

文章编号: 2096-1472(2017)-06-31-04

基于.NET技能抽考系统的设计与实现

夏小翔

(鄂州职业大学计算机学院, 湖北 鄂州 436000)

摘要:近年来,技能抽考在高校正开展得如火如荼,技能抽考重在“抽考”,要抽取指定数量的学生参加技能考试,对于参加考试的学生,还要从题库中随机抽取考试试题,为了让学生无论抽到还是没有抽到,参加考试的学生无论抽到什么样的试题都没有怨言,公平性就显得至关重要,基于.NET技能抽考系统利用随机数解决了公平性的问题。在抽取学生时,利用循环的嵌套保证了抽取学生不重复的问题;另外,本系统还能快速将EXCEL中的文件信息导入到数据库,提高了系统应用的广泛性和实用性;此外,在保存更新后的数据时,后台提供一次性存取,减少了数据库访问的次数,提高了数据库的访问效率,本软件还支持界面和文本文件之间数据的读取。

关键词: .NET框架; ADO.NET技术; 数据库访问; 技能抽考; 内存

中图分类号: TP391 **文献标识码:** A

Design and Implementation of the.NET-Based Skills Examination System

XIA Xiaoxiang

(Dept. of Computer and Technology, Vocational University of Ezhou, Ezhou 436000, China)

Abstract: In recent years, skills examination in colleges and universities are carried out like a raging fire. The key is to extract skills examination, which means to extract a specified number of students to take the examination and randomly extract the examination questions from the question bank. Fairness is quite important to avoid complaints from the students, whether they are extracted to take the examination and no matter what kind of questions are randomly extracted, which is achieved through the random number based on the .NET skills examination system. During the process of student extraction, Repetition is effectively avoided through loop nesting. Additionally, the system can quickly import EXCEL file information into the database, which improves the universality and practicality of the system. When saving the updated data, the skills examination system implements a one-time access in the backstage, which reduces database access and improves the efficiency of database access. The system also supports the reading between the interface and text files.

Keywords: NET framework; ADO.NET technology; database access; skills examination; memory

1 引言(Introduction)

随着软件技术在社会各领域的应用,为了满足信息化建设的需要,计算机程序设计语言的教学工作在高校中的地位日益突现。C#开发技术以其简单、实用、易上手等特点被越来越多的软件开发人员所青睐。本文基于.NET技术设计和开发了技能抽考系统。

2 关键技术(Key technology)

(1).NET技术^[1]

.NET框架体系结构:.NET框架是微软开发战略的核心内容,.NET框架基础类库是一个综合性的可重用类型集合,它为开发人员提供了一个统一的面向对象、层次化并可扩展的编程接口,该类库是生成.NET应用程序、组件和控件的基础。

(2)ADO.NET技术^[2]

ADO.NET技术:ADO.NET是微软在.NET框架中提供的最新的一种数据存储技术,它提供了一套完整的访问数据库类的集合。其中最常用的是Connection(连接数据库)、Command(执行数据库操作命令)、DataAdapter(数据适配器)、DataReader(数据阅读器)和DataSet(数据集)。

3 系统实现(System implementation)^[3,4]

(1)系统的主要功能模块

①试题集:该窗体显示所有技能大赛试集。

②试题集录入管理:该窗体可以录入试题题号、试题(包括文字和图片),并能显示已经录入的试题。

③学生信息录入与管理:包括录入学生学号、姓名、性

别、家庭住址、家庭电话、民族、身份证号、出生日期及班级,并能对学生信息进行添加、修改、删除、保存、取消等操作。还能选择查询条件,并根据查询条件输入查询值进行查询。

