

C#实现的简易身份证号码校验程序

陈明娟

(四川省广安市代市中学校 四川 广安 638000)

[摘要]为保证数据准确性和安全性,在录入身份证号后需及时进行离线、批量校验,提升学校师生工作效率。本文基于国标GB 11643-1999对公民身份证号码的相关规定,用C#实现了一个简易的身份证号码验证程序,检验用户输入的字符串是否符合国标的相关规定。

[关键词]身份证号码;批量;离线;校验

[DOI] 10.12252/j.issn.2096-6288.2021.08.193

一、前言

学校经常会用到各种表册,这些表册中需要录入学生身份证号码。现行公民身份证号码(后续简称身份证号)长18位。数据量大了之后很容易搞混,录入时也容易录入错误,为保证数据准确性,需要在录入身份证号后及时校验。在百度上可以找到很多在线网页,可以进行身份证号校验,但在线校验需要有网络,数据多了后一个一个复制粘贴效率也很低。我们需要一个不会保留数据(确认数据安全),可以离线运行,批量校验的工具。最后我决定自行编写一个应用来实现这些目标。于是就有了这篇论文。

二、为什么选择.net来实现

本次决定基于.net framework4.0平台,采用C#语言来实现。之所以不采用Python,原因有3个。一是需要安装额外的运行时(python组件);二是python不能简单实现图形操作界面;三是因额外组件带来了更多的安全隐患。就我校现用的计算机桌面环境而言,操作系统全是windows。大多数为windows 7,少部分为windows XP,还有一部分windows 10。这些系统中大多预装了.net framework4,即便系统未安装也可以在编译后打包成安装程序一键安装;应用编译好后不用额外安装运行时,一次编译即可随处运行,编译后应用也很小(10KB左右)。基于.Net平台也可以很方便地实现图形操作界面,即便平台本身有漏洞,系统的update也可以自动打补丁提升安全性。之所以采用.net framework4.0而不是更高版本,是因为XP系统不支持更高版本。最终我决定基于.net framework4.0平台,采用C#语言来编写应用。

三、具体实现

《中华人民共和国国家标准 公民身份证号码(GB 11643-

1999)》^[1]规定公民身份证号码由本体码和校验码两部分组成。

由本体码可以得出唯一对应的校验码。本文所称身份证号码校验即为校验给定字符串是否符合本体码、校验码的对应关系,如符合即视同为正确的身份证号,否则认为不是正确的身份证号。

校验过程分为7步:

(1) 输入字符串(下称S)是否为空字符串,若是则为错误身份证号;

(2) S的长度是否等于18位,若不等于则为错误身份证号;

(3) (1.指S的从左往右第i位字符)是否是数字,若不是数字则为错误身份证号;

(4) 是否是数字、X、x之外的其他内容,若是其他内容则为错误身份证号;

(5) 对加权求和得到SUM:。指S的从左往右第i位字符,为对应加权数值;

(6) 对SUM取模求余得到M: $M = \text{SUM} \bmod 11$

(7) 比对M是否等于,若不等于则为错误身份证号,若相等则视为正确身份证号。

部分代码如下:

```
using System;
using System.Windows.Forms;
using System.Threading.Tasks;
using System.Collections.Generic;
using System.Collections.Concurrent;
using System.Text;
using System.Threading;
```

```

.....
/// <summary>
/// 校验身份证
/// </summary>
/// <param name=" CardID" >身份证号</param>
private void Check (string CardID)
{
if (string.IsNullOrEmpty (CardID) ) return;
if (CardID.Length != 18)
{
errMsg.Enqueue ($" {CardID}\t长度不为18" );
return;
}
if (ContainsIllegalCharacters (CardID) )
{
errMsg.Enqueue ($" {CardID}\t包含非法字符" );
return;
}
int sum = 0;
for (int i = 0; i < 17; i++)
sum += Char2Number[CardID[i]]* wi[i];
if (cscCode[sum % 11]!= CardID[17])
errMsg.Enqueue ($" {CardID}\t不符合编码规则" );
}
/// <summary>
/// 包含非法字符 (非[0-9xX]的字符) ?
/// </summary>
/// <param name= "str" >待检测的字符串</param>
/// <returns>包含则true, 否则false</returns>
private bool ContainsIllegalCharacters (string
str)
{
foreach (var ch in str)
{
if (!Char2Number.ContainsKey (ch) ) return true;
}
return false;
}
/// 身份证校验时对应位置数字的加权值
private readonly int[]wi = { 7, 9, 10, 5, 8, 4, 2,
1, 6, 3, 7, 9, 10, 5, 8, 4, 2 };
// 身份证校验时, 末尾余数对应的校验码
private readonly char[]cscCode = { '1', '0',
'X', '9', '8', '7', '6', '5', '4',
'3', '2' };
/// 字符快速转数字
private readonly Dictionary<char, int> Char2Number
= new Dictionary<char, int> ()
{
{ '0', 0}, { '1', 1}, { '2', 2}, { '3',
3}, { '4', 4}, { '5', 5}, { '6', 6}, { '7',
7}, { '8', 8}, { '9', 9}, { 'X', 10}, { 'x', 10
}
};
/// 要追加到rtResult中的文字队列。避免多线程时丢失
ConcurrentQueue<string> errMsg = new
ConcurrentQueue<string> ();

```

四、注意事项

因使用场景为学校, 大多数时候身份证号能填写正确, 即便错误, 更多错误出现在从出生日期开始的内容当中, 故本应用未校对6位数字地址码。若应用到其他场景应在本文代码的基础上补足6位数字地址码的校验。

参考文献

[1] 国家质量技术监督局.GB 11643-1999, 公民身份号码[S].

作者简介:

陈明娟, 高中信息技术一级, 四川省广安市, 四川省广安市代市中学校。