

面向对象程序设计（C++）

大作业报告

袁犇 PB13209203

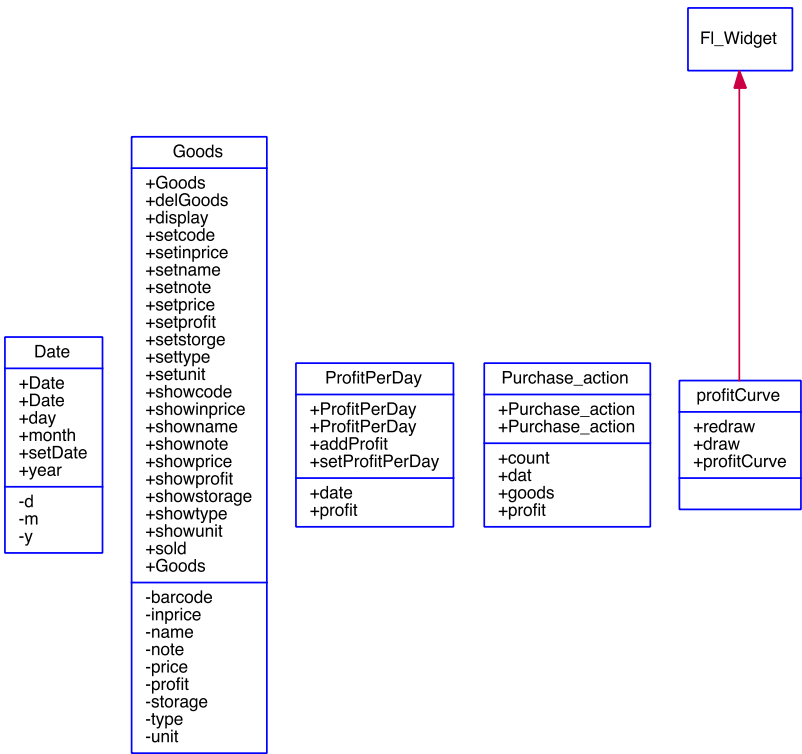
超市收银系统 Supermarket POS system

本次大作业是利用 C++语言，基于 FLTK 图形库来设计超市收银系统。

首先介绍程序的设计思想：

这套程序完全是基于题目所给的要求逐条逐步设计的。我把程序实现大致分为数据层面和图形层面。

数据层面：



程序使用的 Class 一览表

1.“基本信息包括商品名称、商品类别、条形码号、单价、计量单位、进价、库存、备注等项。要求能够新增、删除、修改、检索商品信息，能够检测非法条形码号并防止录入重复信息，允许以文件形式批量导入、导出商品信息。”

题目中描述的基本商品对应的是上图中的 Goods 类，类中成员以及成员函数功能基本上和名字一一对应。

可以看出 Goods 类其实是对于一种商品的描述，而超市的商品实际上是 Goods 类的一个集合，因此，对于超市中所所有商品，这里定义了一个 vector<Goods>来实现，并且该 Goods 向量名字也很形象：Warehouse（仓库）。

定义了该向量之后，搜索商品则变得十分容易，只需通过迭代器遍历访问 Warehouse 即可，而删除商品则是通过将商品信息置为非法值来达到的。

至于检测非法条形码号并防止录入重复信息，这里首先要说明的是，出于编程工作量以及实际商场运作的考虑，**没有**提供手动输入商品信息的接口。检查条形码的工作实际放到了从文件导入商品信息的过程中。

```
if (barcode < 9999999)
{
    cout<< "Wrong Barcode in file"<<barcode<<endl;
    barcode = -1;
}
for (it = WareHouse.begin(); it != WareHouse.end(); it++)
{
    if (it->showcode() == barcode)
    {
        cout<< "Barcode Duplicated in the input file!"<<endl;
        break;
    }
}
```

可见上述代码分别进行了条形码检测（位数是否符合）以及是否重复的检查，该套程序默认有效条形码是八位整数（eg.10239523）。

至于从文件输入输出，这在之前的上机题目中已经有类似的练习，直接利用文件流便可以达成。

2.生成一次购买物品清单，包括消费日期、物品名称、单价、数量、总价等。收银找银功能，根据收入金额计算找银金额。

对于物品清单，仍旧是文件操作，生成的收据清单样例如下



Receipt.txt

Welcome to the Supermarket!

