文章编号: 2096-1472(2019)-08-40-04

DOI:10.19644/j.cnki.issn2096-1472.2019.08.013

# 基于WPF的账单管理程序的设计与实现

## 邹 珺

(苏州农业职业技术学院, 江苏 苏州 215008)

摘 要:早期的账单管理大部分都是人工操作,其缺点是耗费大量的人力和时间。为了让用户高效、便捷地管理账单,账单管理程序能实现简单的个人消费管理,它能够记录和统计个人每月的花销,提供良好的历史信息查询功能。本文主要描述使用WPF技术实现账单管理程序,通过典型的三层架构模式进行开发,为每一个层创建一个类库,为用户界面层创建一个WPF项目,具体由逻辑分层实现程序的功能,使用接口来实现层与层之间的解耦,针对账单管理可以达到良好的效果。

关键词:账单管理程序,架构,WPF中图分类号:TP312 文献标识码:A

# Design and Implementation of the Bill Management Program Based on WPF

ZOU Jun

(Suzhou Agricultural Vocational College, Suzhou 215008, China)

Abstract: Most bill management used to be manually conducted with the shortcoming of consuming a lot of manpower and time. In order to enable users to manage bills efficiently and conveniently, the bill management program can realize simple personal consumption management. It can record and count individual monthly expenses and provide good historical information query function. This paper mainly describes how to use WPF technology to implement bill management program through the typical three-tier architecture model to develop, creating a class library for each tier and a WPF project for the user interface layer, specifically realizing the function of the program by logical hierarchy, using interface to achieve decoupling between tiers, which brings good results for bill management.

Keywords:bill management program;framework;WPF

### 1 引言(Introduction)

随着计算机技术的不断发展,传统的账单管理出现了很多弊端,特别是人工操作,会带来很多误差,甚至很严重的问题。为了避免上述问题,开发了账单管理程序,该程序能提供的功能包括:(1)显示每月的费用记录;(2)存档支出记录;(3)显示最近的消费底线;(4)花费统计图表;(5)特定时间段内的花费统计图表;(6)显示关于每张账单的信息;(7)可以将账单存档记录导出到文件进行备份;(8)可以在多语言间进行切换,实现多语言功能。

本程序使用WPF实现其具体功能,通过软件分层架构模式,解决因软件复杂度的增加而带来的开发和维护问题,将解决方案的组件分隔到不同的层中,自底向上分别是用于访问数据的数据访问层、用于处理业务逻辑的中间业务逻辑层、用于呈现用户界面的表示层,重点是结合分层架构中的逻辑分层来实现<sup>[1]</sup>。

# 2 账单管理程序概述(Overview of the bill management program)

账单管理程序使用三层架构进行开发, 开发过程中, 每

一层会创建一个对应的类库项目,其功能模块分别为账单、图表、存档和杂项。使用TabControl控件来分页显示账单信息,账单页面用一个DataGrid控件显示过去31天的账单记录。用户在底部的文本框中输入账单记录,单击"添加"按钮,输入的账单信息将添加到DataGrid控件中。单击下一页按钮,将会切换到图表页面,该页面使用图表的形式显示账单统计信息,账单管理器主界面如图1所示,账单图表页面如图2所示。



图1 账单管理器主界面

Fig.1 Bill manager home interface

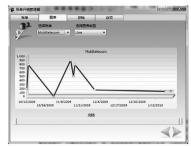


图2账单图表页面

Fig.2 Bill chart page

用户可以在选择账单下拉列表框中选择账单类型,在选择图表类型中选择不同的图表类别显示图表。

在存档页面,提供了过去所有账单历史记录的存档,用 户可以选择不同的账单类别进行支付费用的统计。

在杂项页面,提供了程序的一些配置操作,用户可以导出存档数据到XML文件,程序提供了多语言切换的机制,可以选择不同的语言及时更新用户界面,以及程序的关于信息,存档页面如图3所示,杂项页面如图4所示。