④从EXCEL导入数据到数据库:包括从EXCEL中导入数据到数据库,往数据库中添加、修改、删除记录,能将界面数据一次性保存到数据库等功能。

⑤从数据库导出数据到EXCEL:可以选择班级,然后根据班级导出数据到EXCEL,或者直接对数据进行打印输出。

⑥抽取学生:根据输入学生的数量,抽取指定数量的学生,要求抽取学生不能重复。

⑦抽取试题:要求输入学生姓名,先要检查该学生是否存在,如果存在,就显示随机抽取的题号和题目具体信息。

⑧学生信息精确查询:可根据需要选择查询条件,然后输入对应值,并可以查询到指定的记录。

⑨学生信息模糊查询:可根据需要选择查询条件,然后输入对应查询值,查询值的输入并不要求完全精确,只要求关键字匹配,并可以查询到指定的记录,如果记录不存在,会有恰当的提示信息。

⑩班级信息浏览:可根据需要选择对应班级,并能浏览到相应班级的学生信息。

此外,该技能抽考系统还提供了数据备份、数据还原、冗余数据清理等功能。

(2)业务流程图^[5]

根据系统功能的关联关系,业务流程图如图1所示。

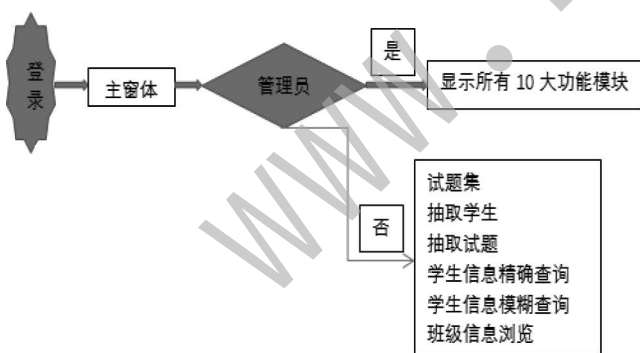


图1 技能抽考系统业务流程图

Fig.1 Skills examination system business flow chart

(3)数据存储设计^[6]

本系统涉及的数据有两类:一类是学生信息、试题集等基本文字信息;另一类是图片,这类图片主要是将试题内容拍成图片,然后将图片以路径的形式保存到数据库中,需要显示的时候,可以将该图片显示在相应的图片框中。另外由于技能抽考涉及各个专业方向,解答问题时需要较多专业软

件的编辑环境,所以对于客观题只提供试题,答题时需要相应专业软件支撑。

阅卷时分选择题部分和主观题部分,选择题部分提供机器阅卷,主观题部分答案只提供分析、思路,需要借助人工辅助完成。

4 核心算法设计(Core algorithm design)

(1)抽取“技能抽考”分析题核心算法设计

抽取“技能抽考”分析题的核心算法,关键有三点:

①用随机数类Random产生随机数,根据产生的随机数在数据库中查询指定的分析题题号,并读取指定题号的题目,将题目内容显示在文本框或图片框中,因为用到了随机数,所以能体现出技能抽考的公平性和技能抽考“抽考”的含义。

②定义全局变量j,设定分析题抽取次数最多为3次,3次抽题机会既给考生抽取试题提供灵活性,也可防止考生由于知识的局限性,导致抽到的试题不会做,但又为了保证考试公平性,不能让考生无限制地抽取下去,因而设置了技能抽考“抽考”次数的上限。

③为了记录学生抽取分析题的次数,学生每次抽取试题,都会将抽取试题学生的姓名、题号、抽取时间、题目写到d\xia.txt文本文件中作为记录保存,等所有学生抽取试题结束时,可打印出该文本文件,便于阅卷时综合评分,一般抽取次数越少,可适当加分,抽取次数越多,可适当扣分。

核心算法代码如下:

引入数据库类命名空间:using System.Data.SqlClient;

定义全局变量j,并给全局变量赋初值,用来统计

抽题次数

j=0;

设定抽题次数最多为3次

if (j<=3)

{连接数据库

SqlConnection con=new SqlConnection("server=(local); database=数据库名;Integrated Security=true;");

if (con.State==ConnectionState.Closed)

{con.Open();}

连接表

SqlDataAdapter da=new SqlDataAdapter("select * from 学生表 where 姓名='"+txtName.Text+"'",con);

DataSet ds=new DataSet();

int a=da.Fill(ds,"学生");

con.Close();

