선출 구조이므로, 후입선출 구조의 스택을 이용하여 수행순서 관리
  - 함수 호출이 발생하면 호출한 함수 수행에 필요한 지역변수, 매개변수 및 수행 후 복귀할 주소 등의 정보를 스택 프레임(stack frame)에 저장하여 시스템 스택에 삽입
  - 함수의 실행이 끝나면 시스템 스택의 top원소(스택 프레임)을 삭제(pop)하면서 프레임에 저장되어있던 복귀주소를 확인하고 복귀
  - 함수 호출과 복귀에 따라 이 과정을 반복하여 전체 프로그램 수행이 종료되면 시스템 스택은 공백 스택이 된다.

```
package test02;

public class Test2_FunctionCall {
    public static void main(String[] args) {
        System.out.println("메인 작업 수행");
        func1();
        System.out.println("메인 작업 끝");
    }
    public static void func1() {
        System.out.println("함수 1 작업 수행");
        func2();
        System.out.println("함수 1 작업 끝");
    }
    public static void func2() {
        System.out.println("함수 2 작업 수행");
        System.out.println("함수 2 작업 끝");
    }
}
```

```
// Console
메인 작업 수행
함수 1 작업 수행
함수 2 작업 수행
함수 2 작업 끝
함수 1 작업 끝
메인 작업 끝
```