项目说明文档

数据结构课程设计

——银行业务

作 者 姓 名： 罗佳瑞

学 号： 1952528

指 导 教 师： 张颖

学院、 专业： 软件学院 软件工程

同济大学

Tongji University

目 录

[1 分析 1](#_Toc29397)

[1.1 背景分析 1](#_Toc6698)

[1.2 功能分析 1](#_Toc26968)

[2 设计 2](#_Toc4920)

[2.1 数据结构设计 2](#_Toc23182)

[2.2 类结构设计 2](#_Toc21533)

[2.3 成员与操作设计 2](#_Toc23915)

[2.4 系统设计 2](#_Toc29055)

[3 实现 3](#_Toc12155)

[3.1 创建功能的实现 3](#_Toc10315)

[3.1.1 创建功能流程图 3](#_Toc321)

[3.1.2 创建功能核心代码 3](#_Toc22813)

[3.2 输出功能的实现 4](#_Toc25219)

[3.2.1 输出功能流程图 4](#_Toc10192)

[3.2.2 打印功能核心代码 4](#_Toc13847)

[3.3 总体系统的实现 5](#_Toc10411)

[3.3.1 总体系统流程图 5](#_Toc4326)

[3.3.2 总体系统截屏示例 6](#_Toc4031)

[4 测试 7](#_Toc17988)

[4.1 功能测试 7](#_Toc15478)

[4.1.1 正常测试，A窗口人多 7](#_Toc13262)

[4.1.2 正常测试，B窗口人多 7](#_Toc2663)

[4.2 边界测试 8](#_Toc12652)

[4.2.1 最小N 8](#_Toc23009)

# 1 分析

## 1.1 背景分析

生活中，我们在银行需要取号到不同窗口办理业务，常常遇到排队的情况。一个良好的派号预测系统十分有意义，应该能够为用户办理业务顺序提供一定的预测信息。

这里我们使用计算机模拟一简易的银行业务情景：设某银行有A，B两个业务窗口，且处理业务的速度不一样：其中A窗口处理速度是B窗口的2倍——当A窗口每处理完2个顾客是，B窗口处理完1个顾客。给定到达银行的顾客序列，按照业务完成的顺序模拟输出顾客序列。（假定不考虑顾客信后到达的时间间隔，并且当不同窗口同时处理完2个顾客时，A窗口的顾客优先输出。）

## 1.2 功能分析

作为一个最简易的银行业务模拟系统，其至少应该具有输入顾客序列、按条件重新排列序列、输出用户业务完成序列等功能。

其中，输入顾客序列：输入一数字N（N<=1000）为顾客总数，后面跟着N位顾客的编号（一行正整数），数字间以空格分隔。

按条件重新排列序列：按照编号为奇数的顾客进入A窗口办理业务，为偶数的顾客则去B窗口的原则，将元素一次性加入序列后，按A序列每输出两位顾客，B序列输出一位的规律进行循环输出，直至一序列输出完成，将另一序列直接输出

输出用户业务完成序列：按照业务处理完成的顺序输出顾客的编号，数字键以空格分隔。

# 2 设计

## 2.1 数据结构设计

如上功能分析所述，该系统要求序列进行“先入先出”的输入输出操作，因此考虑使用队列数据结构进行两序列的存储。同时，为提高内存空间利用率及扩大可用空间，采取循环队列的方式进行存储。

## 2.2 类结构设计

队列类（Queue）应含有成员：前指针（f），尾指针（rear），队列元素数目（count）、储存空间（elements），应含有基本操作：判空（empty）、判满（full）、返回大小（size）、入队（push）、出队（pop）、取队首元素（front）等。

