数据结构实验报告

评分

满分——5分

学号：2015111833 姓名：邓金红 专业：计算机科学与技术

知识范畴：栈与队列 完成日期：2017年03月31日

实验题目：基于循环队列的排队买票模拟程序

实验内容及要求：

编程建立循环队列存储结构，对排队买票过程进行模拟。要求程序在控制台屏幕上显示字符菜单：

1. 排队——输入新到达的买票人姓名，加入买票队列中；

2. 售票——排队队列中最前面的人购票成功，显示信息并将其从队列中删除；

　　3. 查看队列——从队首到队尾依次列出所有正在排队买票人的姓名；

4. 结束——退出系统。

“排队”时，若队满，应提示等待（排队不成功）；

“售票”时，若队空，应提示无人排队（售票失败）。

实验目的：掌握循环队列的基本操作。

数据结构设计简要描述：

//单个人的信息

typedef struct person

{ string name; //排队人的名字

}Person;

typedef struct node

{ Person \*elem;

int n; //队列容量

int f; //队头指针

int r; //队尾指针

}SqQueue;

用结构体Person存储买票人的姓名，SqQueue作为循环队列的存储结构。

算法设计简要描述：

通过用循环对列的方式进行买票人的出队入队操作记录。排队加入队列，售票出队列。

判断队空和队满等多种情况。

输入/输出设计简要描述：

输入为数字，按照题目的要求，不同的数字进行不同的操作，1. 排队，2. 售票，3. 查看队列4. 结束，除去界面悬停是的额外输入，其余任意输入均为错误的输入。

不同操作有着不同的输出结果，但类型都为字符串，具体输入输出见测试。

编程语言说明：

C语言以及部分C++特性,如:引用形参

主要函数说明：

void initQueue(SqQueue &q, int n) //初始化一个循环队列

int Empty(SqQueue &q) //判断一个队列是否为空

int full(SqQueue &q) //判断一个队列是否已满

int enQueue(SqQueue &q, Person e) //入队操作

int delQueue(SqQueue &q, Person &e) //出队操作

void show(SqQueue &q) //展示队列所有元素

void mainMenu() //主菜单的显示

程序测试简要报告：

初始化队列长度为3,以下为程序具体过程：

屏幕打印输出：

1.排队

2.售票

3.查看队列

4.结束

输入：1 //排队

输出：Please in put your name:

输入: Tom

输出：Hello Tom

You are successful! press any key to continue......

输入: 任意键 1 //排队

输出：Please in put your name:

输入: Liam

输出：Hello Liam

You are successful! press any key to continue......

输入: 任意键 1 //排队

输出：Please in put your name:

输入: Eric

输出：Hello Eric

You are successful! press any key to continue......

输入: 任意键 3 //查看队列

输出: Name:Tom

Name:Liam

Name:Eric

输入: 任意键 1

输出：Please in put your name:

输入: 任意姓名 //队满

输出：queue full！press any key to continue......

输入：任意键 2 //售票,出队

输出：name: Tom

Ticket success! press any key to continue......

输入：任意键 3 //售票,出队

输出: Name:Liam

Name:Eric

输入：任意键 4 //退出

程序结束

源程序代码：

#include <iostream>

#include <conio.h>

#include <stdlib.h>

#include <stdio.h>

using namespace std;

//单个人的信息

typedef struct person

{

char name[10]; //排队人的名字

}Person;

typedef struct node

{

Person \*elem;

int n; //队列容量

int f; //队头指针

int r; //队尾指针

}SqQueue;

void initQueue(SqQueue &q, int n)

{

q.r = q.f = -1;

q.n = n + 1;

}

int Empty(SqQueue &q)

{

return q.f == q.r;

}

int full(SqQueue &q)

{

return (q.r + 1)%q.n == (q.f + q.n)%q.n;

}

int enQueue(SqQueue &q, Person e)

{

if((q.r +1)%q.n == (q.f + q.n) % q.n )

return 0; //队满，入队失败

q.r = (q.r + 1) % q.n;

q.elem[q.r] = e;

return 1; //入队成功

}

int delQueue(SqQueue &q, Person &e)

{

if(q.r == q.f)

return 0; //队列为空，出队失败

q.f = (q.f + 1) % q.n;

e = q.elem[q.f];

return 1; //出队成功返回1

}

void show(SqQueue &q)

{

if(Empty(q))

{

cout << "Queue is empty!" << endl;

return;

}

int i;

for(i=q.f + 1; (q.r + q.n)%q.n != (i + q.n)%q.n; i = (i + 1)%q.n)

{

cout << "Name:" << q.elem[i].name << endl;

}

cout << "Name:" << q.elem[i].name << endl;

}

void mainMenu()

{

cout << "1.排队" << endl;

cout << "2.售票" << endl;

cout << "3.查看队列" << endl;

cout << "4.结束" << endl;

}

int main()

{

SqQueue q ;

initQueue(q, 3);

while(1)

{

char ch;

mainMenu();

ch = getch();

switch (ch)

{

case '1' :

{

// string name;

Person p;

cout << "Please input your name:" ;

cin >> p.name;

if(enQueue(q, p))

{

cout << "Hello " << p.name << endl;

cout << "You are successful! press any key to continue......";

}

else

{

cout << "queue full! press any key to continue......";

}

break;

}

case '2':

{

Person p;

if(delQueue(q, p)) //删除成功

{

cout << "name: " << p.name << endl;

cout << "ticket success! press any key to continue......" ;

}

else //队列为空

{

cout << "failed! There is no person is waiting! press any key to continue......";

}

break;

}

case '3':

{

show(q);

cout << " press any key to continue......";

break;

}

case '4':

{

exit(0);

break;

}

default:

cout << "input error, press any key to continue......" << endl;

break;

}

ch = getch();

system("cls");

}

return 0;

}