

7주차 1차시 추상자료형 스택

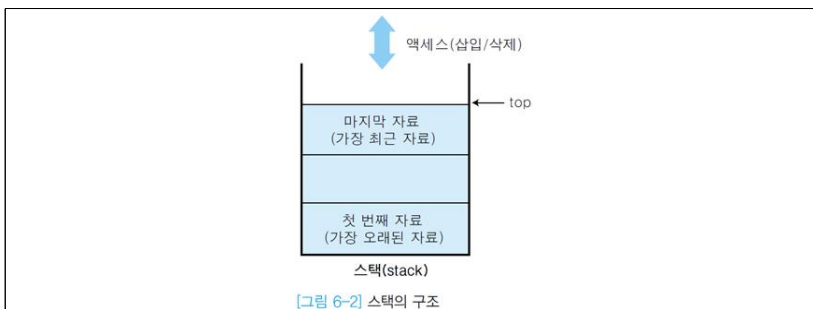
【학습목표】

1. 자료구조 스택에 대해서 설명할 수 있다.
2. 스택의 특징과 연산 방법에 대해 설명할 수 있다.

학습내용1 : 스택이란?

1. 스택(Strack)

- 접시를 쌓듯이 자료를 차곡차곡 쌓아올린 형태의 자료구조
 - 예) 뷔페식당에 쌓이 접시, 책상위에 쌓아둔 책 등
- 같은 구조와 같은 크기의 자료를 정해진 방향만 쌓을 수 있고, top이라고 정해진 곳으로만 접근이 가능
 - top을 통해서 자료가 삽입되므로 현재 스택의 가장 위에 있는 마지막 자료를 가리키며 새 자료는 top이 가리키는 자료의 위에 쌓이게 됨
- 따라서 먼저 삽입한 자료는 아래에 쌓이고 나중에 삽입한 원소는 위에 쌓이는 구조
- 마지막에 삽입된 자료가 가장 먼저 삭제되는 구조
- 후입선출(LIFO : Last In First Out)



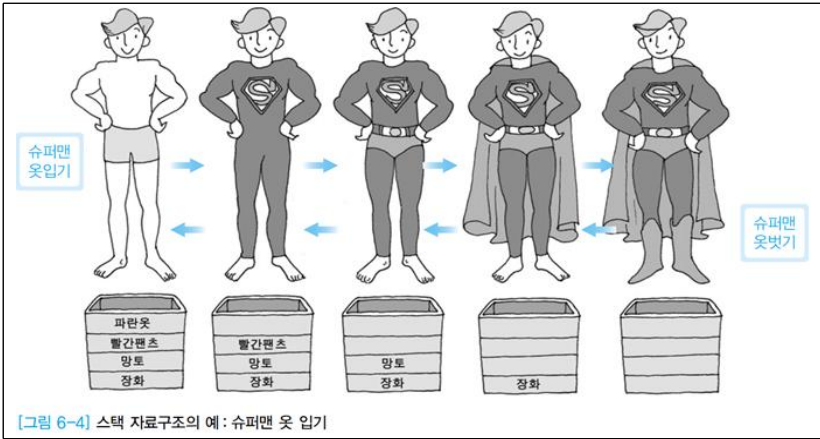
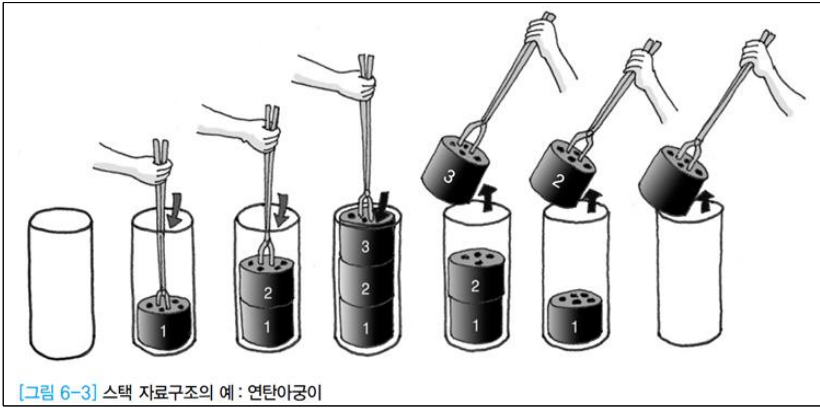
2. 스택 예제

* 연탄아궁이

- 연탄을 하나씩 순서대로 넣어 마지막에 넣은 3번 연탄이 가장 위에 쌓여있음
- 연탄을 아궁이에서 꺼낼 때에는 위에서부터 하나씩 꺼내야 하므로 가장 마지막에 넣은 3번 연탄부터 꺼내야함

