**东北大学计算机科学与工程学院**

**数据结构课程设计报告**

题目 超市商品库存信息管理系统

课题组长 戚子强

课题组成员 吕振峰 张卓

专业名称 计算机科学与技术

班级 计1606

指导教师 马安香

2017年12月

目录

[题目一 超市商品库存信息管理系统 1](#_Toc503211247)

[1 课题概述 1](#_Toc503211248)

[1.1 课题任务 1](#_Toc503211249)

[1.2 课题原理 2](#_Toc503211250)

[1.3 相关知识 3](#_Toc503211251)

[2 需求分析 3](#_Toc503211252)

[2.1 课题调研 3](#_Toc503211253)

[2.2 用户需求分析 4](#_Toc503211254)

[3 方案设计 5](#_Toc503211255)

[3.1 总体功能设计 5](#_Toc503211256)

[3.2 数据结构设计 6](#_Toc503211257)

[3.3 函数原型设计 6](#_Toc503211258)

[3.4 主算法设计 8](#_Toc503211259)

[3.5 用户界面设计 10](#_Toc503211260)

[3.6 输入输出设计 11](#_Toc503211261)

[4 方案实现 15](#_Toc503211262)

[4.1 开发环境与工具 15](#_Toc503211263)

[4.2 程序设计关键技术 15](#_Toc503211264)

[4.3 个人设计实现（按组员分工） 15](#_Toc503211265)

[5 测试与调试 16](#_Toc503211266)

[5.1 个人测试（按组员分工） 16](#_Toc503211267)

[5.2 组装与系统测试 17](#_Toc503211268)

[5.3 系统运行 17](#_Toc503211269)

[6 课题总结 18](#_Toc503211270)

[6.1 课题评价 18](#_Toc503211271)

[6.2 团队协作 18](#_Toc503211272)

[6.3 个人设计小结（按组员分工） 20](#_Toc503211273)

[7 附录 21](#_Toc503211274)

[A 课题任务分工 21](#_Toc503211275)

[B 课题设计文档（光盘） 25](#_Toc503211276)

[C 使用手册（可选） 26](#_Toc503211277)

# 题目一 超市商品库存信息管理系统

## 1 课题概述

### 课题任务

**课程设计任务书**

|  |
| --- |
| **题目：**  超市商品库存信息管理系统 |
| **问题描述：**  对一个中小型超市的库存商品信息管理作一个简单的模拟。 |
| **设计要求：**  设计超市库存商品信息维护管理的模拟程序。  （1）采用顺序表或结构体链表存储结构。  （2）可以随机、文件及人工输入数据。  （3）实现超市商品的创建、分类、出库、入库、修改等管理功能。  （4）实现库存商品的各种查询和统计功能。  （5）实现人机交互界面功能。  （6）其它完善性或扩展性功能。 |
| **指导教师签字：**  **年　　月　　日** |

### 课题原理

数据库建立：全局使用链式表进行存储，链式表节点是单个的商品类，有商品ID、商品名称、商品进价、商品售价、商品数量、商品单位、商品类别和详细的进库出库时间。采用链式表的数据结构存储每个商品类；

入库：由于每个商品的ID是独一无二的，如果数据库中不存在输入的ID，则新建一个商品类，否则在已有的商品类基础上进行修改——包含数量和进出价等的修改。实际上是链式表的插入操作。另外为了方便输入，我们加了一个置空按钮会将所有当前输入置空；

出库：简单的链式表元素修改功能，需要注意的是怕岸段各种可能因素——例如此时的进出价、输入是否合法等等；

查询：这里的查询，由于采用的开发工具是Qt，具有友好的可视化界面，用户在输入各项后所产生的效果是“并”的，即如果用户输入了名称是“苹果”并输入了类别是“电子产品”，那么所查出的结果是“苹果手机”相关而不是水果中的苹果。这实际上在具体实现中假设了用户会输入所有的筛选条件，然后逐个排除的结果，最后将结果以列表的形式可视化地即时出现在用户眼中；

修改：以ID为准，在输入ID后，再输入其他项后如果与已经存在的参数不同则会改为用户输入的参数，这里只有ROOT级用户拥有此权限，原理很简单——在查询的基础上修改链式表的参数即可；

账户：分为三级（游客TOUR、员工EMPOR和超级管理员ROOT），不同权限的用户所看到的界面功能是不同的，这是采用的简单的控件可视化/不可视化实现，默认打开界面是游客界面。实时的时间显示和计时是采用Qt的槽函数实现；

统计中心：分别统计了商品数量、商品总进价、商品总价值和员工数量，这个统计数字也是实时改变的，实现原理也是槽函数；

批量导出：可以把数据库中的数据批量导出为CSV表格文件或者经DES加密的QZQ加密格式文件，实现原理是：CSV文件实际上是文本文件经过逗号和回车分割而成的表格格式；而DES加密则是采用64位，把64位的明文输入块变为64位的密文输出块，它所使用的密钥也是64位，在这里，秘钥是我们直接给定的；

批量导入：支持CSV和DES加密的导入，判断格式后如果是加密文件则调用单独的解密进程再返回CSV文件，所有的缓存文件在使用过后第一时间被清理；

导出操作记录：用户所进行的每一步操作都会被纪录，记录格式是：“谁”“在什么时候”“进行了什么操作”，“结果是”，这个将用于后期分析；

清空数据库：链式表的逐个删除；

帮助和关于：设置了问卷；

换肤：采用的原理是QSS，类似于CSS的格式进行完全的界面重写，这个花费了将近整整一周的时间，虽然不是那么重要。

### 相关知识

* C++语言
* 数据结构中的链式表
* Qt开发
* QSS
* DES加密

## 2 需求分析

### 2.1 课题调研

超市商品信息管理系统是一个由人和计算机等组成的能够提供信息以支持一个组织机构内部的作业、管理、分析和决策职能的系统。系统利用计算机的硬件和软件，手工规程、分析、计划、控制和决策用的模型，以及数据库对信息进行收集、传输、加工、保存、维护和使用。因此该系统也是一个信息处理系统。现在超市商品信息管理系统作为技术、工作自动化及高水平管理的方法和模式,正在得到越来越广泛的应用。早期的信息管理系统通常是以各单位为主体,独自进行开发和应用。从技术上看,大体是以局域网或客户/服务器模式组成应用系统平台,在操作系统和数据库管理系统的平台上开发应用软件系统。但是,由于技术的发展和进步以及应用、管理和数据共享的需要等诸多原因,把分散开发且分布在各个地域的独立的管理信息系统互相连接起来,实现系统硬件、系统软件、应用软件及数据的共享,已经成为越来越迫切的任务。

