

基于 Visual Basic 的教师课表自动汇总系统开发

刘信香

(中原工学院 图书馆, 河南 郑州 450007)

摘要: 根据实际工作的需要, 开发了一套基于 Visual Basic 程序的教师课表自动汇总系统。该系统可自动读取课表数据, 并将读取的数据汇总后自动生成总课表文件, 可代替人工的繁琐重复劳动, 具有效率高、方便快捷的特点。

关键词: 高校; 课表; Visual Basic; EXCEL

中图分类号: TP311.56

文献标识码: A

文章编号: 1674-7720(2012)13-0010-03

Development of teacher schedule automatic collection system based on Visual Basic

Liu Xinxiang

(Library of Zhongyuan University of Technology, Zhengzhou 450007, China)

Abstract: In this paper, according to the need of practice, a set of teachers' schedules automatic collection system is developed based on Visual Basic program. The schedule data can be read automatically, and the total schedule file can be generated automatically after the data be collected in this system. This system can replace the artificial tedious duplication of effort, with high efficiency, and convenient.

Key words: university; schedule; Visual Basic; EXCEL

在高校, 因为工作需要, 二级院系每到学期的开学初, 都需要制作一份本院系所有任课教师上课安排的总课表。每位教师的课表可从学校的教务系统中自动导出, 但不少高校, 总课表往往却不能从系统中自动生成, 而需要教学秘书手工完成。虽然总课表也是一张课表, 但它是将所有课表进行汇总而成, 要靠手工来完成课表的汇总工作, 需要将本院系所有任课教师的课表都打印出来, 一边在计算机上复制、粘贴, 一边进行核对, 以免出现遗漏或重复。这样不仅需要耗费大量的纸张, 而且工作量大、效率低, 一张总课表往往需要一周的时间才能制作完成, 同时汇总的课表差错多, 给实际使用时带来很多问题。本文根据实际工作的需要, 结合课表的特点, 利用 Visual Basic6.0 开发了一套教师课表自动汇总系统^[1-2]。

1 自动汇总系统总体设计

1.1 课表结构分析

本文以中原工学院使用过的清华紫光教务系统为例进行说明。在系统设计之前, 首先要分析课表的结构。

由教务系统导出的教师个人课表是 EXCEL 文件, 图 1 是从课表中截取的某一节课的安排示意图。从图 1 中可以看出, 对每一节课的安排需占用 EXCEL 表中 8 行表格, 第一行表示的是该课程名称和课程号, 后面依次是上课地点、上课周次和上课班级等信息。在 VB 程序读取课程安排时, 每节课都需要连续读取 8 行的数据, 然后将读出的这 8 行数据连接起来, 就构成了一节课的完整安排, 经适当整理后输出到总课表中。

科技英语(9903034008-0)
南区 3 号教学楼 333
1~15 周上(2 学时)
机自 081 机自 082 机自 083

图 1 某一节课的安排示意图

1.2 程序设计思路

先为总课表建立一个二维数组来保存相关数据, 数组的每个成员都代表总课表中的某一节课。在将每张课表中的每节课安排读取后都汇总到对应的数组成员中。这样, 在将所有课表读完后, 总课表的数据汇总工作也就完成了。然后, 先根据所需要的总课表格式生成一张

空白总课表,再将二维数组中的数据放入相应的表格内,一张总课表就制作出来了。

1.3 程序设计流程

课表汇总系统的设计,首先要读出教师名称和人数,人数是作为后面读取课表时判断是否将所有课表读完的依据。然后是顺序读每张课表,读取课表时从周一读到周六,每个工作日的数据是从第1大节读到第6大节。将所有课表读完后,再将汇总到二维数组的数据放入总课表中对应的表格内,最后保存总课表。图2是程序设计流程框图。

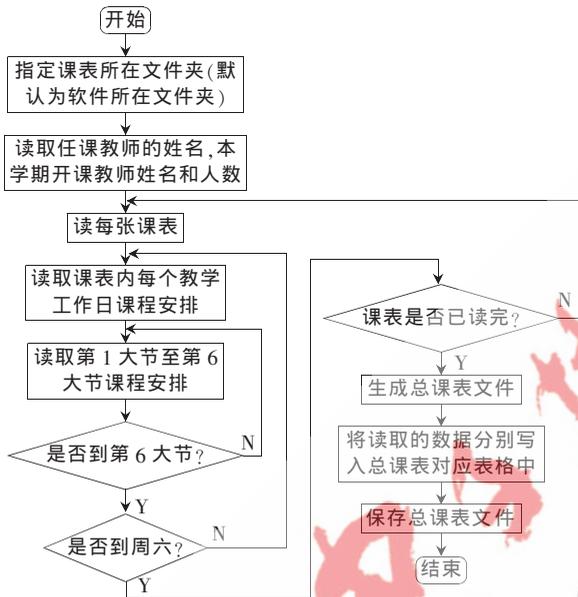


图2 程序设计流程框图

2 软件主要功能模块的实现

2.1 VB与EXCEL数据的链接

VB(Visual Basic)是一种应用十分广泛的程序开发工具,具有可视化、事件驱动编程的特点,它在界面设计、绘图、制表、运算、通信和多媒体开发方面都具有简单易行、功能强大等优点。VB具有强大的数据处理功能,提供了多种数据访问方法,可以将EXCEL数据表视为一个数据库,方便地进行访问和处理^[3]。

本系统利用面向对象的ADO技术实现对数据库(即EXCEL表)的访问。ADO(ActiveX Data Object)是ActiveX数据对象,是Microsoft公司开发数据库应用程序面向对象的新接口。VB利用ADO的Connection和Recordset对象可以直接访问数据,也就是ADO从一个数据源中获得记录只需一个连接和一个记录集^[4]。其过程如下:

