

스택 (Stack)

(Stack 구현)

정의

- 스택은 데이터를 일시적으로 쌓아 놓는 자료 구조로 데이터의 입력과 출력 순서는 후입 선출 (LIFO - Last In First Out)을 갖는다.
- 자바에서는 메서드를 호출하고 실행 할 때 프로그램 내부에서 스택을 사용 하기 때문에 재귀 방식을 통한 접근도 스택 프레임을 사용하게 된다.
- 스택에 쌓인 메소드는 한 번에 처리되는 경우는 없으며, 후입 선출의 구조에 따른다.

```

// 제네릭 스택
public class Stack<E> {
    private E[] stk;           // 스택용 배열
    private int capacity;      // 스택 용량
    private int ptr;           // 스택포인터

    //--- 실행 시 예외 : 스택이 비어 있음 ---//
    public static class EmptyGstackException extends RuntimeException {
        public EmptyGstackException() { }
    }

    //--- 실행 시 예외 : 스택이 가득 참 ---//
    public static class OverflowGstackException extends RuntimeException {
        public OverflowGstackException() { }
    }

    //--- 생성자 ---//
    public Stack(int maxlen) {
        ptr = 0;
        capacity = maxlen;
        try {
            stk = (E[])new Object[capacity]; // 스택 본체용 배열을 생성
        } catch (OutOfMemoryError e) { // 생성할 수 없음
            capacity = 0;
        }
    }

    //--- 스택에 x를 푸시 ---//
    public E push(E x) throws OverflowGstackException {
        if (ptr ≥ capacity) // 스택이 가득 참
            throw new OverflowGstackException();
        return stk[ptr++] = x;
    }

    //--- 스택에서 데이터를 팝(꼭대기의 데이터를 꺼냄) ---//
    public E pop() throws EmptyGstackException {
        if (ptr ≤ 0) // 스택이 비어 있음
            throw new EmptyGstackException();
        return stk[--ptr];
    }

    //--- 스택에서 데이터를 피크(꼭대기의 데이터를 들여다 봄) ---//
    public E peek() throws EmptyGstackException {
        if (ptr ≤ 0) // 스택이 비어 있음
            throw new EmptyGstackException();
        return stk[ptr - 1];
    }

    //--- 스택을 비웁니다 ---//
    public void clear() {
        ptr = 0;
    }
}

```