### 2.2 用户需求分析

超市形态的高速发展，其经营管理也变得愈加复杂，日常所需要处理的数据量也逐渐增大，商业运转的中间环节也越来越多，原始的人工管理已无法应对这复杂的市场。为此，在选题过程中，我们小组选择了超市商品信息管理系统设计题目，依靠现代化的计算机信息处理技术来管理超市，从而节省了大量的人力、物力，改善了员工的工作条件，减轻了劳动强度，并且能够快速反映出商品的进、销、存等状况和各种反馈信息分析，使管理人员快速对市场的变化做出相应的决策，加快超市经营管理效率。

通过对市场需求的详细调查与分析，根据消费者的反馈和超市日常运作机制，我们小组认为对超市商品实行以下功能，能够对超市商品进行有效的管理，大大提高了超市管理的效率，减轻工作人员的负担。

1、完善的进销存功能：从开进货单到库存查询盘点、商品销售，完整的进销存功能。

2、查询销售、浏览销售功能：在前台系统中的销售功能支持按商品名称、条码、商品货号、类别货号来查询商品进行销售；在后台管理系统中的商品查询中支持按类别浏览销售、按条码、货号、商品名称查询销售。

数据库中需要存储的数据：商品名称、条码、商品货号、货号类别、商品的数量、供应商的信息、职工的信息、超市的信息。

(1)该管理软件将对库存商品进行资料管理和统计，对产品的进货单和产品的销售单进行录入管理和统计；

(2)该管理软件将根据用户的需要对进货、销售、库存情况进行查询，并能打印详细信息；

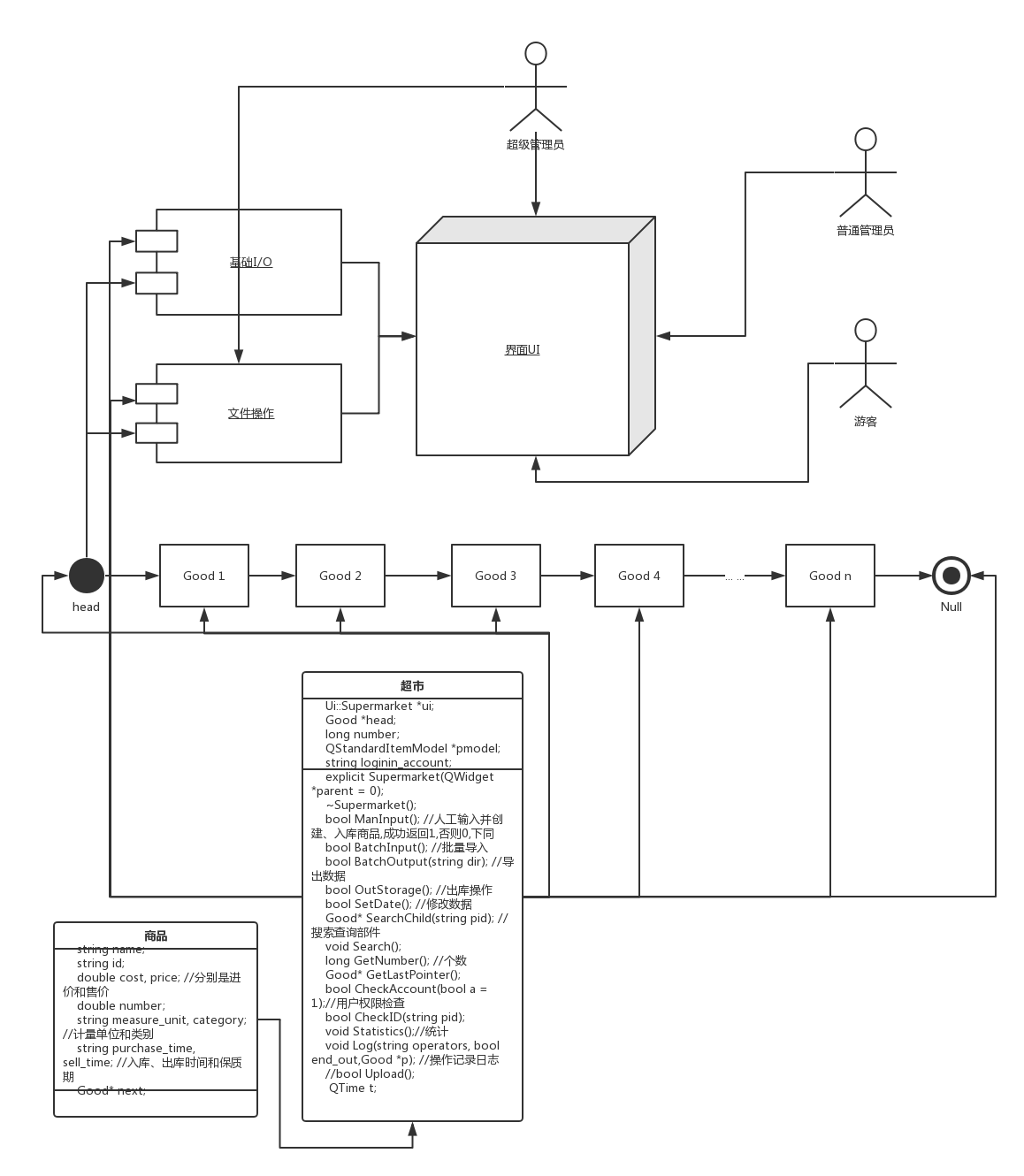
(3)该管理软件将为用户提供账号管理工具，以便用户对账号进行管理，具体为添加用户、删除用户、修改用户密码；

(4)该管理软件提供企业人员管理，能够对企业员工的分类管理，新员工的加入以及员工删除，修改进本信息等功能。

这个商品信息管理系统可以随时查询超市商品的销售情况，管理员可以通过查询了解超市商品的销售情况，进而决定以后是否继续订购某种商品。消费者可以根据自己所查到的商品信息，购买自己所想要的商品。如果超市中某种商品的数量低于系统所设定商品种数量，系统会自动提醒要订货了，当订货到一定数量的时候，系统又会提醒订货的数量够了。这样超市就可以按照各种商品的销售量得到商品的重要信息，进而采取有效的调控手段，为超市的运行、赢利创造更加有利的条件。

