**数据结构实验报告**

**（一）**

**学院 自动化学院**

**学号 08022311**

**姓名 陈鲲龙**

**日期 2023-10-29**

**实验内容**

在文具店的日常经营过程中，存在对各种文具管理问题。当库存文具不足或缺货时，需要进货。日常销售时，需要出库。当盘点货物时，需要查询货物信息。请根据这些要求编写程序完成库存文具的管理功能。

**实验要求**

1)要求实现的库存文具管理功能：

a)实现进货操作；

b)实现出库操作，库存不足时提醒；

c)实现查询操作；

d)货物盘点

2)表示货品信息的数据结构设计要合理；程序功能模块的划分要适当；实验报告中多使用流程图来描述算法结构。

3)调研线性表在实际生活中的其他应用，具体阐述并给出实现思路（可不用代码实现）。

1. **需求分析**

**设计任务：**本实验的需求是用线性表实现一个货品管理的程序，要求货品能进货出货，能查找能盘点，货物不足有提示。

**程序功能：**以上任务内容功能均达成，并且还能将数据保存进.txt文件，出货时能计算销售额以及利润。

**输入规定：**使用者输入时需要根据提示输入，例如商品种类、编号、数量、单价等等，在价格和数量等输入数字的位置不可输入字母，特别在入库时间部分，年、月、日每输入一个都要回车

**输出形式：**全基于cout打印程序对应内容

**测试：**先录入若干商品，再进行按编号、按名称查找测试，最后出库测试，并测试库存不足时仍出库时触发的缺货提示。

1. **概要设计**

说明程序中用到的所有抽象数据类型定义、主程序流程以及各模块之间的层次（调用）关系。

**本实验架构：**

**底层数据结构：**底层数据结构为链表，会作为中层GoodsManage类的私有成员，由于货物管理频繁的需要新增、删减、查找操作，所以选择链表的结构来记录货品的库存情况以及对货品的管理操作。

链表的元素类型也很关键，是自定义的类型Goods,它的成员能记录货品的名称、编号、数量、价格、入库时间等信息，这些信息在查找和出库函数中都要访问，以及链表的下一节点指针。

