

ICE2004 자료구조 실습 2

제 목

보고서 작성 서약서

1. 나는 타학생의 보고서를 베끼거나 여러 보고서의 내용을 짜집기하지 않겠습니다.

2. 나는 보고서의 주요 내용을 인터넷사이트 등을 통해 얻지 않겠습니다.

3. 나는 보고서의 내용을 조작하지 않겠습니다.

4. 나는 보고서 작성에 참고한 문헌의 출처를 밝히겠습니다.

5. 나는 나의 보고서를 제출 전에 타학생에게 보여주지 않겠습니다.

나는 보고서 작성시 윤리에 어긋난 행동을 하지 않고 정보통신공학인으로서 나의 명예를 지킬 것을 맹세합니다.

2021년 10월 23일

학부 정보통신공학과

학년 2학년

성명 김다영

학번 12201856



1. **개요**

다음의 기능을 수행하도록 강의자료에 있는 코드를 수정

* Stack과 같이 push, pop 등의 기능을 하는 ArrayStack class 제공
* ArrayStack class에 대한 객체 mystack(int 값을 저장하는 stack)을 정의
* mystack에 대해 push, pop, size, top 함수가 잘 작동함을 보이기
* mystack이 FullStack일 때 push하면 error message 출력
* mystack이 Emptystack일 때 pop하면 error message 출력
* mystack이 Emptystack일 때 top을 구하려 하면 error message 출력

1. **코드 캡처 및 설명**

**❑ Arraystack.h**

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

강의자료에 제시된 것과 동일하게 ArrayStack, RuntimeException,

StackEmptyException class를 작성한다. 강의자료에는 StackFullException에 대

한 정의가 제시되지 않았으므로 이를 추가한다.

**❑ Arraystack.cpp**

텍스트이(가) 표시된 사진

자동 생성된 설명

int형 변수를 저장할 수 있으며 최대 용량이 3인 ArrayStack의 객체 mystack을 선언한다.

mystack의 최대 용량이 3이므로 3개 초과의 값을 push 하면 error가 발생한다. 또 mystack

이 Empty인 경우 pop을 하거나 top의 위치를 구하려고 하면 error가 발생한다.

① case 1 : FullStack에 push를 할 때 err msg 보이기

Mystack이 FullStack일 때 err msg를 출력하기 위해 try-catch문을 이용한다.

비어있는 상태인 mystack에 push를 3번 진행한다. size() 함수를 이용해 mystack의 크기를 출력하면 mystack을 선언한 후 3번 push를 진행했으므로 3이 출력된다는 것을 예상할 수 있다. mystack의 capacity는 3이므로 현재 FullStack이며, 이 상태에서 한 번 더 push를 하게 되면 error가 발생하게 된다. try문 내에서 error가 발생했으므로 catch문의 코드가 발생하게 된다.

현재 push 함수의 정의에서 size와 capacity가 동일하면 errorMsg를 “Push to full stack”로 초기화하여 StackFullException의 생성자를 불러오므로, err를 cout 하면 Push to full stack가 출력되는 것을 예상할 수 있다. RuntimeException class에 <<연산자가 overloading 되어 있으며 StackFullException는 RuntimeException를 상속하므로 cout << err << endl;과 같은 코드가 error가 발생하지 않는다는 것을 알 수 있다.

텍스트이(가) 표시된 사진

자동 생성된 설명

② case 2 : EmptyStack에 pop을 할 때 err msg 보이기

case 1과 동일하게 예외처리를 위해 try-catch문을 이용한다.

case 1에서 push를 3번 했으므로 현재 mystack의 size는 3이다. 이후 mystack을 3번 pop

을 하면 mystack은 EmptyStack이 되며, 한 번 더 pop을 시도하면 EmptyStack 상태에서

pop을 시도한 것이 되므로 error가 발생함에 따라 catch문 코드가 실행된다.

현재 pop 함수의 정의에서 empty 함수의 반환값이 True면 errorMsg를 “pop from empty

stack”으로 초기화하여 StackEmptyException의 생성자를 불러오므로, err를 cout 하면 pop

from empty stack가 출력되는 것을 예상할 수 있다.

텍스트이(가) 표시된 사진

자동 생성된 설명

③ case 3 : EmptyStack의 top을 구할 때 err msg 보이기

case 1과 동일하게 예외처리를 위해 try-catch문을 이용한다.

case 2에서 pop을 3번 진행하여 mystack이 empty인 상태이다. 따라서 mystack의 top에 위

치한 값을 불러오려고 하면 에러가 발생하게 되고 catch문 코드가 실행된다.

현재 top 함수의 정의에서 empty 함수의 반환값이 True면 errorMsg를 “Top of empty

stack”으로 초기화하여 StackEmptyException의 생성자를 불러오므로 err를 cout 하면 Top

of empty stack가 출력되는 것을 예상할 수 있다.

1. **결과 캡처**

텍스트이(가) 표시된 사진

자동 생성된 설명

코드 실행 결과 1, 2, 3을 순서대로 push하고 top 함수를 호출하면 맨 마지막에 push한 3이 출력되는 것을 알 수 있다. 그 후 pop을 진행하면 3, 2, 1 순서대로 pop이 되며 이를 통해 stack의 특징인 LIFO(last in, first out)가 적용이 된다는 것을 알 수 있다.

비어있는 mystack에 3번 push하면 FullStack이 되며 한 번 더 push를 실행하면 error 메시지가 출력되어 “Push to full stack” 문구가 출력된다.

3번 push된 상태에서 size 함수의 반환값을 출력하면 3이 출력된다.

이후 3번 pop을 진행하여 mystack이 EmptyStack이 된 상태에서 한 번 더 pop을 실행하면 error 메시지가 출력되어 “Pop from empty stack” 문구가 출력된다.

또 EmptyStack 상태에서 top 함수를 실행하면 error 메시지가 “Top of empty stack” 문구가 출력된다.