## 3 方案设计

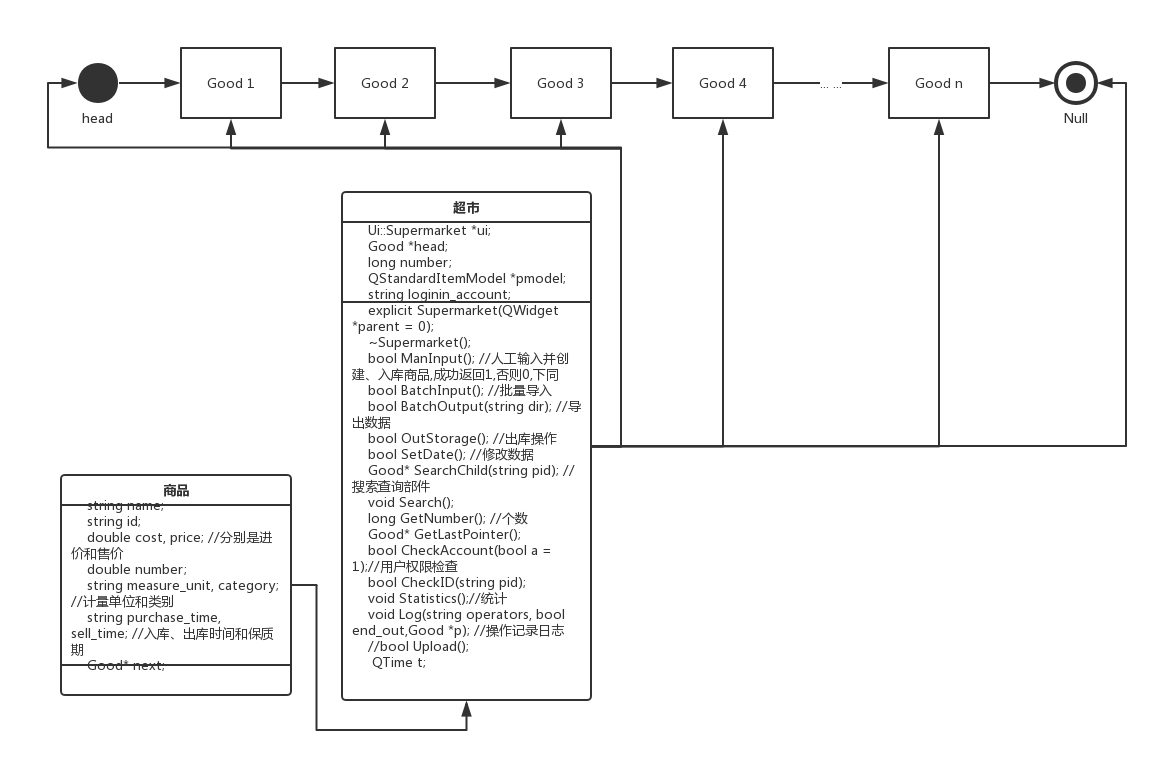
### 3.1 总体功能设计



原png图像文件（<超市库存管理系统-总体功能设计.png>）

原svg矢量图文件（<超市库存管理系统-总体功能设计.svg>）

### 3.2 数据结构设计



原png图像文件（<超市库存管理系统-数据结构设计.png>）

原svg矢量图文件（<超市库存管理系统-数据结构设计.svg>）

### 3.3 函数原型设计

namespace Ui {

class Supermarket;

}//加载UI

class Supermarket : public QMainWindow

{

Q\_OBJECT

public:

explicit Supermarket(QWidget \*parent = 0);

~***Supermarket***();//析构函数，清空数据库销毁界面

bool **ManInput**(); //人工输入并创建、入库商品,成功返回1,否则0,下同

bool **BatchInput**(); //批量导入

bool **BatchOutput**(string dir); //导出数据为CSV

bool **OutStorage**(); //出库操作

bool **SetDate**(); //修改数据

Good\* **SearchChild**(string pid); //搜索查询部件，元搜索

void **Search**();//搜索查询

long **GetNumber**(); //个数

Good\* **GetLastPointer**();//得到当前链表的最后一个节点的地址

bool **CheckAccount**(bool a = 1);//用户权限检查

bool **CheckID**(string pid); //检查当前的商品ID是否合法

void **Statistics**();//统计中心

void **Log**(string operators, bool end\_out,Good \*p); //操作记录日志

//bool Upload();//上传备份到服务器，暂未实现

QTime t; //公共时间显示和计时器

private:

Ui::Supermarket \*ui;

Good \*head;

long number;

QStandardItemModel \*pmodel;

string loginin\_account;

private slots:

QStandardItemModel\* **initForm**();//根据当前条件初始化界面UI、数据和时间

void **initAccount**();//初始化账户设置

void **initUnAccount**();//反初始化账户

void **on\_pushButton\_clicked**();//账户登录

void **on\_pushButton\_3\_clicked**();//入库

void **on\_pushButton\_2\_clicked**();//出库

void **on\_pushButton\_6\_clicked**();//搜索

void **on\_pushButton\_7\_clicked**();//修改

void **on\_action\_O\_triggered**();//导入文件

void **on\_actionCSV\_C\_triggered**();//导出CSV

void **on\_action\_H\_triggered**();//帮助

void **on\_action\_A\_triggered**();//关于

void **on\_action\_B\_triggered**();//反馈

void **on\_pushButton\_8\_clicked**();//置空入库

void **on\_pushButton\_9\_clicked**();//置空搜索

void **on\_pushButton\_10\_clicked**();//置空修改

void **on\_pushButton\_4\_clicked**();//反注册

void **on\_actiond\_2\_triggered**();//清空数据库

void **on\_actiond\_triggered**();//读取文件进入链式表

void **on\_actiondda\_triggered**();//读取文件并判断格式进行解密处理

public slots:

void **timerUpdate**(void); //获取当前时间并转化格式

void **timerAdd**();//计时器

};

//void UpLoadForm(QString Path,QMap<QString,QString> params,QString fileFormName,QFile \*uploadFile,QString newFileName);

### 3.4 主算法设计

//创建和入库主要操作

bool neuSupermarket::ManInput()