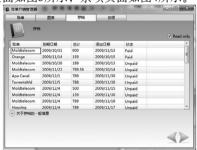


图3 存档页面

Fig.3 Archive page



图4杂项页面

Fig.4 Miscellaneous page

用户可以在语言下拉列表框中选择不同的用户界面语言。导出按钮用于将当前的存档记录导出为XML文件。更改按钮用于更改存档内容,可以选择一个不同的XML文件<sup>[2]</sup>。

# 3 分层结构的实现(Implementation of hierarchical structure)

#### 3.1 实现数据访问层

数据访问层提供了一种弹性的与数据库无关的架构,所以可以使用接口来定义数据访问契约。不同类型的数据库通过实现这个接口,来提供特定数据库类型的功能,而业务逻辑层通过调用这个接口来获取数据,不用知道特定数据库细节信息<sup>[3]</sup>。数据库访问架构如图5所示。

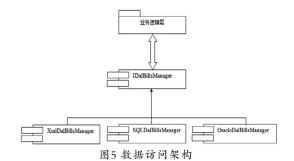


Fig.5 Data access architecture

IDalBillsManager的接口定义了数据访问层契约,可以实现XmlDalBillsManager、SQLDalBillsManager或OracleDalBillsManager来实现对不同数据库的访问,而业务逻辑层只需要关心IDalBillsManager接口,不用与特定的实现细节紧密耦合。

IDalBillsManager实现了账单管理对象的CRUD操作,其 定义代码如下:

// 账单数据管理接口 public interface IDalBillsManager {

// 读取账单

Bills Read(DateTime fromDate, DateTime

toDate);

// 根据指定的账单ID查找账单
Bill ReadById(Guid id);
// 向数据源插入账单
void Insert(Bills bill);
// 从数据源中删除一个账单
int Delete(Guid value);
// 删除特定集合的账单
int Delete(IEnumerable<Guid> guids);
// 从数据库中更新账单
int Update(Guid billId, Bill newBill);
// 为DAL提供者添加设置
object[] Settings
{
 get;
 set;

这些方法和属性需要被实现该接口的类实现, IDalBillsManager接口提供了以下通用的数据访问功能:

- (1)从数据源中读取账单数据,数据源可能是XML、SQL Server、Oracle等数据存储软件。
  - (2)从数据源中删除账单。
  - (3)从数据源中更新账单。
  - (4)插入新的账单到数据源中。

(5)获取或设置数据源选项,例如连接字符串,存档文件 路径等。

定义了这个接口后,业务逻辑层的代码将通过访问接口 来实现对数据的访问,通过面向对象的多态技术,与特定的 实现进行解耦。

#### 3.2 实现业务逻辑层

业务逻辑层实现在BillsBusinessLogicLib项目中,该层实现了与数据访问层相同的架构。将业务逻辑层中将要实现的功能抽象为一个接口,调用方只与接口进行交互,不用关心具体的业务逻辑实现<sup>[4]</sup>。业务逻辑层与数据访问层紧密联系,同时提供给表示层显示的数据,在业务逻辑层中定义了一个接口和一个实现接口的类,以及一个自定义异常类,如图6所示。



图6业务逻辑层类定义

Fig.6 Business logic layer definition

账单管理业务逻辑接口定义了账单管理系统中将要出现的功能逻辑方法或属性,软件分层的目的在于实现层次分明,层与层之间不能过度耦合,通过定义接口,让调用方只与接口交互,不用关心特定的实现细节<sup>[5]</sup>。这样的设计方式也便于维护和扩展,例如软件运行一段时间,现有的业务逻辑不能满足业务处理的需求,那么可以通过定义一个新类,实现业务逻辑接口,让调用端无缝的调用新的业务逻辑层,实现层与层之间的解耦。

