# 循环队列

## 循环队列的实现

SeqQueue.h

#ifndef SEQQUEUE\_H

#define SEQQUEUE\_H

template **<**class T**,** int MaxSize**>**

class SeqQueue

**{**

private**:**

T data**[**MaxSize**];** //存放队列元素数组

int front**,**rear**;** //队首和队尾指针

public**:**

SeqQueue**();** //构造函数，置空

**~**SeqQueue**();** //析构函数

void EnQueue**(**T x**);** //元素x入队

T DeQueue**();** //队首元素出队

T GetQueue**();** //取队头元素（不删除）

bool isEmpty**();** //判断队列是否为空

bool isFull**();** //判断队列是否满

**};**

#endif // SEQQUEUE\_H

SeqQueue.cpp

#include "SeqQueue.h"

#include <iostream>

#include <stdlib.h>

**using** **namespace** std**;**

//循环队列初始化

template **<**class T**,** int MaxSize**>**

SeqQueue**<**T**,**MaxSize**>::**SeqQueue**()**

**{**

front **=** rear **=** 0**;**

**}**

//析构函数

template **<**class T**,** int MaxSize**>**

SeqQueue**<**T**,**MaxSize**>::~**SeqQueue**()**

**{**

**}**

//循环队列入队

template **<**class T**,** int MaxSize**>**

void SeqQueue**<**T**,**MaxSize**>::**EnQueue**(**T x**)**

**{**

**if(**isFull**())**

**{**

cerr **<<** "Queue is full!"**;**

exit**(**1**);**

**}**

rear **=** **(**rear**+**1**)** **%** MaxSize**;** //队尾指针循环意义下+1

data**[**rear**]** **=** x**;** //在队尾插入元素

**}**

//循环队列出队

template **<**class T**,** int MaxSize**>**

T SeqQueue**<**T**,**MaxSize**>::**DeQueue**()**

**{**

**if(**isEmpty**())**

**{**

cerr **<<** "Queue is empty!"**;**

exit**(**1**);**

**}**

front **=** **(**front**+**1**)** **%** MaxSize**;** //队首指针循环意义下+1

**return** data**[**front**];** //取出并返回队首元素

**}**

//取出队首元素

template **<**class T**,** int MaxSize**>**

T SeqQueue**<**T**,**MaxSize**>::**GetQueue**()**

**{**

**if(**isEmpty**())**

**{**

cerr **<<** "Queue is empty!"**;**

exit**(**1**);**

**}**

**return** data**[(**front**+**1**)** **%** MaxSize**];**

**}**

//判空

template **<**class T**,** int MaxSize**>**

bool SeqQueue**<**T**,**MaxSize**>::**isEmpty**()**

**{**

**return** **(**rear **==** front**);**

**}**

//判满

template **<**class T**,** int MaxSize**>**

bool SeqQueue**<**T**,**MaxSize**>::**isFull**()**

**{**

**return** **((**rear**+**1**)** **%** MaxSize **==** front**);**

**}**

main.cpp测试代码：

#include "SeqQueue.cpp"

#include <iostream>

**using** **namespace** std**;**

int main**()**

**{**

SeqQueue**<**int**,**20**>** S1**;**

int a**[**10**];**

**for(**int i **=** 0**;** i**<**10**;** i**++)**

**{**

a**[**i**]** **=** i**;**

S1**.**EnQueue**(**a**[**i**]);**

**}**

**for(**int i **=** 0**;** i**<**10**;** i**++)**

**{**

S1**.**GetQueue**();**

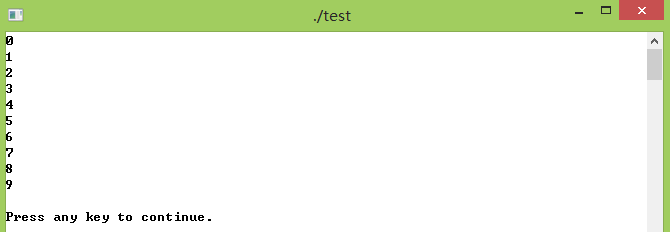
cout **<<** S1**.**DeQueue**()** **<<**endl**;**