{

Good \*p,\*q;

p = (Good\*)new(Good);

cin>>p->name>>p->id>>p->cost>>p->price>>p->number>>p->measure\_unit>>p->category;

p->purchase\_time = time(NULL);

p->next = NULL;

number++;

if(head == NULL)

head = p;

else

{

q = GetLastPointer();

q->next = p;

}

return true;

}

//搜索主要操作

Good\* neuSupermarket::Search(long pid)

{

Good\* p = head, \*q;

if(p == NULL)

{

cerr<<"Nothing to be found"<<endl;

return NULL;

}

else

{

while(p->next!=NULL)

{

if(p->id == pid)

{

q = p;

break;

}

else p = p->next;

}

return q;

}

}

//构造析构和得到最后商品的指针地址

neuSupermarket::neuSupermarket()

{

number = 0;

head = NULL;

}

neuSupermarket::~neuSupermarket()

{

Good \*p;

for(p=head; p->next!=NULL; p=p->next)

{

delete p;

}

delete p;

}

Good\* neuSupermarket::GetLastPointer()

{

Good\* p;

if(head == NULL) return head;

else

{

for(p=head; p->next!=NULL; p=p->next);

return p;

}

}

### 3.5 用户界面设计

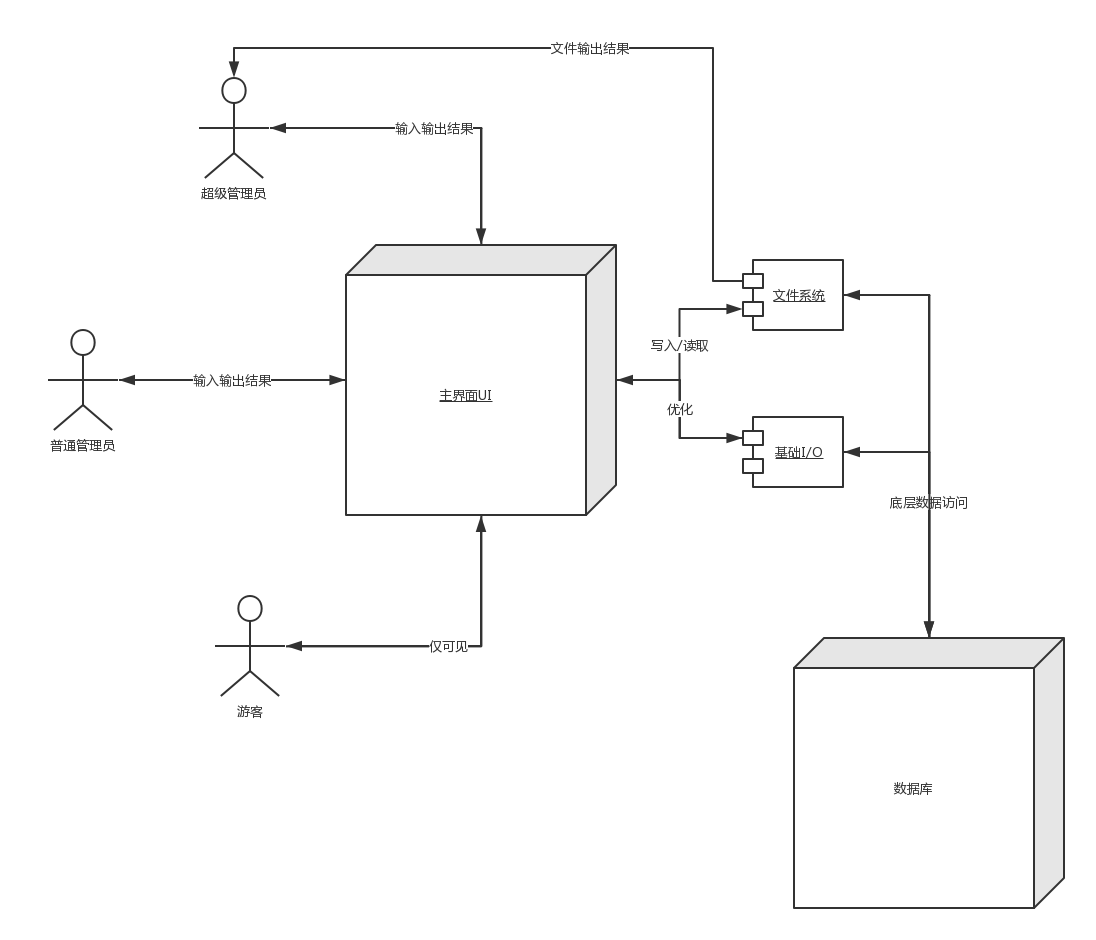
主要界面如下，经QSS重写。详细操作及界面见[屏幕演示录像文件](#_B-4屏幕演示录像文件（可选）_1)