**中层处理层:** GoodsManage这个自定义的类，是本程序的核心。

包含四个私有成员，负责记录货品数量以及储存库存情况的链表的头节点，

还包含9大函数：

DisplayMainMenu();//主菜单显示

AddGoodsInfo();//添加商品信息

DisplayGoodsInfo();//浏览商品信息

SearchByCode();//按照商品编号搜索商品信息

SearchByName();//按照商品名称搜索商品信息

DeleteGoodsInfo();//删除商品信息

SellGoodsInfo();//出售商品信息

SaveGoodsInfo();//保存商品信息

Run();//运营封装（主函数）

它们分别完成了货品管理中的各种功能，参考其各自的注释，具体详介见第3部分：详细设计。

**上层：**主函数仅两行代码两件事：1.创建GoodsManage类的实例，名叫GM；2.调用GM实例的Run()函数。

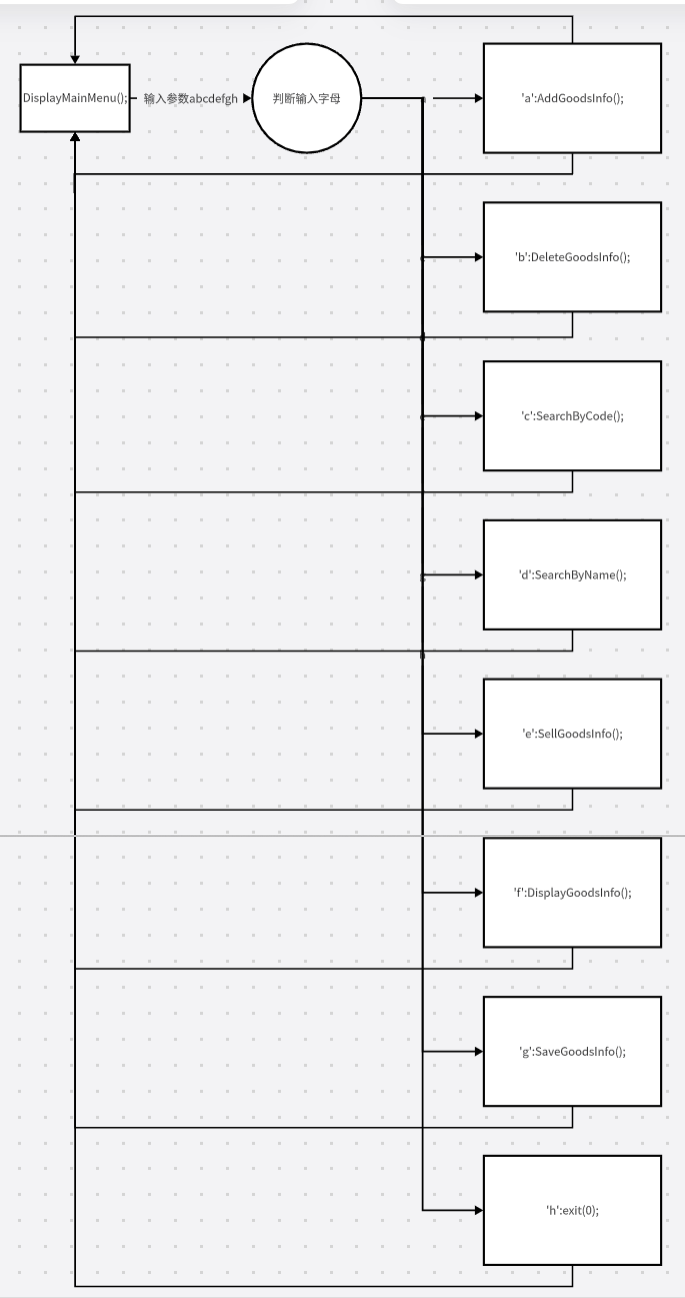
1. **详细设计**

程序核心是GoodsManage类中的9个函数。

**Run():**

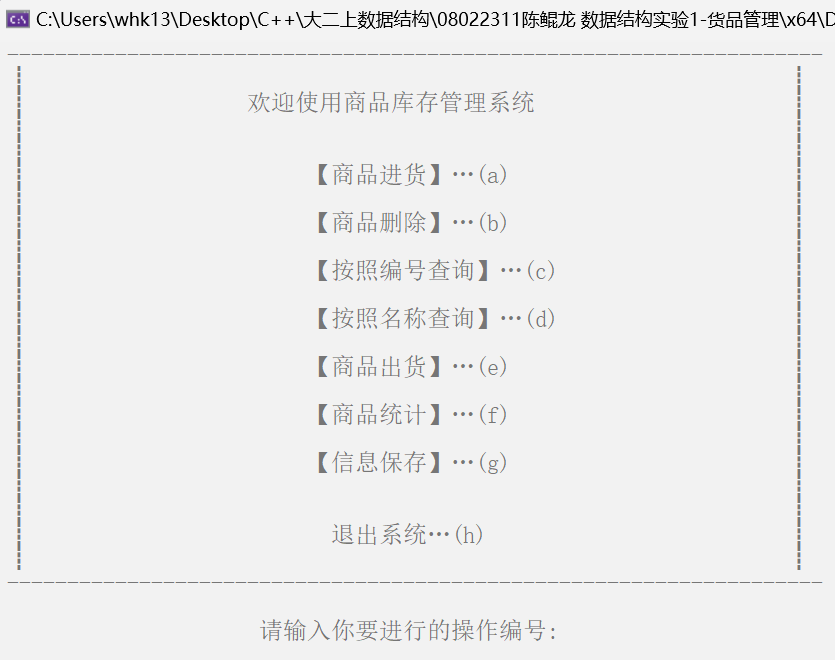
先从主函数中唯一调用到的Run()函数开始解释，Run()函数是对整个程序运行的封装函数，进入函数后先调用DisplayMainMenu()函数向用户显示主菜单，然后进入循环，根据用户输入的字母决定调用方的函数或者退出循环。

Run()函数流程路如下：(h选项跳出循环/退出程序)