**}**

**return** 0**;**

**}**

测试截图：



## 利用flag判满

SeqQueue.h

private**:**

bool flag**;** //判断标志

SeqQueue.cpp

//循环队列入队

template **<**class T**,** int MaxSize**>**

void SeqQueue**<**T**,**MaxSize**>::**EnQueue**(**T x**)**

**{**

**if(**isFull**())**

**{**

cerr **<<** "Queue is full!"**;**

exit**(**1**);**

**}**

flag **=** **true;**

//调整指针移动与赋值顺序

data**[**rear**]** **=** x**;** //在队尾插入元素

rear **=** **(**rear**+**1**)** **%** MaxSize**;** //队尾指针循环意义下+1

**}**

//循环队列出队

template **<**class T**,** int MaxSize**>**

T SeqQueue**<**T**,**MaxSize**>::**DeQueue**()**

**{**

**if(**isEmpty**())**

**{**

cerr **<<** "Queue is empty!"**;**

exit**(**1**);**

**}**

flag **=** **false;**

//调整指针移动与赋值顺序

T temp **=** data**[**front**];**

front **=** **(**front**+**1**)** **%** MaxSize**;** //队首指针循环意义下+1

**return** temp**;** //取出并返回队首元素

**}**

//判空

template **<**class T**,** int MaxSize**>**

bool SeqQueue**<**T**,**MaxSize**>::**isEmpty**()**

**{**

**return** **((**rear **==** front**)** **&&** **!**flag**);**

**}**

//判满

template **<**class T**,** int MaxSize**>**

bool SeqQueue**<**T**,**MaxSize**>::**isFull**()**

**{**

**return** **((**rear **==** front**)** **&&** flag**);**

**}**

## 计数器判满

//循环队列初始化

template **<**class T**,** int MaxSize**>**

SeqQueue**<**T**,**MaxSize**>::**SeqQueue**()**

**{**

front **=** rear **=** 0**;**

count **=** 0**;** //初始化count

**}**

//循环队列入队

template **<**class T**,** int MaxSize**>**

void SeqQueue**<**T**,**MaxSize**>::**EnQueue**(**T x**)**

**{**

**if(**isFull**())**

**{**

cerr **<<** "Queue is full!"**;**

exit**(**1**);**

**}**

//调整指针移动与赋值顺序

data**[**rear**]** **=** x**;** //在队尾插入元素

rear **=** **(**rear**+**1**)** **%** MaxSize**;** //队尾指针循环意义下+1

count**++;** //计数器加1

**}**

//循环队列出队

template **<**class T**,** int MaxSize**>**

T SeqQueue**<**T**,**MaxSize**>::**DeQueue**()**

**{**

**if(**isEmpty**())**

**{**

cerr **<<** "Queue is empty!"**;**

exit**(**1**);**

**}**

//调整指针移动与赋值顺序

T temp **=** data**[**front**];**

front **=** **(**front**+**1**)** **%** MaxSize**;** //队首指针循环意义下+1

count**--;** //计数器减1

**return** temp**;** //取出并返回队首元素

**}**

//判空

template **<**class T**,** int MaxSize**>**

bool SeqQueue**<**T**,**MaxSize**>::**isEmpty**()**

**{**

**return** **((**rear **==** front**)** **&&** **(**count **==** 0**));**

**}**

//判满

template **<**class T**,** int MaxSize**>**

bool SeqQueue**<**T**,**MaxSize**>::**isFull**()**

**{**

**return** **((**rear **==** front**)** **&&** **(**count **==** MaxSize**));**

**}**

main.cpp测试

#include "SeqQueue.cpp"

#include <iostream>

**using** **namespace** std**;**

int main**()**

**{**

SeqQueue**<**int**,**10**>** S1**;**

int a**[**20**];**

**for(**int i **=** 0**;** i**<**20**;** i**++)**

**{**

a**[**i**]** **=** i**;**

S1**.**EnQueue**(**a**[**i**]);**

**}**

**for(**int i **=** 0**;** i**<**10**;** i**++)**

**{**

S1**.**GetQueue**();**

cout **<<** S1**.**DeQueue**()** **<<**endl**;**

**}**

**return** 0**;**

**}**

测试结果：