(1)在工程中引用“Microsoft Excel 11.0 Object Library”,然后在窗体的通用声明部分分别定义一个Connection对象和RecordSet对象:

```
Dim cn As ADODB.Connection
Dim rs As ADODB.Recordset
```

(2)在程序中设计一个EXCEL表读取模块,新建一个Connection对象的实例并将它赋给cn数据源,新建一个Recordset对象的实例并使用Recordset对象的Open方法打开课表。

```
Set cn=New ADODB.Connection
Set rs=1New ADODB.Recordset
```

```
With cn
    .Provider = "Microsoft.Jet.OLEDB.4.0"
    .ConnectionString="Data source=" & address_file &
";Extended Properties='Excel 8.0;HDR=No;IMEX=1'"
    //address_file: EXCEL文件位置
    .CursorLocation = adUseClient
    .Open
End With
```

这样Visual Basic程序就与EXCEL表之间建立了联系,也就可以对表中的数据进行处理了。

2.2 课表数据的读取

程序在与EXCEL表建立链接之后,即开始对课表中的数据进行读取。先定义一个二维数组来保存每节课的安排,每张课表同一时间的课程安排在被读出后都追加到同一个二维数组成员中。其过程如下:

```
Public course_each(5, 6) As String
    //定义二维数组,课程(节,星期)
Dim kecheng As String
    //保存读取的某节课中的数据
Dim week_course As String
    //星期几
Dim counter_row As Integer
    //当前所读取的某节课单元格行数
```

读EXCEL表中的数据

```
For j = 1 To 6
    //读一周中的某天
    rs.MoveFirst
```

```
Select Case j
    //选择星期几
    Case 1
        week_course="星期一"
    Case 2
        week_course="星期二"
    Case 3
        week_course="星期三"
    Case 4
        week_course="星期四"
    Case 5
        week_course="星期五"
    Case Else
        week_course="星期六"
End Select
```

```
For i=1 To 5
    //读一天中的5节课
```

```

counter_row=0
kecheng="" //读取并保存一节课中的数据
Do While counter_row<8
    //读取每节课占的 8 个单元格
    counter_row=counter_row+1
    If kecheng="" Then
        kecheng=kecheng & rs.Fields
            (week_course).Value
    Else
        kecheng=kecheng & "," & rs.Fields
            (week_course).Value
    End If
    If rs.EOF Then
        Exit Sub
    Else
        rs.MoveNext
    End If
    course_each (i,j) =course_each (i,j) &
        name_string(1)& "," & Trim(kecheng)
//将已读完的某节课安排写入对应二维数组成员中
    End If
End If
Next i
Next j

```

2.3 课表数据的汇总

在将所有课表中的数据读出并保存到对应二维数组成员中后,就可生成总课表了。先创建 EXCEL 文档,定义需要的表格格式;然后再将数据写入对应的位置(创建总课表 EXCEL 文件并定义格式的程序段此处略)。下面是将数据写入总课表的程序段。

```

Public Sub writeToexcel(address_file As String)
Dim xlApp As Excel.Application //EXCEL 控件专用声明
Dim xlBook As Excel.Workbook
Dim xlsheet As Excel.Worksheet

Set xlApp = CreateObject("Excel.Application")
//创建 EXCEL 对象
Set xlBook = xlApp.Workbooks.Open(address_file)
//打开已经存在的 EXCEL 文件
xlApp.Visible = False
//设置 EXCEL 对象可见(或不可见)
Set xlsheet = xlBook.Worksheets("sheet1")
//设置活动工作表
For i = 1 To 5 //第 1~第 5 大节
    For j = 1 To 6 //星期一到星期五
        xlsheet.Cells(2 + i, 2 + j) = course_each(i, j)
    
```

```

Next j //向总课表中对应单元格写数据
Next i
xlBook.Save
xlBook.Close
xlApp.Quit //结束 EXCEL 对象
Set xlApp = Nothing //释放 xlApp 对象
End Sub

```

3 软件界面设计

软件的界面主要分为菜单区、工具栏区和课表汇总结果显示区。本系统的软件界面的工具栏是仿 OFFICE 风格,并提供了一些数据处理功能。打开按钮是指定课表所在位置,保存按钮是将最后汇总结果以 EXCEL 文件的形式存到计算机中。此外,软件还提供了剪切、复制、粘贴、撤销、查找、插入删除行、运行、放大、缩小、原始大小等功能按钮,以方便使用。在软件下方状态栏的左边显示的是教师课表存放的位置,右边显示的是开课教师的人数。通过菜单栏中的教师名单选项还可以查看所有开课教师的详细名单,以方便对结果的检查和核对。系统软件界面的主要功能区如图 3 所示。



图 3 课表自动汇总系统软件界面主要功能区

自动汇总系统的开发,较好地实现了对课表数据的读取和汇总,具有方便快捷的特点,大大提高了工作效率,减轻了工作人员的重复性劳动,并且提高了汇总课表的准确性。该系统在应用中效果较好。

参考文献

- [1] 徐谡,徐立.Visual Basic 应用与开发案例教程[M].北京:清华大学出版社,2005:223-225.
- [2] 林桂明.用 VB 开发中职学校国家助学金管理软件[J].湖南科技学院学报,2010(8):68-71.
- [3] 高春艳,李艳.Visual Basic 数据库开发关键技术与实例应用[M].北京:人民邮电出版社,2005:238-241.
- [4] 陈虹颐,何春.VB6.0 中利用 ADO 对象实现数据库编程[J].甘肃科技,2007(4).

(收稿日期:2012-02-06)

作者简介:

刘信香,女,1973 年生,本科,助理馆员,主要研究方向:信息管理。