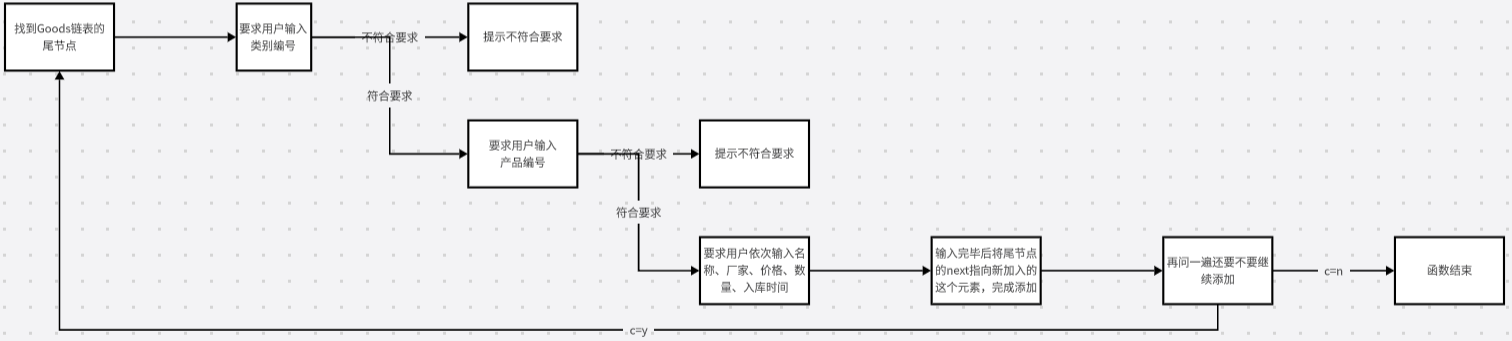
**DisplayMainMenu():**

**cout逐行打印出初始的选项界面：**



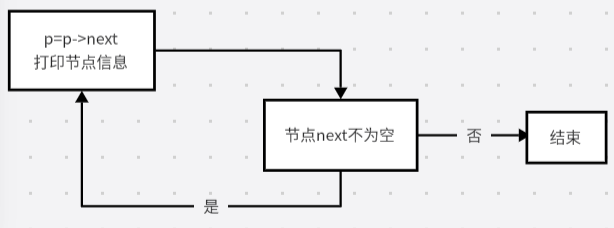
**AddGoodsInfo()：**

AddGoodsInfo()函数负责货品入库的操作，先找到储存库存的Goods链表的尾节点，然后要求用户输入货品的相关信息，注：类别编号会检测有没有超出范围，货品编号会检测重不重复，信息根据屏幕上打印的提示输入完毕后，将新的节点添加在链表尾部，char c用于判断是否要连续入库货物，每次添加节点的过程结束会问还要不要继续添加，如果让c保持y则循环添加节点的过程，如果选n,则退出循环，入库结束。



