银行业务

作 者 姓 名： 王凌

学 号： 1951504

指 导 教 师： 张颖

学院、 专业： 软件工程 \_\_

同济大学

Tongji University

目录

[1 分析 2](#_Toc25528)

[1.1 项目要求 2](#_Toc16262)

[1.2 功能要求 2](#_Toc7834)

[2 设计与实现 3](#_Toc23381)

[2.1 Node类设计与实现 3](#_Toc17133)

[2.2 LinkList类设计与实现 3](#_Toc28599)

[2.2.1自定义构造函数 4](#_Toc18393)

[2.2.2自定义析构函数 4](#_Toc18536)

[2.2.3成员函数Empty()，GetLength()，GetData(int) 4](#_Toc32396)

[2.2.4成员函数ListDelete(int) 6](#_Toc31899)

[2.2.5成员函数ListInsertTail(Node\*) 7](#_Toc2345)

[2.2.6成员函数ClearLinkList() 7](#_Toc22205)

[2.3 主函数设计与实现 7](#_Toc15863)

[3 测试 9](#_Toc23387)

[3.1 功能测试 9](#_Toc291)

[3.1.1 测试1 9](#_Toc28767)

[3.1.2 测试2 9](#_Toc12704)

[3.1.3 测试3 9](#_Toc28405)

[3.1.4 测试4 10](#_Toc27102)

[3.1.5 测试5 10](#_Toc32216)

[3.1.6 测试6 10](#_Toc7032)

# 

# 1 分析

## 项目要求

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----即当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，请按照业务完成的顺序输出顾客序列。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 功能要求

输入说明：输入为一行正整数，其中第一数字N（N<=1000）为顾客总数，后面跟着N位顾客的编号。编号为奇数的顾客需要到A窗口办理业务，为偶数的顾客则去B窗口。数字间以空格分隔。

输出说明：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，但是最后一个编号不能有多余的空格。

测试用例：

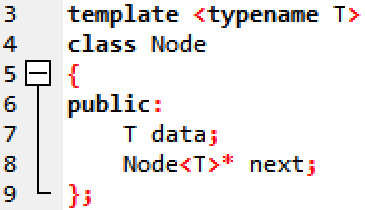
|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 输入 | 输出 | 说明 |
| 1 | 8 2 1 3 9 4 11 13 15 | 1 3 2 9 11 4 13 15 | 正常测试，A窗口人多 |
| 2 | 8 2 1 3 9 4 11 12 16 | 1 3 2 9 11 4 12 16 | 正常测试，B窗口人多 |
| 3 | 1 6 | 6 | 最小N |