登录批量导入后：



### 3.6 输入输出设计



原png图像文件（<超市库存管理系统-输入输出设计.png>）

原svg矢量图文件（<超市库存管理系统-输入输出设计.svg>）

另外，关于DES加密：

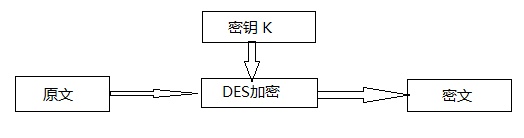
DES是一种对称加密算法，加密和解密使用相同密钥的算法。

DES算法的入口参数有三个：Key、Data、Mode。

Key为8个字节共64位，是DES算法的工作密钥；

Data也为8个字节64位，是要被加密或被解密的数据；

Mode为DES的工作方式，有两种：加密或解密。



DES算法把64位的明文输入块变为64位的密文输出块，它所使用的密钥也是64位，整个算法的主流程图如下：   
其功能是把输入的64位数据块按位重新组合，并把输出分为L0、R0两部分，每部分各长32位，其置换规则见下表：   
58,50,12,34,26,18,10,2,60,52,44,36,28,20,12,4,   
　　62,54,46,38,30,22,14,6,64,56,48,40,32,24,16,8,   
　　57,49,41,33,25,17, 9,1,59,51,43,35,27,19,11,3,   
　　61,53,45,37,29,21,13,5,63,55,47,39,31,23,15,7,   
　　即将输入的第58位换到第一位，第50位换到第2位，...，依此类推，最后一位是原来的第7位。L0、R0则是换位输出后的两部分，L0是输出的左32位，R0 是右32位，例：设置换前的输入值为D1D2D3......D64，则经过初始置换后的结果为：L0=D58D50...D8；R0=D57D49...D7。   
　　经过16次迭代运算后。得到L16、R16，将此作为输入，进行逆置换，即得到密文输出。逆置换正好是初始置的逆运算，例如，第1位经过初始置换后，处于第40位，而通过逆置换，又将第40位换回到第1位，其逆置换规则如下表所示：   
　　40,8,48,16,56,24,64,32,39,7,47,15,55,23,63,31,   
　　38,6,46,14,54,22,62,30,37,5,45,13,53,21,61,29,   
　　36,4,44,12,52,20,60,28,35,3,43,11,51,19,59,27,   
　　34,2,42,10,50,18,58 26,33,1,41, 9,49,17,57,25,   
放大换位表   
　　32, 1, 2, 3, 4, 5, 4, 5, 6, 7, 8, 9, 8, 9, 10,11,   
　　12,13,12,13,14,15,16,17,16,17,18,19,20,21,20,21,   
　　22,23,24,25,24,25,26,27,28,29,28,29,30,31,32, 1,   
单纯换位表   
　　16,7,20,21,29,12,28,17, 1,15,23,26, 5,18,31,10,   
　　2,8,24,14,32,27, 3, 9,19,13,30, 6,22,11, 4,25,   
　　在f(Ri,Ki)算法描述图中，S1,S2...S8为选择函数，其功能是把6bit数据变为4bit数据。下面给出选择函数Si(i=1,2......的功能表：   
选择函数Si   
S1:   
　　14,4,13,1,2,15,11,8,3,10,6,12,5,9,0,7,   
　　0,15,7,4,14,2,13,1,10,6,12,11,9,5,3,8,   
　　4,1,14,8,13,6,2,11,15,12,9,7,3,10,5,0,   
　　15,12,8,2,4,9,1,7,5,11,3,14,10,0,6,13,   
S2:   
　　15,1,8,14,6,11,3,4,9,7,2,13,12,0,5,10,   
　　3,13,4,7,15,2,8,14,12,0,1,10,6,9,11,5,   
　　0,14,7,11,10,4,13,1,5,8,12,6,9,3,2,15,   
　　13,8,10,1,3,15,4,2,11,6,7,12,0,5,14,9,   
S3:   
　　10,0,9,14,6,3,15,5,1,13,12,7,11,4,2,8,   
　　13,7,0,9,3,4,6,10,2,8,5,14,12,11,15,1,   
　　13,6,4,9,8,15,3,0,11,1,2,12,5,10,14,7,   
　　1,10,13,0,6,9,8,7,4,15,14,3,11,5,2,12,   
S4:   
　　7,13,14,3,0,6,9,10,1,2,8,5,11,12,4,15,   
　　13,8,11,5,6,15,0,3,4,7,2,12,1,10,14,9,   
　　10,6,9,0,12,11,7,13,15,1,3,14,5,2,8,4,   
　　3,15,0,6,10,1,13,8,9,4,5,11,12,7,2,14,   
S5:   
　　2,12,4,1,7,10,11,6,8,5,3,15,13,0,14,9,   
　　14,11,2,12,4,7,13,1,5,0,15,10,3,9,8,6,   
　　4,2,1,11,10,13,7,8,15,9,12,5,6,3,0,14,   
　　11,8,12,7,1,14,2,13,6,15,0,9,10,4,5,3,   
S6:   
　　12,1,10,15,9,2,6,8,0,13,3,4,14,7,5,11,   
　　10,15,4,2,7,12,9,5,6,1,13,14,0,11,3,8,   
　　9,14,15,5,2,8,12,3,7,0,4,10,1,13,11,6,   
　　4,3,2,12,9,5,15,10,11,14,1,7,6,0,8,13,   
S7:   
　　4,11,2,14,15,0,8,13,3,12,9,7,5,10,6,1,   
　　13,0,11,7,4,9,1,10,14,3,5,12,2,15,8,6,   
　　1,4,11,13,12,3,7,14,10,15,6,8,0,5,9,2,   
　　6,11,13,8,1,4,10,7,9,5,0,15,14,2,3,12,   
S8:   
　　13,2,8,4,6,15,11,1,10,9,3,14,5,0,12,7,   
　　1,15,13,8,10,3,7,4,12,5,6,11,0,14,9,2,   
　　7,11,4,1,9,12,14,2,0,6,10,13,15,3,5,8,   
　　2,1,14,7,4,10,8,13,15,12,9,0,3,5,6,11,   
在此以S1为例说明其功能，我们可以看到：在S1中，共有4行数据，命名为0，1、2、3行；每行有16列，命名为0、1、2、3，......，14、15列。   
　　现设输入为： D＝D1D2D3D4D5D6   
令：列＝D2D3D4D5   
　　行＝D1D6   
　　然后在S1表中查得对应的数，以4位二进制表示，此即为选择函数S1的输出。下面给出子密钥Ki(48bit)的生成算法   
　　从子密钥Ki的生成算法描述图中我们可以看到：初始Key值为64位，但DES算法规定，其中第8、16、......64位是奇偶校验位，不参与DES运算。故Key 实际可用位数便只有56位。即：经过缩小选择换位表1的变换后，Key 的位数由64 位变成了56位，此56位分为C0、D0两部分，各28位，然后分别进行第1次循环左移，得到C1、D1，将C1（28位）、D1（28位）合并得到56位，再经过缩小选择换位2，从而便得到了密钥K0（48位）。依此类推，便可得到K1、K2、......、K15，不过需要注意的是，16次循环左移对应的左移位数要依据下述规则进行：   
循环左移位数   
1,1,2,2,2,2,2,2,1,2,2,2,2,2,2,1   
　　以上介绍了DES算法的加密过程。DES算法的解密过程是一样的，区别仅仅在于第一次迭代时用子密钥K15，第二次K14、......，最后一次用K0，算法本身并没有任何变化。

