C++ 프로그래밍

list concat 구현

130206 선윤태

#include <iostream>

using namespace std;

//Node

class Node {

public:

int data;

Node\* prevNode;

Node(int data);

~Node();

};

Node::Node(int data) {

this->data = data;

}

Node::~Node() {

cout << "deleting node (" << this->data << ")" << endl;

}

//Stack

class Stack {

public:

Node \* topNode;

int length;

Stack();

Stack(Stack& st);// copy reference생성자

void push(int data);//stack의 맨 위에 요소를 추가

int pop();//stack의 맨 위에 잇는 요소를 제거하며 return

int peek();// stack 맨 위에 있는 요소를 return

void concat(Stack st);

};

Stack::Stack() {

this->topNode = NULL;

this->length = 0;

}

Stack::Stack(Stack& st) {

this->topNode = NULL;

this->length = 0;

Stack s1;

int n = st.length;

for (int i = 0; i < n; i++) {

Node\* old = s1.topNode;

s1.topNode = st.topNode;

st.topNode = st.topNode->prevNode;

s1.topNode->prevNode = old;

s1.length++;

}

for (int i = 0; i < n; i++) {

Node\* old = this->topNode;

this->topNode = s1.topNode;

s1.topNode = s1.topNode->prevNode;

this->topNode->prevNode = old;

length++;

}

cout << "Stack(st)실행" << endl;

}

void Stack::push(int data) {

if (this->topNode == NULL) {

this->topNode = new Node(data);

}

else {

Node\* old = this->topNode;//old에 topNode를 넣고

this->topNode = new Node(data);//topNode에 새로운 node를 넣고

this->topNode->prevNode = old;//prevNode에 old(최초topNode)를 넣는다

}

length++;

cout << "push:" << data << endl;

}

int Stack::pop() {

int data = this->topNode->data;

this->topNode = this->topNode->prevNode;

length--;

return data;

}

int Stack::peek() {

return this->topNode->data;

}

void Stack::concat(Stack st) {

Stack s3;

int n = st.length;

int data;

for (int i = 0; i < n; i++) {

s3.push(st.pop());

}

for (int i = 0; i < n; i++) {

data = s3.pop();

push(data);

st.push(data);

}

}

int main()

{

Stack s1;

s1.push(1);//1넣음(1)

cout << "s1.peek:" << s1.peek() << endl;

s1.push(2);//2넣음(1 2)

cout << "s1.peek:" << s1.peek() << endl;

s1.push(4);//4넣음(1 2 4)

cout << "s1.peek:" << s1.peek() << endl;

Stack s2;

s2.push(100);//100넣음(100)

cout << "s2.peek:" << s2.peek() << endl;

s2.push(200);//200넣음(100 200)

cout << "s2.peek:" << s2.peek() << endl;

s2.push(400);//400넣음(100 200 400)

cout << "s2.peek:" << s2.peek() << endl;

s1.concat(s2);//s1(1 2 4 100 200 400) s2(100 200 400)

cout << "s1.peek:" << s1.peek() << endl;

cout << "s2.peek:" << s2.peek() << endl;

s1.pop();

cout << "s1.peek:" << s1.peek() << endl;

s1.pop();

cout << "s1.peek:" << s1.peek() << endl;

s1.pop();

cout << "s1.peek:" << s1.peek() << endl;

s1.pop();

cout << "s1.peek:" << s1.peek() << endl;

s1.pop();

cout << "s1.peek:" << s1.peek() << endl;

}