# 

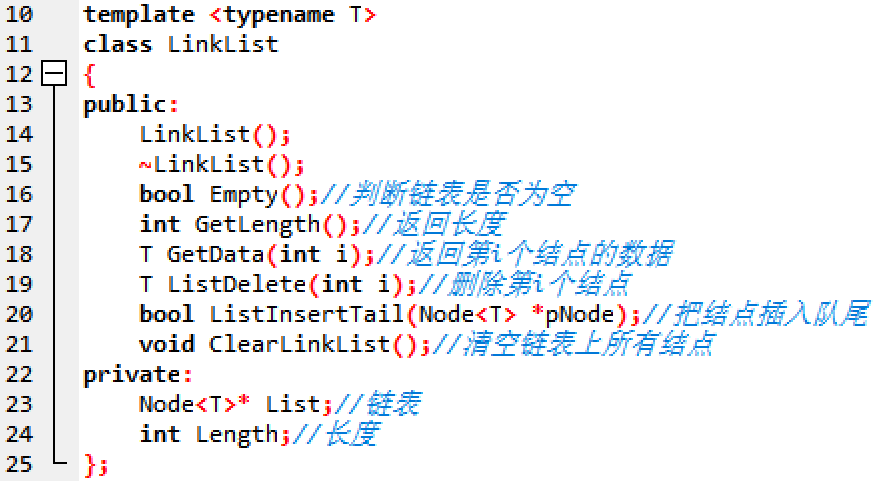
# 2 设计与实现

## 2.1 Node类设计与实现

Node类是一个模板类，每一个Node对象为链表上的基础节点。它能够存储一个T类型的数据和一个指向下一个节点的指针变量。



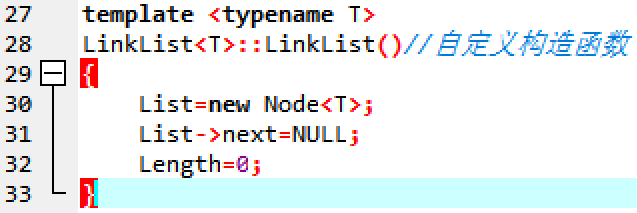
## 2.2 LinkList类设计与实现



有两个私有成员变量List数组和链表长度Length，起到了封装的作用。由于是私有成员变量，而链表上某些结点的值和链表的总长度无法在类外访问，所以又因此设置了两个公共成员函数GetData()和GetLength()来访问结点的值和链表的总长度。

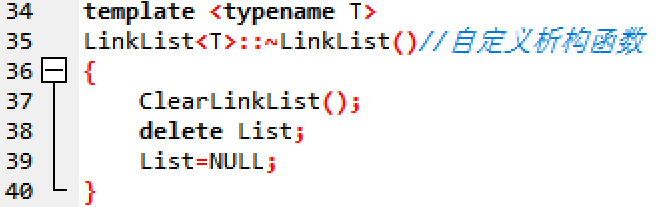
每一个成员函数的作用依次是：Empty()函数用来判断链表是否为空；GetLength()用来返回链表长度；GetData(int i)用来返回第i个结点的数据；ListDelete(int i)用来删除链表上第i个结点； ListInsertTail(Node<T> \*pNode)用来把结点插入链表尾部；ClearLinkList()用来清空链表上所有结点。

### 2.2.1自定义构造函数



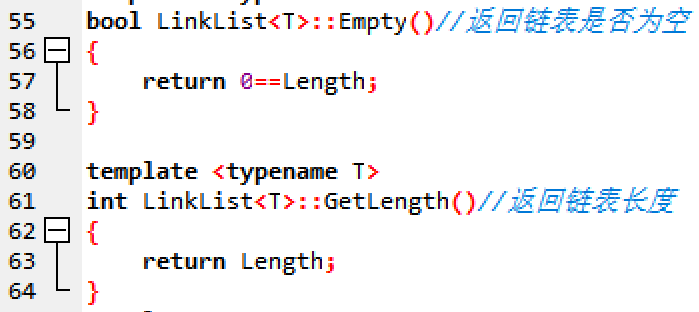
在链表头部生成一个头指针，注意，这个头结点是不存任何数据的，只标记链表的头部。初始化头节点的next为空，链表的长度为空。这样的目的是减少删除节点的分类讨论，详细的说明在2.2.4结点删除成员函数中，此处不再赘述。

### 2.2.2自定义析构函数



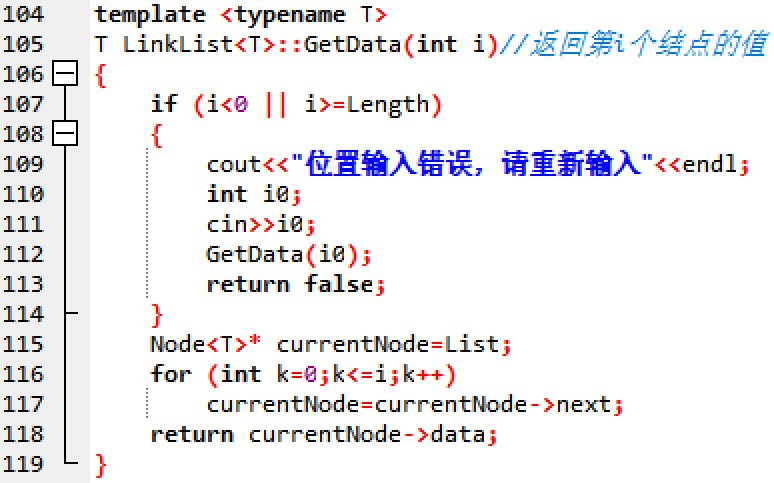
首先调用ClearLinkList()函数，清除除了List表头外的所有结点，再删除List表头结点，将表头置空。这样就删除了链表上的所有元素。