=====

Date:(mm/dd/yyyy) 5/7/2016

=====

Name	Price	Count
LaTiao	10.00	2

=====

Total: 20

Recieved:1000

Change:980

其中涉及的类有 Date 类，专门用来管理和日期相关的操作。同时，针对购买行为也专门涉及了 Purchase_action 类，这样可以将商品和日期之间关联起来。方便后面进行统计操作。

3.根据选择的时间跨度，生成营业报表。包括总营业额、利润曲线、各类商品的收益比重等统计功能

关于搜寻给定日期内的消费记录（Purchase_action），由于在进行消费的时候创建了 Purchase_action 的向量，因此也是在该向量（History）中进行迭代器遍历，寻找对应日期以及对

应商品的利润信息。此处需要注意的是，按商品归类的利润存储在 Goods 类的成员 profit 中，而按日期归类的利润则存贮在 ProfitPerDay 类中。

图形层面：

仔细阅读 main 函数会发现，该函数中主题部分都是定义 FLTK 库中的对象。此处主要说明程序中用到的一些对象：

Fl_Input

这个类是 FLTK 库中用来管理图形界面输入的。其中用到的具体类有：

Fl_Int_Input

用于定义仅仅能输入整数的输入框（0-9）。

Fl_Float_Input

用于定义仅仅能输入浮点数的输入框（0-9 & .）。

创建这些对象的时能够生成图形界面的文本输入框，而 FLTK 的事件触发（按下按钮，敲击回车，输入框数值改变等）是通过调用对象的 when 函数（不仅仅对于 Fl_Input 类，对所有的可互动类均是如此）。而触发之后的操作则是通过配置该对象的 callback 函数来实现。举例而言，如果要读入某个输入框的输入值，那么需要针对该输入框（对象）写一个读入（变量赋值）函数，并且将函数作为形参传递给 callback 函数。这些函数统一都放在了 FLTK_functions.h 文件中。

Fl_Output

如其名所示，是用于管理 FLTK 输出的类。本程序中利用该类来生成输出文本框。改变其输出值只用调用其成员函数 value 即可。

Fl_Button

如其名所示，是用于管理 FLTK 按钮的类，使用按钮的原因是显式地告诉用户如何进行操作，直接按回车等方法总是容易让人感觉不直观，也不容易知道操作的是哪一部分。因此，按钮的作用是将输入和输出进行关联。

Fl_Box

这个类在本程序中的作用主要是标清楚各个区域，让用户比较直观地找到在哪里搜索，在哪里消费，在哪里查看统计信息。

至于绘图以及输出各个商品所占利润百分比，程序中采用的是 fl_draw 系列的函数。需要注意的是，FLTK 并不支持直接调用这些函数进行绘图，必须将其作为用户自定义类中的成员函数才能使用。这也是程序中 profitCurve 类的设计初衷。

最后附上一张程序运行结果图：

Search Section

Barcode:

Name

Price

Category

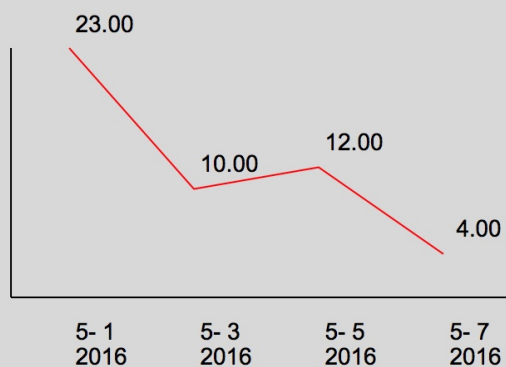
Barcode

Storage

Today(mm/dd/yyyy):

Start(mm/dd/yyyy):

End(mm/dd/yyyy):



Total Profit: 49.00

Trading Section

Barcode: Num:

Name

Price

Category

Barcode

Storage

(after)

Total:

Received:

Change:

Profit Contribution

Sweater: 47.92 %

Chicken: 12.50 %

Milk: 8.33 %

LaTiao: 8.33 %

Calculator: 25.00 %