## 2.3 成员与操作设计

**队列类（Queue）**

**私有成员：**

int rear, f,count;//尾指针，前指针，队列元素数目

T\* elements;//储存空间

**公有操作：**

bool Queue::empty() ;//判断是否为空

bool Queue::full() ;//判断是否已满

int Queue::size() ;//返回队列当前大小

void Queue::push(T element) ;//入队

void Queue::pop();//出队

T Queue::front() ;//取队首元素

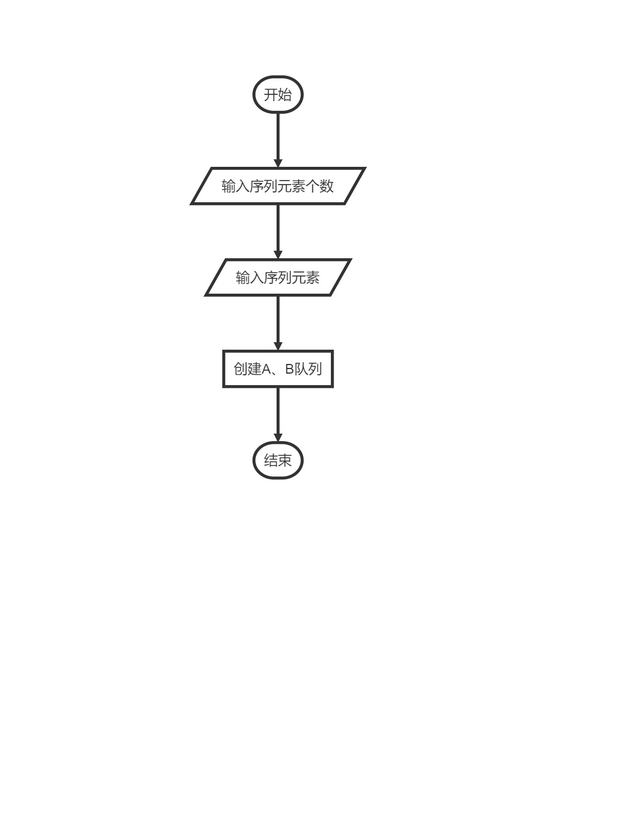
## 2.4 系统设计

系统首先提示用户输入，完成对两个队列A、B的创建和输入数据工作，根据规则输出打印结果链。

# 3 实现

## 3.1 创建功能的实现

### 3.1.1 创建功能流程图



### 3.1.2 创建功能核心代码

int N, num;//顾客总数+顾客序号

Queue<int> A, B;//A、B窗口队列

cout << "请输入顾客总数、顾客序号（空格隔开）:" << endl;

cin >> N;

cout << "业务完成顺序:" << endl;

for (int i = 0; i < N; i++) {

cin >> num;

if (num % 2) { // 根据序号奇偶加入A（奇）或B（偶）序列

A.push(num);

}

else {

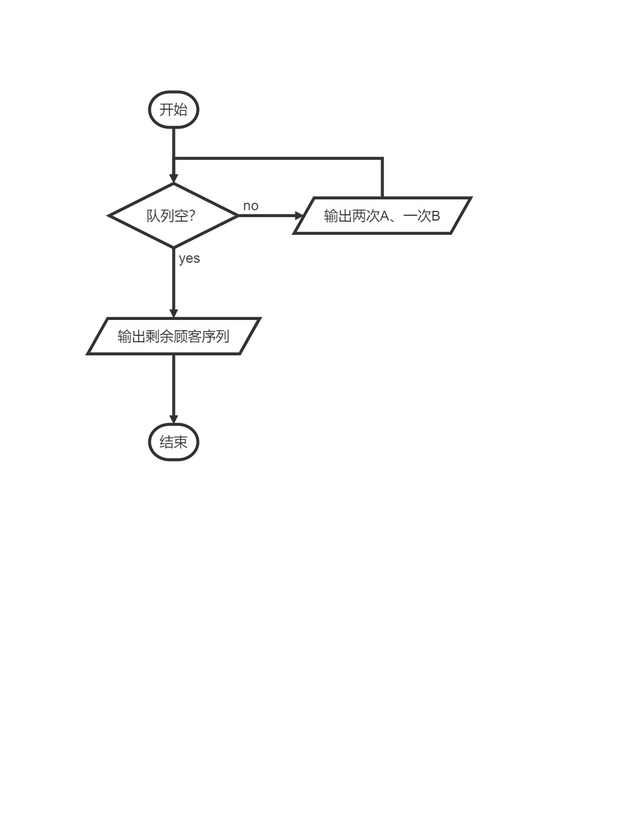
B.push(num);

}

}

## 3.2 输出功能的实现

### 3.2.1 输出功能流程图



### 3.2.2 打印功能核心代码

while (!A.empty() || !B.empty()) {

for (int i = 0; i < 2; i++) {//A输出两次

if (!A.empty()) {

if (B.empty() && A.size() == 1) {

cout << A.front();

}

else {

cout << A.front() << ' ';