### 2.2.3成员函数Empty()，GetLength()，GetData(int)



当长度为0时Empty()函数返回1，否则返回0。

封装之后要读取私有成员变量Length需要设计一个GetLength()函数。



返回第i个结点的值，若输入i不在长度范围之内的话，报错且递归调用自身重新输入直到符合条件为止。循环i+1次，找到第i个结点，返回该结点的数据。

### 2.2.4成员函数ListDelete(int)

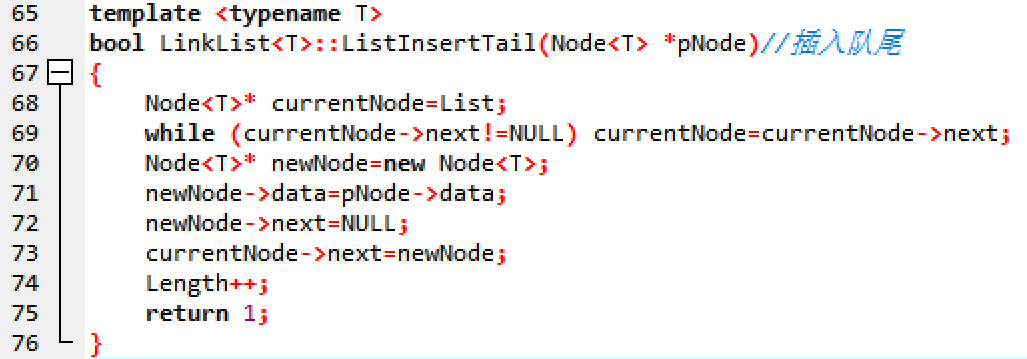


删除第i个结点，若输入i不在长度范围之内的话，报错且递归调用自身重新输入直到符合条件为止。

之后记录两个指针变量，currentNodeBefore变量记录当前结点之前一个结点的地址，currentNode结点记录当前结点地址。循环i+1次找到第i个结点和第i-1个结点。将第i-1个结点的后继（next）改为第i个结点的后继（next）这样原链表就删除了第i个结点。但是我们还需要得到第i个结点的值，并且删除第i个结点的存储空间碎片，所以我们记录第i个结点的数据，之后删除第i个结点，并且令链表总长度减一，这样就消除了删除结点的影响。最后返回所记录的第i个结点的值。

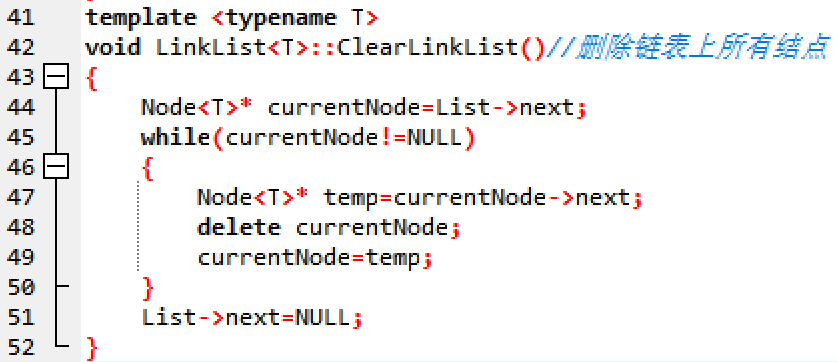
在张老师上课的PPT中链表中删除结点要分两种情况考虑，第一种是表头删除，第二种是表中和表尾。那为什么我只有一种情况呢？主要是因为我的链表设置了一个空表头List结点，这个结点不计数据，只记录表头位置，那么之后记录数据的结点相当于是表中或者是表尾了，所以不需要分情况讨论，节省了代码量与时间。

