Project 1 技术报告

17341015 计一陈鸿峥

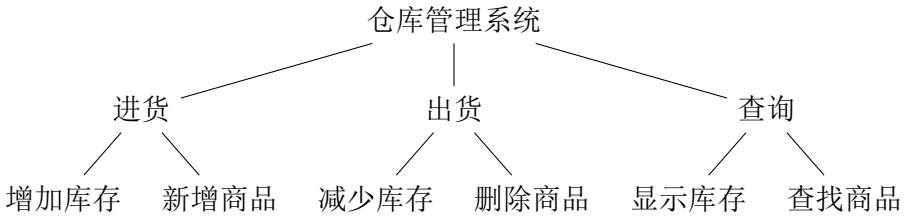
1 需求分析

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

且以上功能均需通过日志文件保存，以防丢失。

2 实现思路



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

1. 进货模块
   细分为增加库存和新增商品子功能。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。

2. 出货模块
   细分为减少库存和删除商品子功能。出货时，若此商品在仓库中没有库存，则报错显示出货失败；若此类商品库存充足，则根据出货量减少对应商品的数目；若出货成功并且库存恰好为0时，删除仓库目录中此商品项目。

3. 查询模块
   细分为显示所有库存和查找某一商品子功能。

最后用文件流实现数据的读取及保存。

3 数据设计

```
class Warehouse
{
public:
        Warehouse(){};
        bool import_goods(string name, int count);
        bool export_goods(string name, int count);
        int find_goods(string name);
        void show_goods();
        void save();
private:
        map <string ,int> data;
        bool increase_count(string name, int count);
        bool add_to_list(string name, int count);
        bool decrease_count(string name, int count);
        bool delete_from_list(string name);
};
```

所有商品数据都被存储在一个 `Warehouse` 类中，且以 `map` 形式存储，其中 `map` 是由商品名字(`string` 类型)到商品库存量(`int` 类型)的映射。至于类方法，只有进货、出货、查询、保存是公有的，其他的方法均设置为私有。

4 函数设计

详情请参见代码，这里仅仅给出每个函数的用途。

4.1 进货

1. 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 Warehouse::import_goods(string name, int count)`
2. 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 Warehouse::increase_count(string name, int count)`
3. 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 Warehouse::add_to_list(string name, int count)`
4. 封装函数
   `void import_process(Warehouse& house)`

4.2 出货

1. 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 Warehouse::export_goods(string name, int count)`
2. 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 Warehouse::decrease_count(string name, int count)`
3. 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 Warehouse::delete_from_list(string name)`
4. 封装函数
   `void export_process(Warehouse& house)`

4.3 查询

1. 显示当前库存列表，包括商品名及其库存量
   `void Warehouse::show_goods()`
2. 查看仓库中名为name的商品，若不存在返回0，否则返回该商品库存量
   `int Warehouse::find_goods(string name)`
3. 封装函数
   `void query_process(Warehouse& house)`

4.4 输入与输出

1. 从文档中读入数据
   `void read_in_data(Warehouse& house)`
2. 保存数据入文档
   `void Warehouse::save()`

4.5 其他

1. 交互界面
   `void warehouse_interface()`
2. 显示菜单
   `void show_manual()`
3. 商品名合法性测试
   `bool validname_test(string x)`