

컴퓨터공학 All in One

C/C++ 문법, 자료구조 및 심화 프로젝트 (나동빈)
제 54강 - C++ STL 컨테이너 어댑터

학습 목표

C++ STL 컨테이너 어댑터

- 1) C++ STL 컨테이너 어댑터의 필요성과 사용 방법에 대해서 이해할 수 있습니다.

C++ STL 컨테이너 어댑터

C++ STL 컨테이너 어댑터

- 1) STL 컨테이너 어댑터 라이브러리는 매우 활용도가 높은 자료구조를 제공합니다.
 - 2) 기존의 C언어를 이용하면 구현하기 까다로웠던 다양한 자료구조를 손쉽게 이용할 수 있습니다.
- Stack(스택)
 - Queue(큐)
 - Priority Queue(우선순위 큐)

C++ STL 컨테이너 어댑터

C++ STL 컨테이너 어댑터: 스택

C++ Stack STL은 다음과 같은 함수로 구성되어 있습니다.

- 추가: `push(원소)`
- 삭제: `pop()`
- 조회: `top()`
- 검사: `empty()` / `size()`

C++ STL 컨테이너 어댑터

C++ STL 컨테이너 어댑터: 스택

```
#include <iostream>
#include <stack>

using namespace std;

int main(void) {
    stack<int> s;
    s.push(7); s.push(5); s.push(4); s.pop(); s.push(6); s.pop();
    while (!s.empty()) {
        cout << s.top() << ' ';
        s.pop();
    }
    system("pause");
}
```

C++ STL 컨테이너 어댑터

C++ STL 컨테이너 어댑터: 큐

C++ Queue STL은 다음과 같은 함수로 구성되어 있습니다.

- 추가: `push(원소)`
- 삭제: `pop()`
- 조회: `front()` / `back()`
- 검사: `empty()` / `size()`

C++ STL 컨테이너 어댑터

C++ STL 컨테이너 어댑터: 큐

```
#include <iostream>
#include <queue>

using namespace std;

int main(void) {
    queue<int> q;
    q.push(7); q.push(5); q.push(4); q.pop(); q.push(6); q.pop();
    while (!q.empty()) {
        cout << q.front() << ' ' ;
        q.pop();
    }
    system("pause");
}
```

C++ STL 컨테이너 어댑터

C++ STL 컨테이너 어댑터: 우선순위 큐

```
#include <iostream>
#include <queue>

using namespace std;

int main(void) {
    int n, x;
    cin >> n;
    priority_queue<int> pq;
    for (int i = 0; i < n; i++) { cin >> x; pq.push(x); }
    while (!pq.empty()) {
        cout << pq.top() << ' ' << '\n';
        pq.pop();
    }
    system("pause");
}
```


배운 내용 정리하기

C++의 STL 컨테이너 어댑터

- 1) C++에서는 STL 컨테이너 어댑터에 속하는 Stack, Queue, Priority Queue를 손쉽게 이용할 수 있습니다.