### 2.2.5成员函数ListInsertTail(Node\*)



先用while循环从表头一直循环到表尾，先将该结点的数据存储，并将后继next改为空，之后再将原来最后一个结点的后继next改为要插入队尾的这个结点。之后链表总长度加一，返回1，说明结点插入链表队尾成功。

### 2.2.6成员函数ClearLinkList()



用while循环一直删除表头List的后继结点一直到表头的后继结点为空。最后剩下表头结点List。

这个函数是要搭配自定义析构函数一起用的，析构函数清空链表上所有含有数据的结点之后，再删除List表头结点，清空链表类的所有存储碎片。

## 2.3 主函数设计与实现

首先，创建两个LinkList链表，代表了A，B两个柜台，之后，用一个循环一边读去编号一边根据编号分别存储进A、B两个链表之中。编号为奇数插入链表A队尾 ，去柜台A排队；编号为偶数插入链表B队尾，去柜台B排队。

然后进入while循环，循环中，有三个判断及其操作。首先做两遍，“如果队列A不为空，则输出A队列的头节点，并且删除当前结点”的操作。之后做一遍，“如果队列B不为空，则输出B队列的头节点，并且删除当前结点”的操作。这就是在模拟现实中的场景，当A窗口每处理完2个顾客的时候，B窗口处理完1个顾客，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。

一直循环到两个队列都为空则退出循环，否则继续循环。

在这段循环中有两点需要注意：

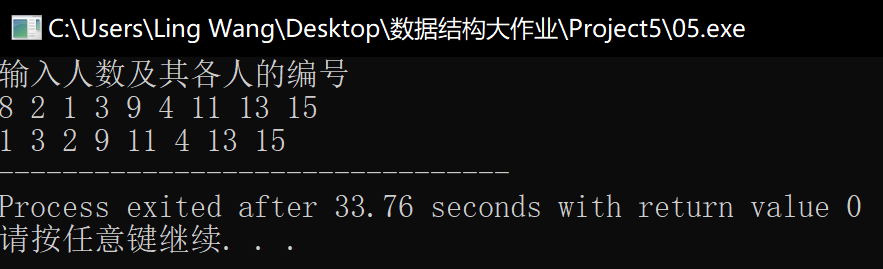
1. 三个操作都要判断链表是否为空的原因是，如果其中一个队列为空的话,那么就需要把另一个队列的所有元素都输出了；
2. 第二，题目中要求输出的最后一个数据的末尾不能有空格，所以我们需要判断当前输出的元素是不是所有元素中的最后一个元素。那么应该如何判断呢？因为长度不是负数，所以当listA的长度加listB的长度为1的时候说明还剩最后一位没有输出，此时输出的数据不带空格。



