

## stack을 만드는 2가지 방법

- stack 클래스를 처음 부터 새롭게 만든다.
- list의 인터페이스(함수 이름)을 변경해서 stack 처럼 보이게 한다.

## Adapter 디자인 패턴

- 클래스의 인터페이스를 변경해서 새로운 클래스 처럼 보이게 만드는 디자인 패턴

## STL Container Adapter

- sequence container 의 인터페이스(함수 이름)를 변경해서 다른 컨테이너 처럼 보이도록 만든 것.
- 3가지의 Container adapter가 제공된다.

<stack>	stack
<queue>	queue, priority_queue

- adapter와 adapter에 사용 가능한 컨테이너

stack	컨테이너의 뒤쪽에 요소 삽입/삭제가 가능해야 한다. deque, vector, list
queue	양쪽으로 접근이 가능해야 한다. deque, list      * vector 사용할 수 없다.
priority_queue	Random access iterator 만 가능. vector, deque      * list 사용할 수 없다.

## priority\_queue

- 요소를 꺼낼 때 삽입한 순서에 상관없이 **우선순위가 가장 높은 요소가** 먼저 나온다.
- 우선순위를 변경하려면 템플릿 인자로 **함수 객체를 전달**한다.