**技术报告**

姓名： 学号： 班级：教务X班

（本模板以实现一个仓库管理系统为例）

（这个技术报告模板并不完整，你们要自行完善）

## 需求分析：

仓库管理系统的功能如下：

1. 在仓库进货时，如果仓库中没有此商品，则为仓库增添新的商品项目
2. 在仓库进货时，如果仓库中已有此商品，则增加此商品的库存量
3. 在仓库出货时，减少对应商品的库存量
4. 在仓库出货时，如果这是货物是此商品的最后一批货（库存量为0），则删除仓库中此商品项目
5. 查询功能：可以随时查看当前仓库的库存，包括商品名和剩余量

## 实现思路:

功能结构图：

仓库管理系统

进货

出货

增加库存

新增商品

减少库存

删除商品

查询

显示库存

查找商品

本仓库管理系统主要分为出货、进货、查询三大模块，分别对出货、进货和查询的操作进行管理。

进货模块中又细分为增加库存和新增商品子功能，当进货时，若此商品在仓库中没有库存，则在仓库库存条目中新增此商品项目，若已有此类商品，则根据进货量增加对应的库存量。

出货模块中又细分为减少库存和删除商品子功能，当出货时，减少对应商品的数目，注意到当库存不足时，出货失败，且若出货成功并且库存为0时，删除仓库目录中此商品项目。

查询模块，分为显示库存和查找某一商品是否存在库存及其库存量

## 数据设计：

struct Goods {

char name[100]; //记录货物名

int count; //记录货物数量

};

struct goods GOODS[];//用于存储货物信息

## 函数设计：

进货：

//进货，对应进货模块，表示当前进货一批数量为count的name商品

bool add\_goods(char name[], int count);

//更新库存信息，对应增加库存子功能，对name商品新增count数量

bool increase\_count (char name[]，int count);

//更新库存列表，对应新增商品子功能，新增name商品且初始数量为count

bool add\_to\_list(char name[], int count);

出货：

//出货，对应出货模块，表示当前出货一批数量为count的name商品

bool delete\_goods(char name[], int count);

//更新库存信息，对应减少库存子功能，对name商品减少count数量

bool increase\_count(char name[], int count);

//更新库存列表，对应删除商品子功能，删除商品列表中name商品

bool delete\_from\_list(char name[]);

查询：

///显示当前库存列表，包括商品名及其库存量

void show\_goods();

//查看仓库中的name商品

struct Goods find\_goods(char name[]);

//把出库和入库记录写入日志文件夹

void write\_log(char name[], int count,int flag)

//文件读写

//写文件

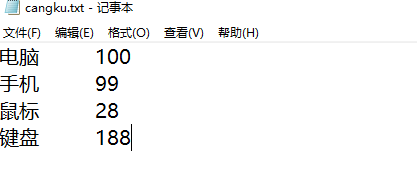
void Save\_file()

//读文件

void Open\_file()

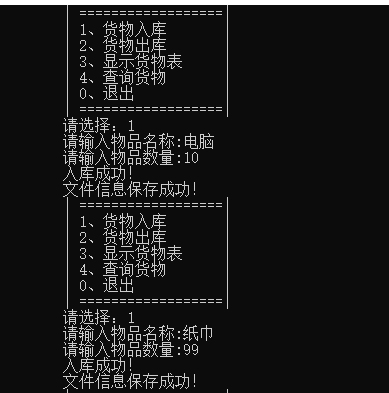
## 输入与输出

1.用作记录仓库数据的文件

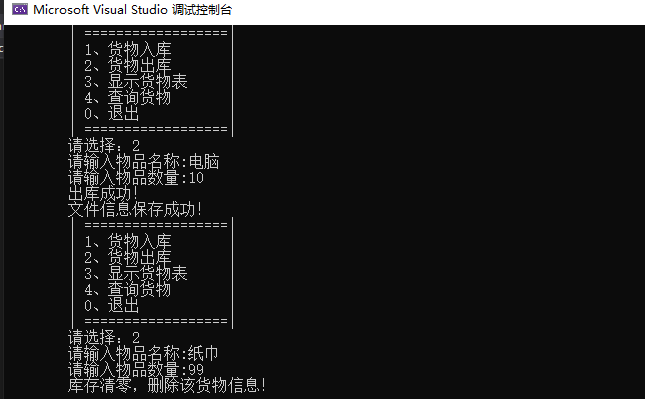


2.程序的入库，出库，以及查询等操作

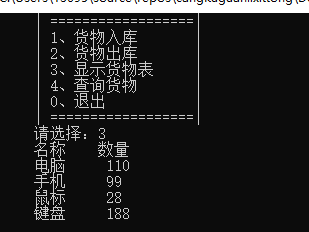
1）入库

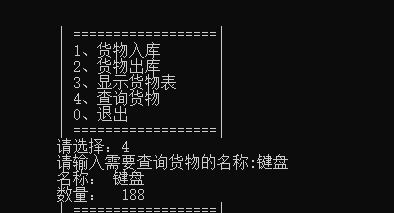


2）出库

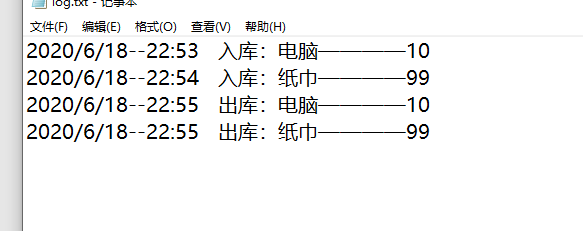


3）查询





3.日志文件



记录了出库和入库的数据

## 实验感想

1.在这次仓库管理系统的实验中，我建立了一个较完整的数据管理系统，这个系统在数据处理方面的函数较为冗杂，需要尽可能的增加函数的复用性，来精简整个系统。同时提升系统的可拓展性和健壮性。避免在增加功能或者修改功能时出现bug或者要大规模重构的现象。

2.加深了我对程序构建的理解，像这次实验中的程序就分为了2层，纯数据处理层用户交互层，程序这样的分层使得程序更好阅读和维护，在将来合作开发的时候可能更为适应团队一些。

3.同时我们在程序设计时，要有探索的心理，要勇于尝试新的方法，思考新的东西。同时也要抱有严谨的态度，去审视自己的程序有无改进的空间，一开始是否能做的更好，或者下次能否设计的更加符合需要。