# 3 测试

## 3.1 功能测试

### 3.1.1 测试1



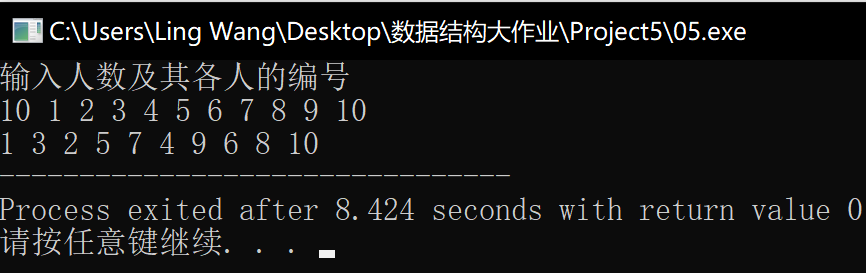
### 3.1.2 测试2



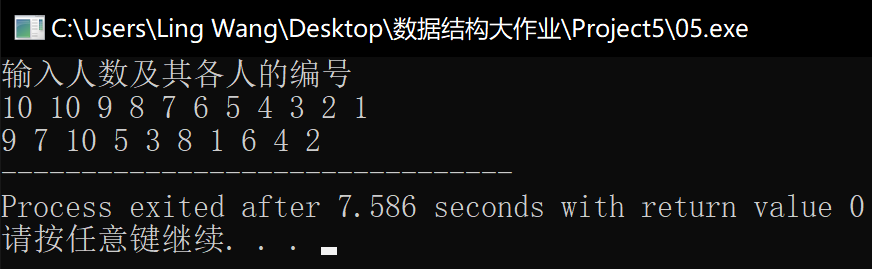
### 3.1.3 测试3



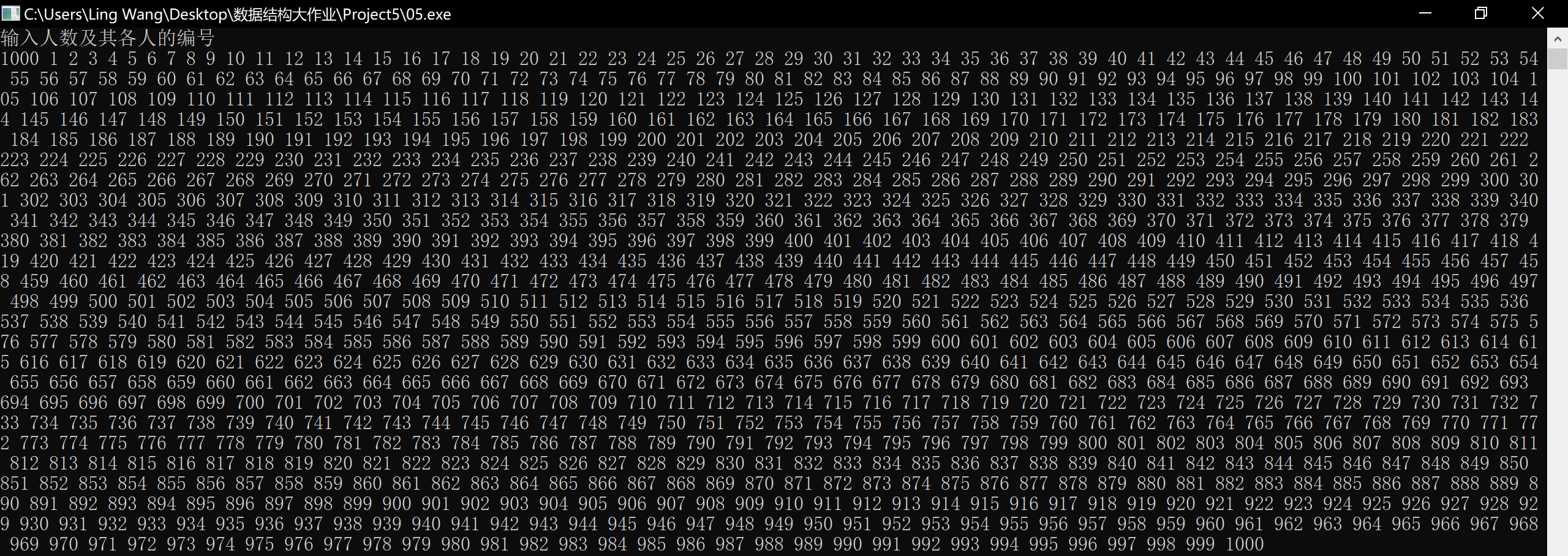
### 3.1.4 测试4



### 3.1.5 测试5



### 3.1.6 测试6



因为数据规模最大为1000，不妨写一个函数生成1~1000然后进行验证。

结果为：