## 4 方案实现

### 4.1 开发环境与工具

开发环境：Windows 10 + Qt 5.10 + MinGw 5.3

开发工具：Qt Creator 4.4 + NotePad++ 7.2

### 4.2 程序设计关键技术

* **数据结构中的链式表**
* **QSS**
* **DES加密**

### 4.3 个人设计实现（按组员分工）

#### 4.3.1 戚子强设计实现

主要负责类定义；批量CSV文件、DES加密文件导入；批量CSV、DES加密文件导出；进行操作记录；操作记录导出；换肤；界面引擎重写；Qt移植的设计实现。

在类定义中，在充分参考了已有的相似的超市仓库管理系统之后，确定了商品ID、名称、进价、售价、数量、单位、分类、进货时间、出货时间这九个最为基础的元素作为一个商品节点的组成因素，选取了商品的创建、入库、出库、元查询、查询、修改、数量统计、批量输入输出等几个操作作为基础操作并进行了合适的定义声明。

在批量CSV文件和DES加密文件处理中，采取了文件缓冲区的方式，即程序直接读取和输出CSV文件作为缓冲，DES加密则是独立于主程序之外的独立进程的文件加密程序，读取CSV文件并加密为QZQ文件后清理缓冲区，读取操作反之。对于DES加密，在3.6中有详细介绍，在此不再赘述。

在界面引擎方面，采用了QSS把几乎所有可见界面进行了完全重写，并封装成小型QUI库，类似于重写QMainwindow和QMessage。

#### 4.3.2吕振峰设计实现

主要负责创建商品；商品入库；商品出库的设计实现。

#### 4.3.3张卓设计实现

主要负责商品查询，展示符合查询ID的名称、进价、售价、数量、单位、分类和操作记录；商品修改：对符合输入的商品ID、名称或二者进行各种信息的修改；清空数据库：将链表中数据全部删除，清空所有数据；账户权限：识别登录的身份，对登录者分配权限，超级管理员拥有所有包括批量和单个操作权限，普通员工只有单个商品的操作权限；账户操作：展示登入的账户的身份，登录时长与当前时间；统计中心：显示商店员工总数，统计库存商品总数，目前总共支出与总收入；帮助：给出链接记事本的帮助信息；反馈：链接到调查问卷的网站，统计用户的反馈信息；关于：链接到。

## 5 测试与调试

### 5.1 个人测试（按组员分工）

#### 5.1.1 戚子强测试

主要负责创建商品、商品入库、商品出库、商品查询、商品修改、清空数据库、账户权限、账户操作、统计中心的测试。

在商品入库测试中发现，如果输入一个已有ID但是进价不同的BUG，我们进行了修改，即当输入价格不同的时候，将已有的商品价值和所输入的商品价值相加再除以总数量，得到平均价格，作为当前商品价格。另外，之前的版本中，账户操作做的很复杂，后来在移植Qt的时候我们考虑到，这可能比较难实现并且不是主要功能，所以在之后的版本中实际上删减了很多。但我们相信如果在实际中有这方面的需要并且有更多的时间允许的话，我们会做得更好更全一些。

#### 5.1.2吕振峰测试

主要负责界面方面和批量读取写入的测试。在写入的时候，发现一个BUG是，在copyFile()函数中，如果在某一个文件夹已经存在同名文件，则不会进行任何复制操作，后来修改了一下，覆盖了原文件并做了“文件缓存区”来降低加密文件内容和链表数据库直接交互所产生的风险。

#### 5.1.3张卓测试

在商品入库测试时发现，如果输入相同ID和名称的商品可以输入不同的商品种类，我们进行了修改，使其不能具有不同的种类；如果输入相同ID和名称的商品不能归结到一个商品中，我们进行了修改，即相同ID和名称为一种商品。

### 5.2 组装与系统测试

**组装：**由三位成员完成各自负责的C++原生代码设计后，由戚子强将原生代码合并并移植到Qt平台，期间主要是替换所有I/O操作和解决一些逻辑兼容问题。

**系统测试：**组装完毕之后，在Qt平台编译DEBUG版本并分发到各位成员手中进行各种可能的数据测试，一旦出现不符合实际情况的部分立即进行分析并进行记录，找到目标问题所在的函数位置；如此进行反复测试。将所在函数位置修改完成后在Qt中进行局部替换，经过多次测试后生成RELEASE版本并进行程序打包成成安装包，请各位同学安装并进行了多次测试。期间出现的问题也会及时记录并修改。

### 5.3 系统运行

最终版本是安装包，内置了Qt各个组件以保证在各种平台正常运行而不会出现缺少组件的情况。用户可以像平常安装软件一样进行操作。

## 6 课题总结

### 6.1 课题评价

在本次整个的课题中，我们每个人在巩固了数据结构知识的同时，都学习到了很多新知识，尤其是与现代互联网、软件公司的实际开发中的工具和方式方法等，比如Qt跨平台开发、QSS界面重写、DES加密算法以及各种文件处理和软件测试工作，当然还团队协作这一块儿。

本次课题中的优点之处：一是进行了大量的课外知识补充，比如打破了“小黑框”界面，通过QSS重写界面引擎，更具有人机交互的友好界面和更加高效的操作方式，再比如DES加密，可以有效地增加数据的安全性；二是对已有的知识的掌握更加熟练了，这使得我们在第一天就完成了任务一的所有C++原生代码，为之后的各种移植和测试提供了大量时间和精力；三是各位成员之间通过有效高效的事先“约定”，遵循软件开发流程和代码规范之后，虽然牺牲了一点点时间去学习，但这是事半功倍的，越往后体现出来的便捷性就越大，尤其是在DEBUG阶段出现各种逻辑错误的时候，我们可以很快的找到问题所在位置并及时迅速的进行修改、移植和合并工作。

本次课题的不足之处：一是时间仓促并没有考虑到更多的实际因素比如打折、促销等等活动；二是答辩阶段没有进行针对性的提前准备导致了老师问出一些问题之后觉得“诶，这种问题也是问题吗？”或者“这种问题也需要解释吗？”，这实际上是与实际的应用市场环境有了一些脱节，也暴露出了把精力只集中于技术方面而忽略了语言表达上的问题。

### 6.2 团队协作

在团队协作方面，我们在课题开始之前便进行了一些“约定”，正是这些让之后的开发沟通变得高效和准确。现摘抄原文如下：

数据结构课设实现方案（草案）

1. 思路及分工

（一）

对于超市商品库存，分两个文件（Supermarket.h和Supermarket.cpp）写代码，Supermarket.h中定义商品类并给出需要注意的各项规范和函数列表，Supermarket.cpp负责将Supermarket.h中所涉及函数逐个实现。

