레포트

자료구조<과제> -Stack과 Queue 클래스 작성-

소프트웨어학부

20201718

강대겸

1. 소스코드(main.cpp)

```cpp
#include<iostream>
#include<algorithm>
#include"Stack.h"
#include"Queue.h"
#include"Stack.cpp"
#include"Queue.cpp"


using namespace std;

int main() {
	int num0;
	Stack<int> stack0;
	cout << boolalpha << stack0.IsEmpty() << " : Empty == true" << endl;
	for (int i = 0; i < 10; i++) {
		stack0.Push(i);
	}
	cin >> num0;
	stack0.Push(num0);
	cout << "Top : " << stack0.Top() << endl;
	stack0.Pop();
	stack0.Pop();
	cout << "Top : " << stack0.Top() << endl;
	cout << boolalpha << stack0.IsEmpty() << " : Not Empty == false" << endl;

	Queue<int> queue0;
	cout << boolalpha << queue0.IsEmpty() << " : Empty == true" << endl;
	for (int i = 10; i < 20; i++) {
		queue0.Push(i);
	}

	cin >> num0;
	queue0.Push(num0);
	queue0.Pop();
	queue0.Pop();
	cout << "Front element : " << queue0.Front() << endl;
	cout << "Rear element : " << queue0.Rear() << endl;
	cout << boolalpha << queue0.IsEmpty() << " : Not Empty == false" << endl;

	return 0;
}
```

2. 소스코드(Stack.cpp)

```cpp
#include"Stack.h"
#include<algorithm>
```

```
using namespace std;
template <typename T>
Stack<T>::Stack(int stackCapacity) : capacity(stackCapacity) {
        if (capacity < 1) throw "Stack capacity must be > 0";
        stack = new T[capacity];
        top = -1;
}

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

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

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

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

3. 소스코드(Queue.cpp)

```
#include"Queue.h"
```

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

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

template<typename T>
inline T& Queue<T>::Rear() const {
        if (IsEmpty()) throw " Queue is empty. No rear element";
        return queue[rear];
}

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

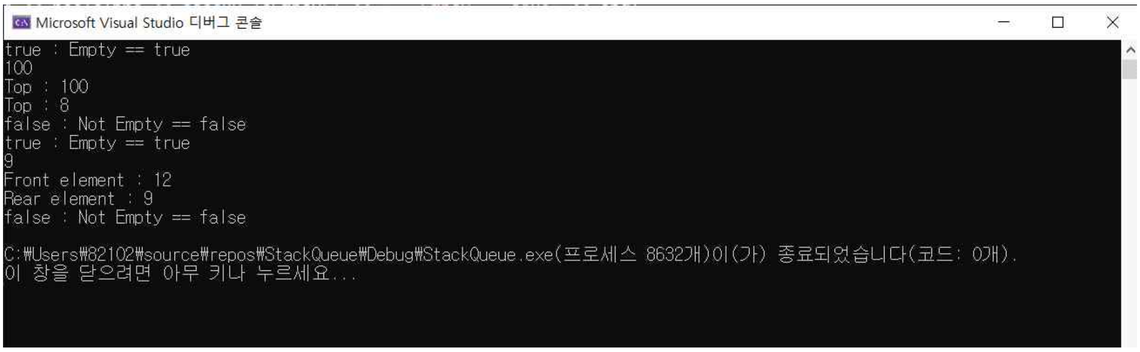
4.헤더파일(Stack.h)

```
#pragma once
template <typename T>
class Stack {
private :
        T* stack;
        int top;
        int capacity;
public :
        Stack(int stackCapacity = 10);
        bool IsEmpty() const;
        T& Top() const;
        void ChangeSize1D(T*& a, const int oldSize, const int newSize);
        void Push(const T& item);
        void Pop();
};
```

5.헤더파일(Queue.h)

```
#pragma once
template <typename T>
class Queue {
private :
        T* queue;
        int front, rear, capacity;
public :
        Queue(int queueCapacity = 10);
        bool IsEmpty() const;
        T& Front() const;
        T& Rear() const;
        void Push(const T& item);
        void Pop();
};
```

2.실행화면



```
true : Empty == true
100
Top : 100
Top : 8
false : Not Empty == false
true : Empty == true
9
Front element : 12
Rear element : 9
false : Not Empty == false

C:₩Users₩82102₩source₩repos₩StackQueue₩Debug₩StackQueue.exe(프로세스 8632개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```