业务逻辑层的接口定义代码如下

```
// 账单管理业务逻辑接口
public interface IBillsManager
{
    // 从数据访问层中获取账单集合
    Bills Read(DateTime fromDate, DateTime toDate);
```

```
Bills Read(DateTime fromDate, Date'

// 根据账单Id号返回账单数据
Bill ReadById(Guid id);

// 向数据库中插入账单列表
void Insert(Bills bill);

// 从数据源中删除一个账单
int Delete(Guid bill);

// 使用指定的Guid值集删除账单
int Delete(IEnumerable<Guid> guids);

// 向数据库中更新账单
int Update(Guid billId, Bill newBill);
```

```
// 业务逻辑层设置属性
object[] Settings
{
    get;
    set;
}
```

#### 3.3 实现表示层

表示层提供用户界面来与用户交互,表示层可以考虑为一个Windows Forms程序、一个Silverlight客户端或WPF应用程序。表示层将通过业务逻辑层接口来与业务逻辑层交互。通过WPF实现表示层的设计,为了使用户界面更专业,使用了WPFToolkit中的DataGrid控件实现网格的显示,以及一些第三方的图表控件来增强用户界面的显示效果<sup>[6]</sup>。

主界面使用了WPF的TabControl构建了Tabbed风格的用户界面,在TabControl控件中,包含了四个TabItem,每一个TabItem存放一项分类,一共有四个分类页面,具体功能如下:

(1)账单:在DataGrid 控件中显示最近31天,也就是一个月以来的人账记录。

(2)图表:显示某类型账单的最近支付的图表显示。

(3)存档:在DataGrid控件中显示所有的存档内容。

(4)杂项:提供对语言选择,存档路径设置及软件版本信息内容显示。

在账单管理页面,使用了DataGrid来实现用户界面的显示。因为要实现多语言的切换,所以使用了本地化标识Uid来唯一的标识一个控件。DataGrid控件的生命代码如下:

-的标识一个控件。DataGrid控件的生命代码如下: <WPFToolkit:DataGrid x:Uid="\_dgBills" Style="{StaticResource DataGrid}"

 $\label{lem:containerStyle} ItemContainerStyle="{StaticResource ItemContStyle}" \\ x:Name="\_dgBills" Margin="8,42.96,8,167.08"$ 

IsReadOnly="False" AutoGenerateColumns="False"

CanUserAddRows="False">

<!--定义DataGrid的列-->

<WPFToolkit:DataGrid.Columns>

<!--显示账单名称的文本列-->

<WPFToolkit:DataGridTextColumn</pre>

x:Uid="Custom:DataGridTextColumn\_1" Header="Bill" Width="100"

Binding="{Binding Name}"/>

<!--显示账单到期日期的文本列,使用了自定义的转换器DataConverter-->

<WPFToolkit:DataGridTextColumn</pre>

x:Uid="Custom:DataGridTextColumn\_2"
Header="Due Date" Width="100"

Binding="{Binding DueDate,

Converter={StaticResource DateConverter}}"/>

<!--显示账单总数的文本列-->

<WPFToolkit:DataGridTextColumn x:Uid="Custom:
DataGridTextColumn\_3"</pre>

<!--显示账单添加日期的文本列,使用了自定义的转换器DataConverter-->

<WPFToolkit:DataGridTextColumn x:Uid="Custom:
DataGridTextColumn\_4"</pre>

 $\label{eq:header} H\ e\ a\ d\ e\ r=\ ''\ A\ d\ d\ e\ d\quad O\ n\ ''\quad W\ i\ d\ t\ h=\ ''\ 1\ 0\ 0\ ''$  Binding="{Binding AddedOn,

Converter={StaticResource DateConverter}}"/>

<WPFToolkit:DataGridComboBoxColumn
Header="Status" Width="75"</pre>

SelectedItemBinding="{Binding BillStatus}"