* 슈퍼맨 옷 갈아입기

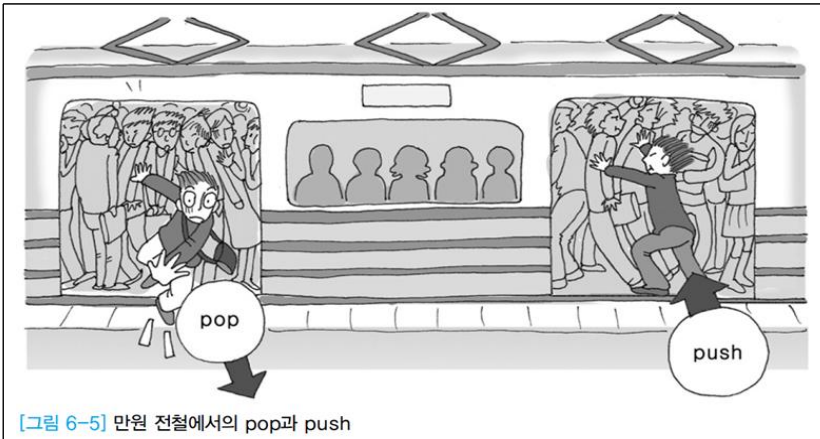
- 슈퍼맨이 옷을 벗는 순서 : ①장화 → ②망토 → ③빨간 팬츠 → ④파란 옷
- 슈퍼맨이 옷을 입는 순서 : ①파란 옷 → ②빨간 팬츠 → ③망토 → ④장화



학습내용2 : 스택의 연산

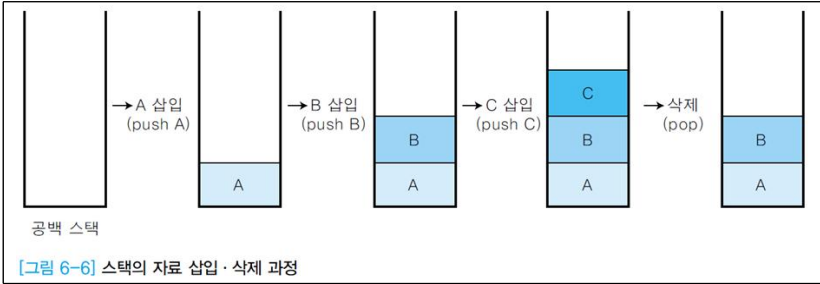
1. 스택 연산

- push : 스택에서 top을 통한 삽입 연산
- pop : 스택에서 top을 통한 삭제 연산



2. 스택에서의 원소 삽입·삭제 연산

- 공백 스택에 원소 A, B, C 순서대로 삽입하고 다시 자료 하나를 삭제하는 연산과정 동안의 스택 변화



학습내용3 : 스택 알고리즘

1. 스택에 대한 추상자료형

ADT Stack

데이터 : 0개 이상의 원소를 가진 유한 순서 리스트

연산 :

S ∈ Stack; item ∈ Element;

createStack(S) ::= create an empty stack S;
// 공백 스택 S를 생성하는 연산

push(S, item) ::= insert item onto the top of Stack S;
// 스택 S의 top에 item(원소)을 삽입하는 연산

isEmpty(S) ::= if (S is empty) then return true
 else return false;
// 스택 S가 공백인지 아닌지를 확인하는 연산

pop(S) ::= if (isEmpty(S)) then return error
 else { delete and return the top item of Stack S;
 // 스택 S의 top에 있는 item(원소)을 스택 S에서 삭제하고 반환하는 연산

delete(S) ::= if (isEmpty(S)) then return error
 else delete the top item of Stack S;
 // 스택 S의 top에 있는 item(원소)을 삭제하는 연산

peek(S) ::= if (isEmpty(S)) then return error
 else return (the top item of the Stack S);
 // 스택 S의 top에 있는 item(원소)을 반환하는 연산

End Stack

2. push 알고리즘

알고리즘 6-1 스택의 push 알고리즘

push(S, x)

top ← top+1; // ❶

if (top > stack_SIZE) then

overflow;

else

S(top) ← x; // ❷

end push()

- ❶ top ← top+1;
- 스택 S에서 top이 마지막 자료를 가리키고 있으므로 그 위에 자료를 삽입하려면 먼저 top의 위치를 하나 증가 시킴
 - 이때 top의 위치가 스택의 크기(stack_SIZE)보다 크면 오버플로우 상태가 되므로 삽입 연산을 수행하지 못하고 연산이 종료

❷ $S(top) \leftarrow x$;

- 오버플로우 상태가 아니라면 스택의 top이 가리키는 위치 $S(top)$ 에 x 를 삽입

3. pop 알고리즘

알고리즘 6-1 스택의 push 알고리즘

```
push(S, x)
  top ← top+1;           // ❶
  if (top > stack_SIZE) then
    overflow;
  else
    S(top) ← x;          // ❷
  end push()
```

알고리즘 6-2 스택의 pop 알고리즘

```
pop(S)
  if (top = 0) then underflow;
  else {
    return S(top);       // ❶
    top ← top-1;         // ❷
  }
end pop()
```

❶ return $S(top)$;

- 스택이 공백이 아니라면 top이 가리키는 원소를 먼저 반환

❷ $top \leftarrow top - 1$;

- 스택의 마지막 자료가 삭제 되었으면 top의 위치는 그 아래 자료가 되므로 top의 위치를 하나 아래로 감소

【학습정리】

1. 스택은 구조와 크기가 같은 자료를 정해진 방향으로만 쌓을 수 있고, top이라고 정해진 곳을 통해서만 접근할 수 있다.
2. 스택에서 원소를 삽입하는 연산을 push라고 하고, 삭제하는 연산을 pop이라고 한다.