1. 用顺序表存储。好处：一是简单，二是方便于后面的批量文件导入导出。坏处：占用空间大一些，不过相比于界面引擎可以忽略不计了。每个商品是一个结构体，包含商品名称、ID、进价、售价、库存量、销量、类别、进货时间、出货时间、保质期，另外要有日志记录所有操作；
2. 人工输入就按正常做。文件导入要自定义一种文件格式，有单独入口进行批量操作。随机可以设置一个专门的测试按钮生成随机数据。这三种IO方式各自单独实现；
3. 即对单链表的数据的成员的修改操作；
4. 查询即写一个搜索函数，统计需要有商品种数、每件商品的历史记录（其他的呢？）
5. 界面重写；
6. 高级特性：界面引擎、云备份、日志和数据加密处理；

（二）

对于学生成绩条形图，同分两个文件（Statistics.h和Statistics.cpp），Statistics.h建立二叉排序树，实现核心统计功能。Statistics.cpp只负责输入输出统计数据即可。

1. 众数、中位数、平均数、数据分组、方差；（还有其他的吗）
2. 当数据量足够大的时候应该用什么方法提高效率保证稳定性；
3. 绘制图形；

1. 代码规范

1. 包含头文件

• 先系统头文件，后用户头文件。

• 系统头文件应用：#include <xxx.h>

• 自定义同文件应用：#include "xxx.h"

• 只引用需要的头文件。

1. h和cpp文件

• 定义写在.h里，函数实现在.cpp里

• 头文件命名为\*.h，内联文件命名为.inl；C++文件命名为\*.cpp

• 文件名和函数名、类名、常量用大小写混合，例如DiyMainView.cpp。变量用小写加下划线，如goods\_price。不要用无意义的名称。

• 头文件使用#ifdef控制块。

• 头文件#endif应采用行尾注释。



• 头文件，首先是包含代码块，其次是宏定义代码块，然后是全局变量，全局常量，类型定义，类定义，内联部分。

• CPP文件，包含指令，宏定义，全局变量，函数定义。

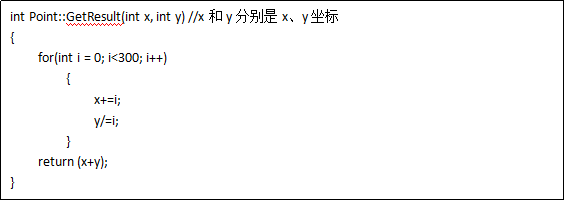
3. 文件结构

• 文件一定要包含文件头注释和内容。

• 函数与函数、类与类之间用2个空行。

• 文件头、控制块，#include部分、宏定义部分、class部分、全局常量部分、全局变量部分、函数和函数之间，用两个空行。

• 函数按cpp规范写：



### 6.3 个人设计小结（按组员分工）

#### 6.3.1戚子强小结

在本次的程序设计中，从我所负责的设计中看，需要学习很多课外的东西，更需要有坚实的数据结构基础，这样可以更好的理解新的知识并快速的运用到实际中，大大降低了学习成本。

另外，也深切体会到了中期测试和后期运维的重要性。

在中期的测试环节，我们通过输入大量的非正常数据进行了测试，之后把大多数的BUG进行了及时的修复，但是截止交稿时期，还是有一些没有修复的BUG和一些隐藏的BUG，这些BUG虽然在正常使用中不会出现，但是一旦出现，有一些BUG将会是对程序非常致命的。之前的时候常常是，写完程序能够正常运行就OK，但是忽略了很多在非常情况下所出现的BUG，这一次在时间允许的情况下，我们进行了尽可能多的人员和测试方式，发现了很多的问题。这其中也有一些甚至是平台自身所带的缺陷，例如在DOS窗口下，在输入数字的时候输入非法中文字符，程序会瞬间崩溃，在读取文件也曾出现致命错误。

#### 6.3.2吕振峰小结

本次试验中，主要体会到了程序完成之后测试的重要性。更多的程序错误和失误往往是逻辑错误而不是低级的编译语法方面；而且当程序越来越大的时候，逻辑错误将更难以找出根源，所以这也就要求我们每个人在写程序的时候要严格遵循规范文档，只有这样才能够将测试的压力降低一些。

#### 6.3.3张卓小结

通过这次课程设计，程序设计部分使我懂得了写代码之前的规划的重要性，这样具有了逻辑才能更快更准确的写出程序；在遇到不懂的问题时，要不断学习，请教他人，遇到问题时要耐心寻找并解决问题，经过多次调试，才能使程序变的更加严谨，只有不断的调试，积累经验才能变成一个编程高手，要捋清思路，模拟一下运行过程，错误就会被发现，在测试过程中，要细致，每个功能每个细节都要考虑到，以免出现难以发现的问题。

