

과목: 자료구조

교수: 최재현 교수님

자료구조 <과제 5>

- 스택과 큐 클래스의 작성 -

홍지훈

이름: 홍지훈

학과: 소프트웨어학부

분반: 나

학번 : 20201777

0. 과제

- 강의내용과 교재를 참고하여 스택과 큐 클래스를 작성하고 예제코드를 작성하여 제출

1. 소스코드

1-1. Stack.h

```
#pragma once
#include <iostream>

using namespace std;

//cpp 파일을 따로 나누면 오류가 생겨서 헤더파일에 합쳤습니다.
template<class T>
class Stack
{
private:
    T* stack;
    int top;
    int capacity;

public:
    Stack(int stackCapacity = 10);

    bool IsEmpty() const;

    T& Top() const;

    void Push(const T& item);

    void Pop();

    void print();
};

template <class T>
void ChangeSize1D(T*& a, const int oldSize, const int newSize)
{
    if (newSize < 0) throw "New length must be >= 0";

    T* temp = new T[newSize];
    int number = min(oldSize, newSize);
    copy(a, a + number, temp);
    delete[] a;
    a = temp;
}

template<class T>
Stack<T>::Stack(int stackCapacity) : capacity(stackCapacity)
{
    if (capacity < 1) throw "Stack capacity must be > 0";
    stack = new T[capacity];
}
```

```

    top = -1;
}

template<class T>
inline bool Stack<T>::IsEmpty() const { return top == -1; }

template<class T>
inline T& Stack<T>::Top() const
{
    if (IsEmpty()) throw "Stack is empty";
    return stack[top];
}

template<class T>
void Stack<T>::Push(const T& x)
{
    if (top == capacity - 1)
    {
        ChangeSize1D(stack, capacity, 2 * capacity);
        capacity *= 2;
    }
    stack[++top] = x;
}

template<class T>
void Stack<T>::Pop()
{
    if (IsEmpty()) throw "Stack is empty. Cannot delete.";
    stack[top--].~T();
}

template<class T>
void Stack<T>::print()
{
    for (int i = 0; i <= top; i++) {
        cout << stack[i] << " ";
        if (i == top)
            continue;
        cout << "-> ";
    }
    cout << endl;
}

```

1-2. Queue.h

```

#pragma once
#include <iostream>

```

```

using namespace std;

//cpp 파일을 따로 나누면 오류가 생겨서 헤더파일에 합쳤습니다.
template<class T>
class Queue
{
private:
    T* queue;
    int front;
    int rear;
    int capacity;

public:
    Queue(int queueCapacity = 10);

    bool IsEmpty() const;

    T& Front() const;

    T& Rear() const;

    void Push(const T& item);

    void Pop();

    void print();
};

template <class T>
Queue<T>::Queue(int queueCapacity) : capacity(queueCapacity)
{
    if (capacity < 1) throw "Queue capacity must be > 0";
    queue = new T[capacity];
    front = rear = 0;
}

template <class T>
inline bool Queue<T>::IsEmpty() const { return front == rear; }

template <class T>
inline T& Queue<T>::Front() const
{
    if (IsEmpty()) throw "Queue is empty. No front element";
    return queue[(front + 1) % capacity];
}

template <class T>
inline T& Queue<T>::Rear() const

