

레포트#2

자료구조(SW) 2분반

소프트웨어학과

32170578 김산

<코드>

```
/*report#2.c, linked stack, linked queue pop and push program, by 32170578 san kim, waterfog9580@gmail.com*/
#include <iostream>
using namespace std;
class LinkedStack; // LinkedStack 전방선언
class LinkedQueue; // LinkedQueue 전방선언
class ChainNode{
friend class LinkedStack;
friend class LinkedQueue;
private:
    int data;
    ChainNode *Link;
public:
    ChainNode(int element = 0 , ChainNode * next = 0 ) // 생성자
    {
        data = element;
        Link = next;
    }
};
class LinkedStack{           //연결스택
private:
    ChainNode *top;
public:
    LinkedStack(){ top = 0 ;};
    void printStack();
    void Push(const int&);    // 삽입 함수
    int *Pop(int &);         // 삭제 함수
};
void LinkedStack::printStack(){ //스택 내용 출력 함수
    ChainNode* p = top;
    cout << "스택(LIFO 순) : ";
    while (p){
        cout << p->data << " ";
        p = p->Link;
    }
    cout << endl;
}
void LinkedStack::Push(const int& e){
    top = new ChainNode(e,top);
}
int* LinkedStack::Pop(int& x){
    if (top==0 ) return 0 ;    //빈 스택이면 0을 반환
    ChainNode* delNode = top;
    x = top->data;              //top 노드의 data 필드를 x에 저장
    top = top->Link;           //top 위치를 다음 노드로 이동
    delete delNode;           // 노드 삭제
    return &x;
}
class LinkedQueue{           //연결 큐
private:
```

```

    ChainNode *front;
    ChainNode *rear;
public:
    LinkedQueue(){ front = 0 ; rear = 0 ;};
    void printQueue();
    void Push(const int&);    // 삽입 함수
    int *Pop(int &);        // 삭제 함수
};

void LinkedQueue::printQueue(){ //큐 내용 출력 함수
    ChainNode* p = front;
    cout << "큐(FIFO 순) : ";
    while (p){
        cout << p->data << " ";
        p = p->Link;
    }
    cout << endl;
}

void LinkedQueue::Push(const int& e){
    if (front==0 ) front=rear=new ChainNode(e,0 ); //공백큐
    else rear=rear->Link=new ChainNode(e,0 ); //노드를 삽입하고 rear를 수정함
}

int* LinkedQueue::Pop(int& retvalue){
    if (front==0 ) return 0 ;    //공백 큐면 0을 반환
    ChainNode* delNode = front;
    retvalue = front->data;    //삭제되는 값을 retvalue에 저장
    front = front->Link;    //앞의 노드를 제거
    delete delNode;    //노드 반환
    return &retvalue;
}

int main(){
    LinkedStack l_stack;
    LinkedQueue l_queue;
    int menu;
    int value;
    cout << "      ---- 메뉴 ----      " << endl;
    cout << "1.스택에 삽입    2.큐에 삽입 " << endl;
    cout << "3.스택에서 삭제   4.큐에서 삭제 " << endl;
    cout << "5.스택 내용 보기   6.큐 내용 보기" << endl;
    while (1 )
    {
        menu = 0;
        value = 0;
        cin >> menu; //메뉴 선택
        if (menu <= 2 ) cin >> value; //1,2 선택시 삽입할 값을 입력 받음
        switch (menu)
        {
            case 1 :
                l_stack.Push(value);
                break ;
            case 2 :
                l_queue.Push(value);
                break ;
            case 3 :
                l_stack.Pop(value);
                break ;
            case 4 :
                l_queue.Pop(value);
                break ;

```

```

case 5 :
    l_stack.printStack();
    break ;
case 6 :
    l_queue.printQueue();
    break ;
default : //메뉴 이외 숫자를 입력받으면 프로그램 종료
    cout <<"프로그램 종료"<< endl;
    return 0;
}
}
}

```

〈실행 화면〉

```

TERMINAL  PROBLEMS  OUTPUT  DEBUG CONSOLE
-----
      ---- 메뉴 ----
1.스택에 삽입      2.큐에 삽입
3.스택에서 삭제    4.큐에서 삭제
5.스택 내용 보기   6.큐 내용 보기
1 30
1 58
1 26
5
스택(LIFO 순) : 26 58 30
3
5
스택(LIFO 순) : 58 30
2 1
2 2
2 3
2 4
2 5
6
큐(FIFO 순) : 1 2 3 4 5
4
6
큐(FIFO 순) : 2 3 4 5
7
프로그램 종료
[1] + Done          "/usr/bin/gdb" --interpreter=mi --tt
san@DESKTOP-ORAM3EQ:/mnt/c/workspaces$

```