在以后的学习中，除了注意课本的学习，也要课下多进行代码编程，多动手实践才能掌握地更准确。

## 7 附录

### A 课题任务分工

#### A-1 课题程序设计分工

**课题程序设计分工**

|  |  |  |  |
| --- | --- | --- | --- |
| 学号 | 姓名 | 程序设计函数原型、类 | 功能说明 |
| 20164625 | 戚子强 | struct Good  {  string name;  string id;  double cost, price; //分别是进价和售价  double number;  string measure\_unit, category; //计量单位和类别  string purchase\_time, sell\_time; //入库、出库时间和保质期  Good\* next;  };  class Supermarket : public QMainWindow  {  Q\_OBJECT  public:  explicit Supermarket(QWidget \*parent = 0);  ~***Supermarket***();  bool **ManInput**(); //人工输入并创建、入库商品,成功返回1,否则0,下同  bool **BatchInput**(); //批量导入  bool **BatchOutput**(string dir); //导出数据  bool **OutStorage**(); //出库操作  bool **SetDate**(); //修改数据  Good\* **SearchChild**(string pid); //搜索查询部件  void **Search**();  long **GetNumber**(); //个数  Good\* **GetLastPointer**();  bool **CheckAccount**(bool a = 1);//用户权限检查  bool **CheckID**(string pid);  void **Statistics**();//统计  void **Log**(string operators, bool end\_out,Good \*p); //操作记录日志  //bool Upload();  QTime t;  private:  Ui::Supermarket \*ui;  Good \*head;  long number;  QStandardItemModel \*pmodel;  string loginin\_account;  private slots:  QStandardItemModel\* **initForm**();  void **initAccount**();  void **initUnAccount**();  public slots:  void **timerUpdate**(void);  void **timerAdd**();  };  //void UpLoadForm(QString Path,QMap<QString,QString> params,QString fileFormName,QFile \*uploadFile,QString newFileName);  #endif // SUPERMARKET\_H | 实现基本的节点和商品类，定义了各个函数的返回值和形参列表；完成了批量CSV文件、DES加密文件导入；批量CSV、DES加密文件导出；进行操作记录；操作记录导出；换肤；界面引擎重写。 |
|  |  | //入库和创建  bool Supermarket::**ManInput**();  //出库  bool Supermarket::**OutStorage**() | 负责完成实现了创建商品；商品入库；商品出库的设计。其中入库操作进行了大量的优化处理和错误应急处理。 |
|  |  | //初始化账户  void Supermarket::**initAccount**();  //账户登录  void Supermarket::**on\_pushButton\_clicked**();  //账户权限检查(三级:超管,管理,游客;a =0 为超管)  bool Supermarket::**CheckAccount**(bool a);  //清空数据库  void Supermarket::**on\_actiond\_2\_triggered**();  bool Supermarket::**CheckID**(string pid);  //元搜索  Good\* Supermarket::**SearchChild**(string pid)  //搜索  void Supermarket:: **Search**();  //设置修改参数  bool Supermarket::**SetDate**();  //统计  void Supermarket::**Statistics**();  //置空操作  void Supermarket::**on\_pushButton\_8\_clicked**(); | 负责完成了商品查询；商品修改；清空数据库；账户权限；账户操作；统计中心；帮助；反馈；关于的设计实现。其中元搜索作为其他诸多函数的通用子函数使用，在设置参数方面进行了大量的优化使得所期望的效果得以实现。 |

#### A-2 课题报告分工

**课题报告分工**

|  |  |  |
| --- | --- | --- |
| 章节 | 内容 | 完成人 |
| 1 课题概述 | 1.1 课题任务  1.2 课题原理  1.3 相关知识 | 戚子强 |
| 2 需求分析 | 2.1 课题调研  2.2 用户需求分析 | 张卓 |
| 3 方案设计 | 3.1 总体功能设计  3.2 数据结构设计  3.3 函数原型设计  3.4 输入输出设计  3.5 主算法设计  3.6 用户界面设计 | 戚子强 |
| 4 方案实现 | 4.1 开发环境与工具  4.2 程序设计关键技术  4.3 个人设计实现（按组员分工）  4.3.1  4.3.2  4.3.3 | 戚子强 |
| 5 测试与调试 | 5.1 个人测试（按组员分工）  5.1.1  5.1.2  5.1.3  5.2 组装与系统测试  5.3 系统运行 | 张卓 |
| 6 课题总结 | 6.1 课题评价  6.2 团队协作  6.3 下一步工作  6.4 个人设计心得（按组员分工）  6.4.1  6.4.2  6.4.3 | 吕振峰 |

### B 课题设计文档（光盘）

#### B-1课程设计报告（电子版）

即本文。

#### B-2源程序代码（\*.H，\*.CPP）

head.h

quiwidget.h

supermarket.h

main.cpp

quiwidget.cpp

supermarket.cpp

des.h

des.cpp

main.cpp

#### B-3工程与可执行文件

工程请见目录下Neusupermarket文件夹。

#### B-4屏幕演示录像文件（可选）

<Neusupermarket.mp4>

### C 使用手册（可选）

#### C.1 运行环境说明

操作系统: Win XP+ X86 & X64

配置： 内存 2GB+

硬盘 40GB+

#### C.2 操作说明

首先登录到超级管理员或者普通管理员：

入库时逐个输入要输入的信息：

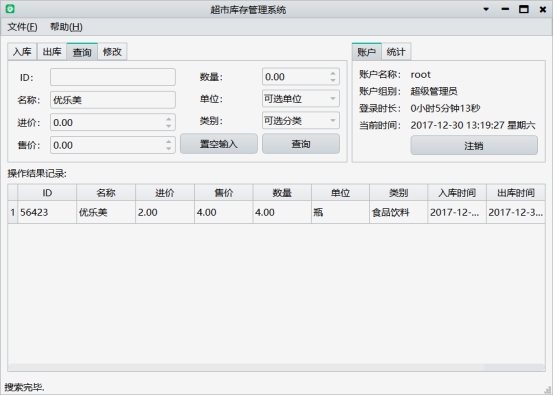


可以选择进价售价单位数量分类等信息。

出库时输入出库的ID和数量



查询时可以选择查询的ID，名称



修改时必须输入ID

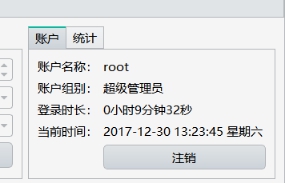




统计页面中有商品种数，员工总数，总支出和价值的统计



账户界面显示登录时长，当前时间，也可以选择注销



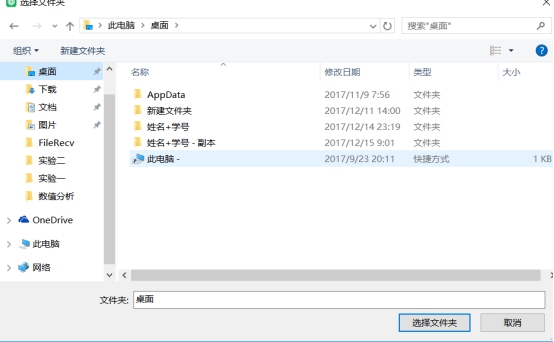
工具栏中选择文件，可以进行导入、导出、清空数据库等批量操作。

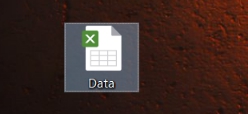


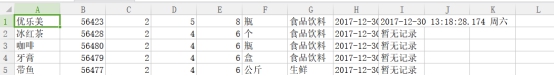
导出数据使用Excel表格进行操作

C:\Users\Andy\AppData\Local\Temp\ksohtml\wpsEF95.tmp.jpg

导出时选择导出地址







导入时选择要导入的表格文件

导入后显示如下



同时也可以导出加密文件

C:\Users\Andy\AppData\Local\Temp\ksohtml\wpsEFAA.tmp.jpg

帮助栏有反馈，关于等操作



可以根据用户需要修改界面颜色



也可以最小化最大化。