判断内存表中是否有查询到的记录

```

        if (a==0)
        {
            MessageBox.Show("查无此学生, 请重新输入", "提示信息");
            txtName.Focus();
            return;
        }
        else
        {
            //产生1至100之间的随机数, 根据产生的随机数(作为题号)查询数据库中指定的技能抽考试题
            int i;
            Random rd=new Random();
            i=rd.Next(1,100);
            txtTiHao.Text=i.ToString();

            SqlDataAdapter da1=new SqlDataAdapter("select 题目 from 技能抽考试题 where 题号='"+txtTiHao.Text+"'",con);
            Da1.Fill(ds,"试题");
            txtTiMu.Text=ds.Tables["试题"].Rows[0][0].ToString();

            MessageBox.Show("这是第"+j.ToString()+"次抽题, 最多可以抽取3次!", "提示信息");
            j++;
            //将抽取试题学生的姓名, 题号, 抽取时间, 题目写到d\xia.txt文本文件中作为历史记录
            FileStream fs=new FileStream("d:\\xia.txt", FileMode.Append);
            StreamWriter sw=new StreamWriter(fs);
            string str="姓名: "+txtName.Text+" "+
            "题号: "+txtTiHao.Text+" "+
            "抽取时间: "+DateTime.Now.ToString()+" "+
            "题目: "+txtTiMu.Text;
            sw.WriteLine(str);
            sw.Close();
        }
    }
}

```

(2) 抽取“技能抽考”学生(学生不能重复)核心算法设计^[7]

抽取“技能抽考”学生核心算法, 关键有两点:

①新建一个整型数组, 并且将产生的随机数依次存放在整型数组中, 但产生的随机数一定不能重复。本算法的思路是通过循环依次产生随机数, 并分别对数组元素赋值, 当再次产生随机数时, 通过循环的嵌套, 将新产生的随机数分别与先前产生的随机数进行对比, 如果发现新产生的随机数先前出现过, 则此次生成的随机数无效, 如此反复, 直到产生

规定数量并且没有重复的随机数为止。

②通过循环, 根据数组中的随机数分别读数据库, 查询指定的记录, 并且将查询到的学生信息显示在数据视图中。

核心算法代码如下:

```

        StudentNumber=Int32.Parse(txtStudentNum.Text);

        int[] stu_n=new int[StudentNumber];
        //产生指定数量学生的随机数, 并依次给整型数组元素赋值
        Random rd=new Random();
        for(i=0;i<StudentNumber;i++)
        {
            stu_n[i]=rd.Next(0,StudentNumber);
            //通过循环, 检测新产生的随机数是否与前面产生的随机数相同, 如果相同, 则新产生随机数无效, 该循环可保证抽取学生不重复
            for (j=0;j<i;j++)
            {
                if (stu_n[i]==stu_n[j])
                {
                    i=i - 1;
                }
            }
        }
        //连接数据库并根据学生编号查询数据库表中的数据
        if (con.State==ConnectionState.Closed)
        {
            con.Open();
        }
        DataSet ds=new DataSet ();
        for(i=0;i<StudentNumber;i++)
        {
            da=new SqlDataAdapter("select*from 学生表 where 编号='"+stu_n[i]+"'",con);
            da.Fill(ds,"被抽取学生");
        }
        con.Close();
        dgvSelectedStudent.DataSource=ds.Tables["被抽取学生"];
    }
}

```

(3) “技能抽考”从EXCEL中导入数据核心算法设计^[8]

“技能抽考”导入EXCEL中数据核心算法, 关键有二点:

①把EXCEL当成一个数据库, 连接该EXCEL数据库, 从EXCEL表中导入学生信息放到内存新建的表中, 并且将内存表中数据在界面数据视图中显示出来, 难点是连接EXCEL数据库的方式与SQL Server数据库的方式有比较大的不同。另外, 编程时有一个小的技巧, 可以把工作簿类比为数据库, 工作簿中的表类比为数据库中的表, EXCEL表中的行、列与SQL Server数据库表中的行、列对应起来。

