Rajvaibhav Rahane

17u283 223045

SE-C Comp,Viit,Pune

***Problem Statement-*** *Implement a Pizza Order System using a Queue DS.*

***CODE:***

#include<iostream>

#include<exception>

using namespace std;

class FullQueueException:public exception{

public:

const char\* what()const throw(){

return "FullQueueException";

}

};

class EmptyQueueException:public exception{

public:

const char\*what()const throw(){

return "EmptyQueueException";

}

};

#define DEFAULT\_QUEUE\_SIZE 5

class Queue{

private:

int \*arr;

int elements,queue\_size,head,tail;

public:

Queue(int size=DEFAULT\_QUEUE\_SIZE){

if(size<0)

throw std::bad\_array\_new\_length();

arr=new int[size];

queue\_size=size;

elements=0;

head=tail=0;

}~Queue(){

delete []arr;arr=NULL;

cout<<"Queue Destructed\n";

}

bool isEmpty(){

return elements==0;

}

bool isFull(){

return elements==queue\_size;

}

int deQueue();

void enQueue(int element);

void printQueue();

};

void Queue::enQueue(int element){

if(!isFull()){

arr[tail]=element;

tail=(tail+1)%queue\_size;

elements++;

}else throw FullQueueException();

}

int Queue::deQueue(){

if(!isEmpty()){

int element=arr[head];

head=(head+1)%queue\_size;

elements--;

return element;

}else throw EmptyQueueException();

}

void Queue::printQueue(){

int printer=head,iterations=elements;

if(iterations!=0)

while(iterations--){

cout<<arr[printer]<<" ";

printer=(printer+1)%queue\_size;

}

else cout<<"Empty Queue";

cout<<endl;

}

void printMenu(){

cout<<"1)Place Order\t";

cout<<"2)Deliver Order\t";

cout<<"3)Print Pending Orders\t";

cout<<"4)Close for the Day!\nChoice : ";

}

int main(){

int orderSize,choice,element;

cout<<"Enter Max Orders for the Day\t";cin>>orderSize;

try{

Queue pizzaStore(orderSize);printMenu();

do{

cin>>choice;

switch(choice){

case 1:{

cout<<"Enter Order No ";cin>>element;

try{

pizzaStore.enQueue(element);

}catch(FullQueueException& e){

cout<<e.what()<<" was thrown\n";

}break;

}

case 2:{

try{

element = pizzaStore.deQueue();

cout<<"Order "<<element<<" was Delivered\n";

}catch(EmptyQueueException& e){

cout<<e.what()<<" was thrown\n";

}

break;

}

case 3:{

pizzaStore.printQueue();break;

}

case 4:{break;}

}

}while(choice!=4);

}catch(exception &e){

cout<<"Exception "<<e.what()<<" was thrown"<<endl;

}

return 0;

}

***Output:***