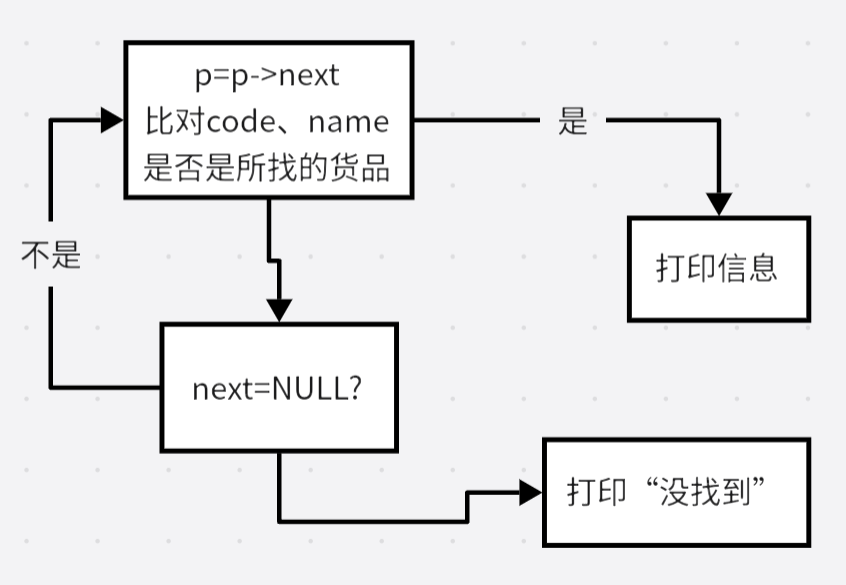
**DisplayGoodsInfo()：**

DisplayGoodsInfo()即用循环，遍历链表，一一打印货品信息，直到尾节点next为空指针NULL。



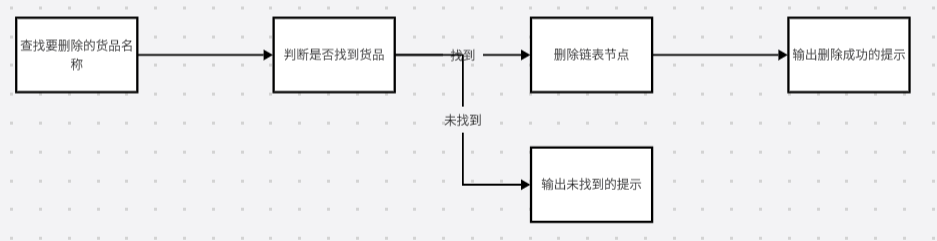
**SearchByCode() & SearchByName()：**

由于链表的元素是Goods这个自定义的类，它的成员包含了编号和名称，所以我们可以遍历链表，通过比较这两个信息一不一致来检索查询我们要找的货品。



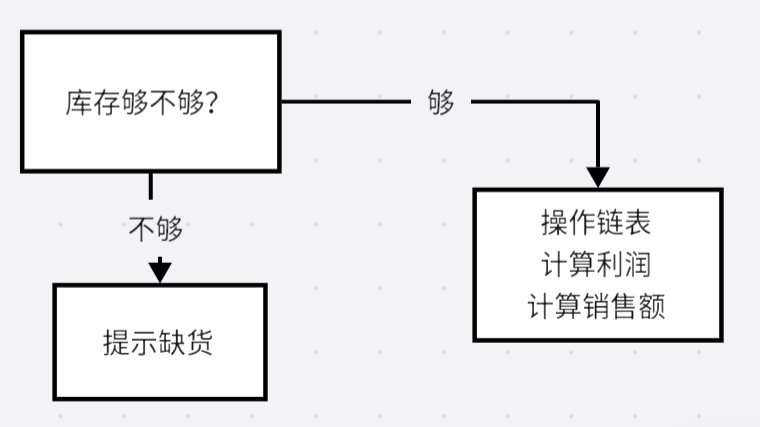
**DeleteGoodsInfo()：**

DeleteGoodsInfo()是删除货品的函数，和查找的原理一样，先找到这个货品，之后再做一个在链表中删除节点的操作。



**SellGoodsInfo()：**

SellGoodsInfo()是卖出货品时的操作函数，其逻辑很简单，有货就出，缺货提示，但代码有些冗长，主要因为要进行的判断以及输入输出的信息较多，比如判断有没有这个货，以及这个货够不够，在出货时要输入出库时间年月日。



**SaveGoodsInfo()：**

SaveGoodsInfo()通过ofstream output("货物信息.txt", ios::out);//定义输出文件"货物信息.txt"，将数据通过循环（p=p->next)保存进.txt文件,这样就不用重复录入了。

1. **调试分析**
2. 调试过程中遇到的问题及其解决办法，以及对整个设计和实现过程的回顾讨论与分析；

在第3部分中提到的很多判断语句其实都是遇到问题后添加的，写程序的时候不仅要写自己想达成的功能，还要有保护性的变量和判断语句，用来防止空指针的出现以及用户不按要求输入的情况。

