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；
4. 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。

#ifndef POINT\_H

#define POINT\_H

...

#endif //Point.h

　• 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：

int Point::GetResult(int x, int y) //x和y分别是x、y坐标

{

for(int i = 0; i<300; i++)

{

x+=i;

y/=i;

}

return (x+y);

}