ItemsSource="{Binding Source={StaticResource
mvEnum}}"/>

</WPFToolkit:DataGrid.Columns>

</WPFToolkit:DataGrid>

## 4 其他关键技术(Other key technologies)

#### 4.1 BooleanSwitch

BooleanSwitch是定义在System.Diagnostics命名空间中的一个开关变量,用来控制调试和跟踪的输出<sup>门</sup>。该开关被定义在app.config配置文件的appSettings配置节中,可以通过该参数来获取应用程序运行时的跟踪信息。appSettings配置节代码如下:

<appSettings>

<add key="Culture" value="zh-CN" />

<!--定义区域信息[可选值有zh-CN, ro-RO,

en-US]-->

/>

<add key="ArchivePath" value="archive.xml"</pre>

<!---存档路径--->

<add key="ShowBillsFromDays" value="31" />

<!--显示最近31天以来的账单信息-->

<add key="ClientSettingsProvider.ServiceUri"</pre>

value=""/>

<!---跟踪信息--->

<add key="BooleanSwitch" value="true"/>

<!---跟踪文件路径--->

<!--跟踪文件最大大小1045680b==1 MB-->

<add key="SizeOfTrace" value="1045680"/>

</appSettings>

配置代码将BooleanSwitch设置为true,表示允许启用跟踪功能。PathToTraceFile表示跟踪信息的输出路径,SizeOfTrace表示最大跟踪文件大小,超过这个大小的文件将被删除重建。

#### 4.2 Trace

Trace类位于System.Diagnostics命名空间,提供了一组帮助跟踪代码执行的方法和属性,是为了在不打断程序的调试或跟踪下,用来记录程序执行的过程<sup>[8]</sup>。

Trace产生的信息可以被写入文本文件、XML或者记事本文件中。Trace的功能强大之处是跟踪监听器。使用监听器,是的Trace产生的调试信息可以直接写入监听器中,由监听器统一进行操作。在.NET中提供了三种类型的监听器,DefaultTraceListener提供了对于传统的OutputDebugString API的封装,将信息写到输出窗口中;EventLogTraceListener将调试信息写入到事件日志中;TextWriterTraceListener将调度或跟踪信息写入到文本文件中。可以使用下面的代码向Trace中添加一个监听器:

//实例化TextWriterTraceListener

TextWriterTraceListener listener=new TextWriterTra ceListener("MyTrace.txt");

Trace. AutoFlush=true:

//自动写入

Trace.Listeners.Add(listener); //将监听器加入Trace的Listener集合中

Trace.WriteLine(123, "Category 1"); //向监听器中写人消息

#### 5 结论(Conclusion)

本文介绍了使用WPF技术实现了一个账单管理程序, 其核心是分层技术在WPF应用程序开发中的具体应用。依 据软件分层的架构,自底向上分别实现了数据访问层、业务 逻辑层和表示层,通过接口来实现层与层之间的解耦,通过 XLINQ来查询XML档、多线程中的读取和写入模式在实际工 作中的运用、实现跟踪调试信息写入到文本文件中。

### 参考文献(References)

- [1] 刘波,朱丽华.基于Java语言的超市账单管理系统的设计与实现[]].安阳工学院学报,2016(06):34-35.
- [2] 骆剑华.T型账户在编制银行存款余额调节表教学中的应用 [[].企业导报,2016(14):98-99.
- [3] 张志刚.会计新手处理乱账方法及实例[J].农村财务会计,2018(08):45-46.
- [4] 张伟.基于VFP技术的银企自动对账系统设计[J].财会月刊,2014(19):12-13.
- [5] 马莉.电子账单系统在集装箱码头管理企业的应用[J].中国水运,2017(02):83-84.
- [6] 徐湛.基于条码数字化的电费账单管理系统建设[J].科技创新导报,2009(27):28-29.
- [7] 李霞,郭肇毅.家庭财务管理系统的设计与实现[J].信息与电脑(理论版),2016(23):75-76.
- [8] 张琳莹,张云洁.家庭财务管理系统设计与实现[J].电脑知识与技术,2018(11):104-105.

#### 作者简介:

邹 珺(1981-),女,硕士,讲师.研究领域:软件开发.