}

A.pop();

}

}

if (!B.empty()) {//B输出一次

if (A.empty() && B.size() == 1) {

cout << B.front();

}

else {

cout << B.front() << ' ';

}

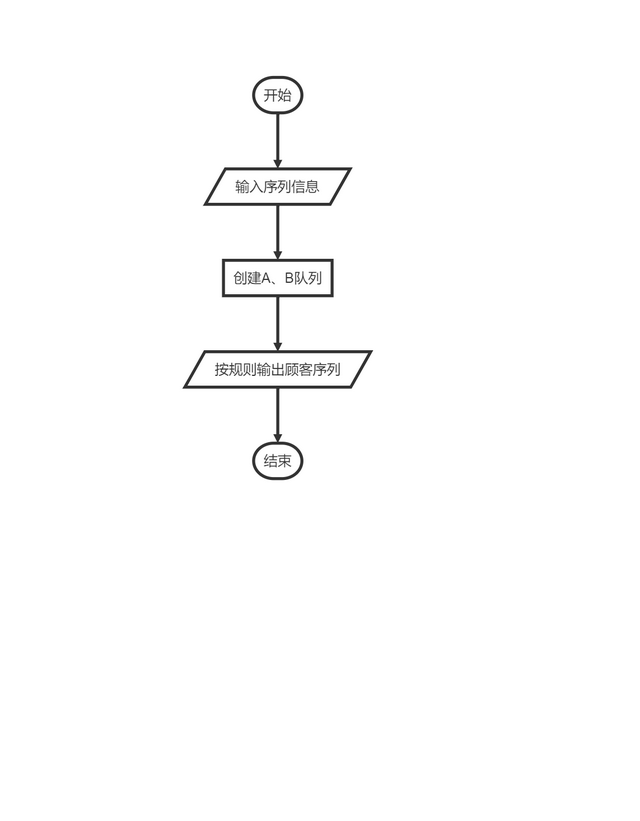
B.pop();

}

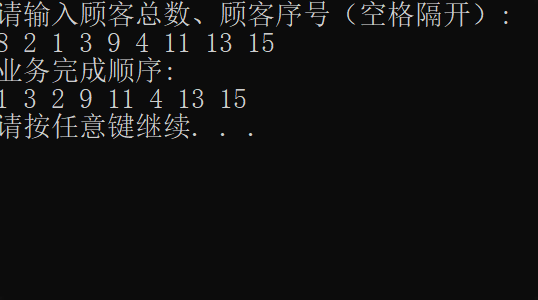
}

## 3.3 总体系统的实现

### 3.3.1 总体系统流程图



### 3.3.2 总体系统截屏示例



# 4 测试

## 4.1 功能测试

### 4.1.1 正常测试，A窗口人多

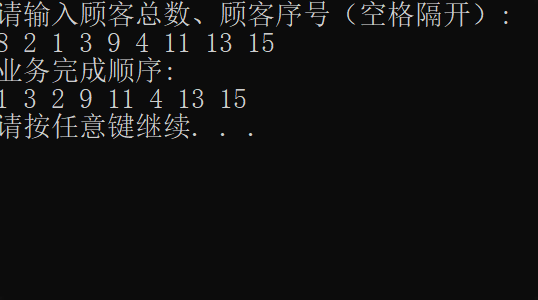
**测试用例：**

8 2 1 3 9 4 11 13 15

**预期结果：**

1 3 2 9 11 4 13 15

**实验结果：**



### 4.1.2 正常测试，B窗口人多

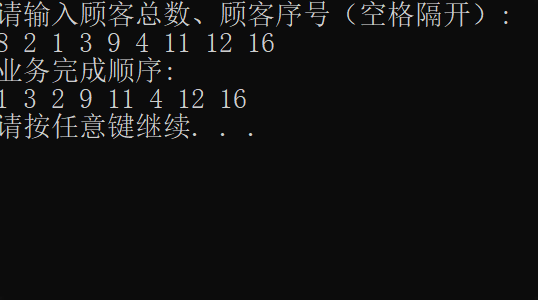
**测试用例：**

8 2 1 3 9 4 11 12 16

**预期结果：**

1 3 2 9 11 4 12 16

**实验结果：**



## 4.2 边界测试

### 4.2.1 最小N

**测试用例：**

1 6

**预期结果：**

6

**实验结果：**