②可以在数据视图中对导入信息进行添加、修改和删除操作,操作完成后可将所有记录一次性保存到后台数据库中,此时,通过循环的嵌套依次访问表中每一行的每一个字段,外循环依次扫描行,内循环依次扫描当前行的每一列,以记录的方式保存到数据库中。

该核心算法的优点:可以避免频繁地访问后台数据库,可有效地提高数据库的访问效率。

核心算法代码如下:

引入命名空间:

```
using System.Data.SqlClient;
```

```
using System.Data.OleDb;
```

连接EXCEL,并读取EXCEL中数据信息

```
string strConn="Provider=Microsoft.Jet.OLEDB.4.0;"+
"Data Source="+文件路径及文件名+".xls;"+
"Extended Properties=Excel 8.0;";
```

```
OleDbConnection con=new OleDbConnection(strConn);
```

```
string strExcel="";
```

```
OleDbDataAdapter da=null;
```

```
DataSet ds=ds=new DataSet();
```

```
strExcel="select * from[Sheet1$]";
```

```
da=new OleDbDataAdapter(strExcel,strConn);
```

```
if (con.State==ConnectionState.Closed)
```

```
{con.Open();}
```

```
da.Fill(ds,"t1");
```

```
this.dgvInputInformation.DataSource=ds.Tables["t1"];
//将内存表中数据在数据视图显示
```

可在数据视图中对数据信息进行添加、修改和删除

将内存新建表中数据保存到数据库

```
SqlConnection conn=new SqlConnection("server=
(local);database=数据库名;integrated security=true;");
```

```
SqlCommand com;
```

```
com=conn.CreateCommand();
```

```
com.CommandText="insert into 学生表(学号)
values(@学号)";
//这里只举出一个字段为例
```

```
com.Parameters.Add(new SqlParameter("@学号",
SqlDbType.NChar));
```

```
for (int i=0;i<dt.Rows.Count;i++)
```

```
{com.Parameters["@学号"].Value=dt.Rows[i][0].
```

```
ToString();
```

```
if (conn.State==ConnectionState.Closed)
```

```
{conn.Open();}
```

```
com.ExecuteNonQuery();
```

```
conn.Close();
```

```
}
```

5 结论(Conclusion)

本文提出的基于.NET技能抽考系统的设计方案已经实现,经鄂州职业大学各学院验证完全可行,能稳定运行,安全性好,使用方便。尤其在抽取学生、抽取试题方面充分体现出公平性原则,得到了在校师生的一致好评;此外,该系统能够快速导入EXCEL表中数据信息,从而间接地解决文件在数据库中的存储问题。整个系统开发过程中,全程都有在校学生参与,由于该系统具有很强的针对性、实用性和使用的及时性,因而大大地激发了参与学生在开发系统过程中的开发兴趣和开发热情。

参考文献(References)

- [1] Sheng xuna Chen,Design and implementation of Enterprise Training Management System Based on .NET[J].China Communications,2016(3):193-202.
- [2] Shu Guo Cao,Grade Exam Management System Design and implementation Based on ADO.NET[J].Advanced Mechanics And Materials,2013(347):2930-3936.
- [3] Ling Ling Zhao,Qiao Yun Sun,Xiao Xiao Liu.The Design and Implementation of the Colleges Network Examination System Based on B/S Architecture[J].Applied Mechanics and Materials,2014,3082:543-545.
- [4] Xin Yu WANG,Qing Song Zhang. Design and Implementation of Library Management System Based on .NET[J].Advanced Materials Research,2013,2542:760-762.
- [5] 侯春贡,任华,林忠会.C#程序设计[M].北京:航空工业出版社,2016.
- [6] 胡锦丽,雷筱珍,刘芳.数据库原理及应用案例教程[M].北京:北京大学出版社,2015.
- [7] 王晶晶.C#面向对象程序设计[M].北京:机械工业出版社,2016:129.
- [8] 李志云,等,Visual C# 2005数据库开发案例教程[M].北京:机械工业出版社,2013:84-88.

作者简介:

夏小翔(1973-),男,硕士,副教授.研究领域:群体智能算法,数据库系统开发,嵌入式系统。