```

```

{
    if (IsEmpty()) throw "Queue is empty. No rear element";
    return queue[rear];
}

template <class T>
void Queue<T>::Push(const T& x)
{
    if ((rear + 1) % capacity == front)
    {
        T* newQueue = new T[2 * capacity];

        int start = (front + 1) % capacity;
        if (start < 2)
            copy(queue + start, queue + start + capacity - 1, newQueue);
        else
        {
            copy(queue + start, queue + start + capacity - 1, newQueue);
            copy(queue, queue + rear + 1, newQueue + capacity - start);
        }
        front = 2 * capacity - 1;
        rear = capacity - 2;
        capacity *= 2;
        delete[] queue;
        queue = newQueue;
    }
    rear = (rear + 1) % capacity;
    queue[rear] = x;
}

template <class T>
void Queue<T>::Pop()
{
    if (IsEmpty()) throw "Queue is empty. Cannot delete.";
    front = (front + 1) % capacity;
    queue[front].~T();
}

template <class T>
void Queue<T>::print()
{
    for (int i = front+1; i <= rear; i++)
    {
        cout << queue[i] << " ";
        if (i == rear)
            continue;
        cout << "-> ";
    }
}

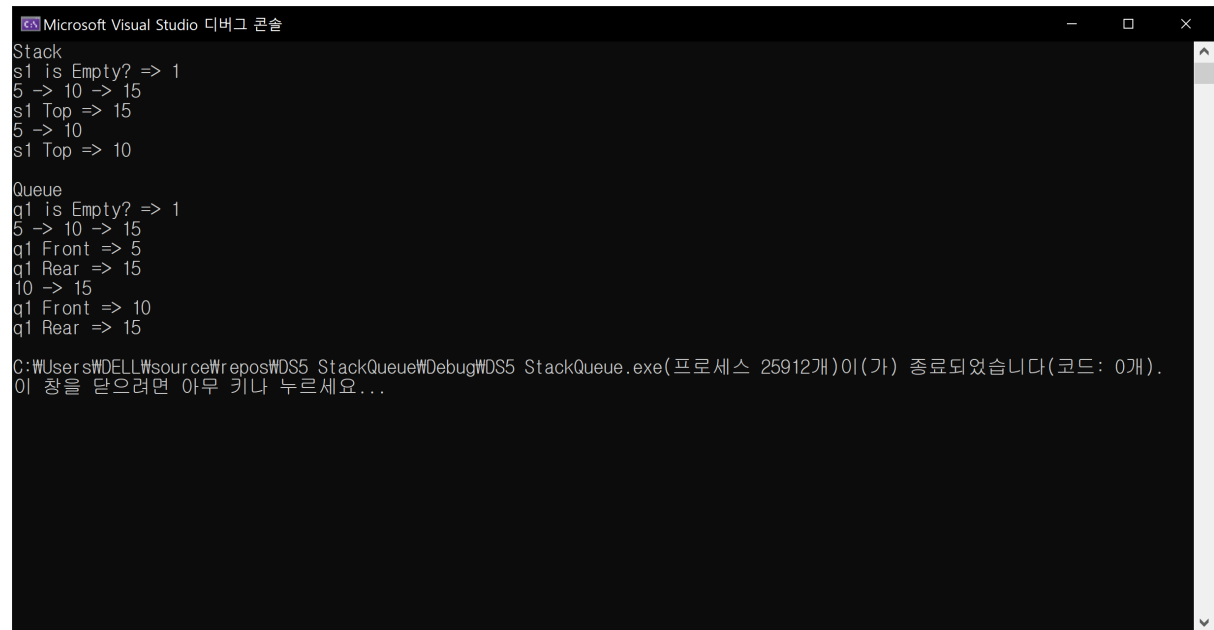
```

```
}  
cout << endl;  
}
```

1-3. Main.cpp

```
#include "Stack.h"  
#include "Queue.h"  
  
using namespace std;  
  
int main(void) {  
    //Stack  
    cout << "Stack" << endl;  
    Stack<int> s1;  
    cout << "s1 is Empty? => " << s1.IsEmpty() << endl;  
    s1.Push(5);  
    s1.Push(10);  
    s1.Push(15);  
    s1.print();  
    cout << "s1 Top => " << s1.Top() << endl;  
    s1.Pop();  
    s1.print();  
    cout << "s1 Top => " << s1.Top() << endl;  
  
    //Queue  
    cout << endl << "Queue" << endl;  
    Queue<int> q1;  
    cout << "q1 is Empty? => " << q1.IsEmpty() << endl;  
    q1.Push(5);  
    q1.Push(10);  
    q1.Push(15);  
    q1.print();  
    cout << "q1 Front => " << q1.Front() << endl;  
    cout << "q1 Rear => " << q1.Rear() << endl;  
    q1.Pop();  
    q1.print();  
    cout << "q1 Front => " << q1.Front() << endl;  
    cout << "q1 Rear => " << q1.Rear() << endl;  
    return 0;  
}
```

2. 실행 화면



```
Microsoft Visual Studio 디버그 콘솔
Stack
s1 is Empty? => 1
5 -> 10 -> 15
s1 Top => 15
5 -> 10
s1 Top => 10

Queue
q1 is Empty? => 1
5 -> 10 -> 15
q1 Front => 5
q1 Rear => 15
10 -> 15
q1 Front => 10
q1 Rear => 15

C:\Users\w\DELL\source\repos\DS5 StackQueue\Debug\DS5 StackQueue.exe(프로세스 25912개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```