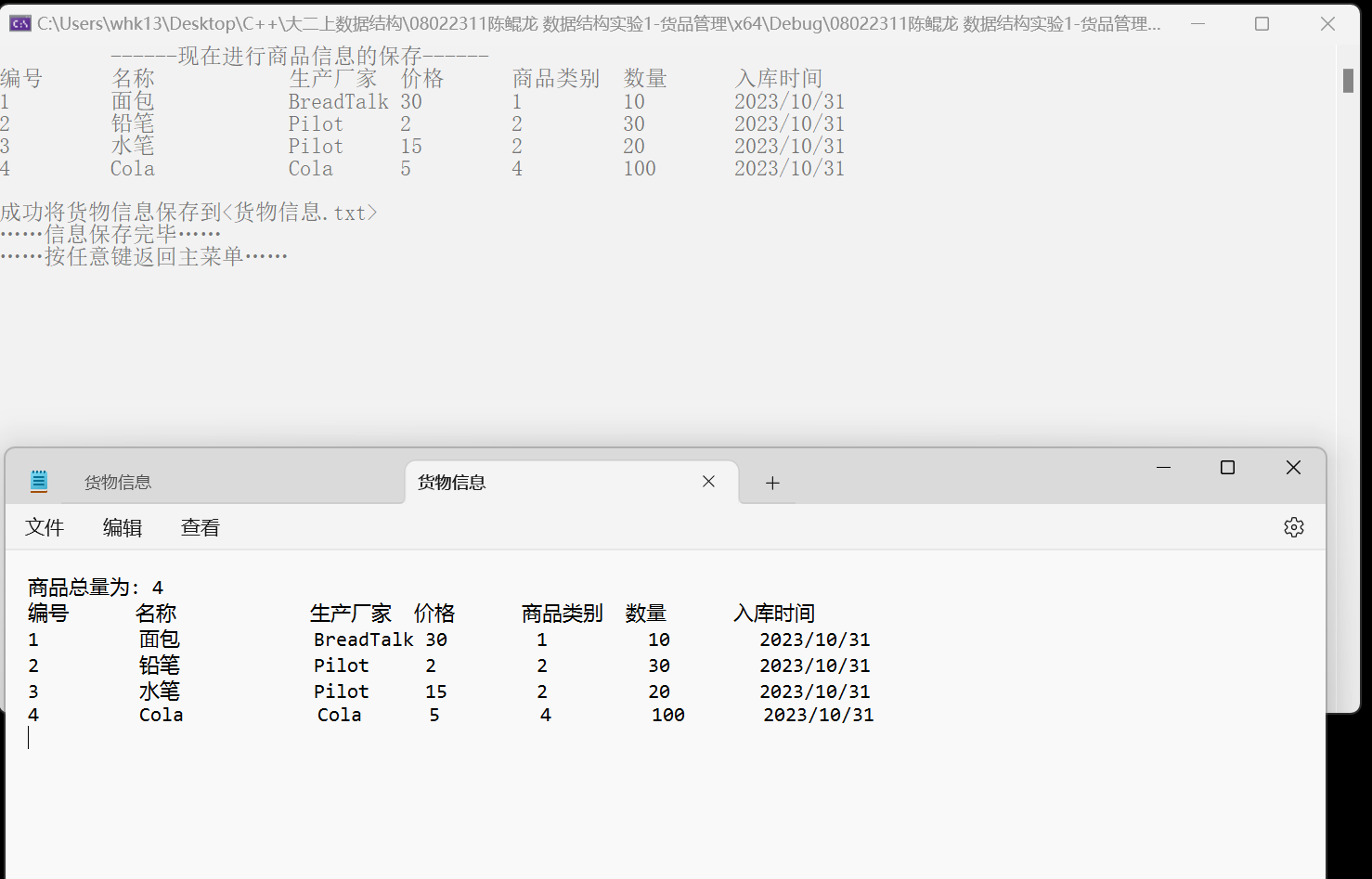
1. 算法的时空分析（包括基本操作和其他算法的时间复杂度、空间复杂度分析）；

由于本实验我的数据结构使用的是链表，链表的操作几乎都有找尾指针，所以我的时间复杂度是O(n)。

1. **测试结果**

列出测试结果，包括输入和输出。

先输入一些货品：



再验证各功能：

货品盘点：



按编号查找：



按名称查找：



货品出库：



1. **实验总结**

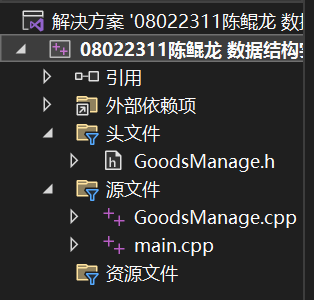
本次实验我主要是通过实战，将链表的一些操作复习了一下，相较于之前只会写一个cpp文件主，在一次次大大小小的项目程序中，我对函数、对自定义类的封装也更加熟练了，能将一些共性的东西抽象出来，自定义一个类型来处理一类变量，代码合理性和保护性写法我觉得可以再做得更好些，可能还有没发现的bug，算法上也有优化空间。

1. **附录**

列出源程序文件名清单。

（所列源程序文件打包作为实验报告的一个重要组成部分，作为实验评分依据）

包含c++类：GoodsManage;以及main.cpp



1. **应用调研**

其实我觉得生活中只要没有分叉的数据结构，都能使用线性表，比如各种图书、员工管理系统已经成了很多学生第一个跟着学的项目了，还有一些是根据线性表的特殊性，比如栈“先进后出”的性质，我在做迷宫游戏项目的时候，迷宫生成时，道路就储存在栈中，再比如队列，在自动化导论中，我已经听到过很多次生产者和消费者中间的缓冲，它先进先出就吻合队列的性质。学习了线性表之后，在生活中，我看待一些事物的想法也产生了变化，会不由得去想它的背后会不会是由线性表来达成的，比如，我就想分享一下英雄联盟云顶之弈的玩法，从线性表的角度去看它的游戏机制，让我好奇它背后代码是如何实现的，它的机制简单来说就是玩家有9个备战席位，游戏中玩家会抽牌，抽到的牌放在备战席上，每三张相同的牌能合成一张这张牌的二星版本，三张二星能合成三星(即一共九张牌)，如果说备战席是个线性表的话，其实他的操作是非常多且频繁的，如果是我来做，我还是会用链表，因为备战席上的牌可改变、交换位置，这就涉及到了数据的移动，其次抽出三张相同的牌就要自动合成，这就涉及到了线性表元素的访问和更改内容；再说回暑期学校做的迷宫游戏，其中迷宫的信息、玩家的位置等信息其实都是记录在了线性表中，总之线性表的使用是很广泛的，重要的是根据需求选对线性表的类型，是栈、队列、链表还是最基础的数组等等。

