



# 广东工业大学

## 设计报告

### 学生签到转换软件

学 院 自动化学院

年级班别 2019 级自动化 8 班

学 号 3119001152

学生姓名 梁梓熙

2020 年 3 月 18 日



## 摘 要

因 2020 初受新型冠状病毒影响，各类学校纷纷采用线上直播的方式进行授课。随着各大学校的线上授课考勤制度的完善，需要老师们在学生签到之后将签到结果登记并反馈给学校。

考虑到授课软件生成的签到表格和学校规定的签到表格具有差异性，故设计一款专用的签到转换软件以代替麻烦耗时无意义的人工转换，节省老师的精力，提高授课质量。

**关键词：**线上授课，转换，签到表，考勤

## Abstract

Affected by the new coronavirus at the beginning of 2020, various schools have used online live broadcasts to teach. With the improvement of the online lecture and attendance system of major universities, teachers are required to register and report back to the school after the students sign in.

Considering the difference between the sign-in form generated by the teaching software and the school-specified sign-in form, a special sign-in conversion software was designed to replace the troublesome, time-consuming and meaningless manual conversion, saving the teacher's energy and improving the quality of teaching.

**Key words:** Online teaching, conversion, registration form, attendance

## 目录

设计报告.....	1
学生签到转换软件.....	1
<b>摘 要.....</b>	<b>1</b>
<b>Abstract.....</b>	<b>1</b>
<b>1 功能设计.....</b>	<b>3</b>
1.1 主要设计目的.....	3
1.2 辅助功能.....	3
<b>2 框架设计.....</b>	<b>3</b>
2.1 总体设计思路及分析.....	3
2.2 程序封装与拓展.....	4
<b>3 模块设计及实现.....</b>	<b>4</b>
3.1 模块设计与思路.....	5
3.2 模块实现.....	6
<b>4 调试及结果.....</b>	<b>8</b>
3.1 调试及改进.....	8
3.2 运行结果.....	10
3.3 性能评估.....	11
<b>5 总结.....</b>	<b>12</b>
<b>附录.....</b>	<b>13</b>
1. Sign-inTableConversion 签到表转换程序.....	13
2. Sign-inTableConversion2.0 签到表转换程序.....	13

# 1 功能设计

## 1.1 主要设计目的

在保证效率和消耗内存能使用户接受的情况下，实现从腾讯课堂软件生成的签到表格到广东工业大学签到表格的转换功能。减少老师浪费不必要的时间以把大部分精力投入到授课中，提高教学质量。

考虑到可能存在的各种情况，程序将不需要用户在运行时进行任何操作，只需要将资源表格和模板文件放在源代码同目录的文件夹中即可。

## 1.2 辅助功能

因为涉及到数据的转换，因此程序必定会收集所有学生的资料信息。利用这个特点，可以设计出许多额外的辅助功能。如：判断指定学生是否进行签到与在线时间，程序可以通过在处理中检索出用户指定的学生，从而返回该学生的信息，签到与在线时间。

因为为了尽可能减少程序的消耗内存，如同排序学生在线时间等在 Excel 表格可以实现的功能，在该程序中并没有实现。

# 2 框架设计

## 2.1 总体设计思路及分析

程序的主要思路为：提取——输出。相比起 提取——储存——输出 的程序思路，该程序能够减少平均约

$$\left(1 - \frac{N}{(32 \times 3) \times N}\right) \times 100\% \approx 98.96\%$$

的内存占用，在极端情况下能够减少约 99.50% 的空间占用量。

在程序速度性能方面，易得腾讯课堂的签到表格次序与广东工业大学签到表次序之间并无关联，且学生的备注之间具有较大的差异，故能够节约空间和时间的二分查找算法在该情况之中无法使用，而二叉树、红黑树、散列表、字符查找树等一系列查找数据结构虽然能带来非常可观的速度性能，但是考虑到学生数量较少(约 80~160 名)的情况

下，使用这些数据结构所带来的内存消耗和构造时间显然是不可接受的。故经过取舍后，采用在时间性能和速度性能上都比较折中的暴力算法。其能够在 提取——输出 模式下达到线性级别的处理时间。

## 2.2 程序封装与拓展

在 C 语言与 Python 语言的选取中，Python 虽然运行时间较慢但是在对 Excel 表格的处理上远远优于需要通过转换成.csv 后缀文件的 C 语言。故该程序采用 Python3.7 来制作。

对于封装，该程序将程序主要部分封装成类的形式来进行调用，保证了主代码干净整洁的情况下能够让其他程序员更容易维护这份代码。此外，本人对程序进行了大量的重构处理，以保证代码符合简洁易懂易维护的原则。

因方便不同用户对该程序在不同情况下的使用，该程序将部分变量单独引出，方便用户能够自己设定出自己适用的输出格式。

在了解到广东工业大学签到表的风格是固定的以后，特意设定一个资源文件夹，让程序不再是在目标表格上修改，而是复制资源文件的备份，在备份中修改后生成一个新的最终输出文件。



```
mark = '√' #签到成功的标记
```

图 1 可供修改的自定义输出

## 3 模块设计及实现

### 3.1 模块设计与思路

程序中包含两个大模块：录入模块与输出模块。

在录入模块中，程序将会打开用户指定的资源文件夹并依次读取源文件进行循环处理。根据腾讯课堂的签到排序风格，我们能得出教师本次授课的内容并在模板文件中进行关键字检测，直到匹配到模板文件为止；若无相关文件，则会默认使用第一个模板；若有多个重复签到表，程序则会跳过这个阶段。

设有  $N$  个模板，每次筛选前  $M$  为学生，则平均每次命中的搜寻所需时间为：

$$\frac{M}{2} \times \frac{N}{2} \sim MN$$

随后，程序会复制该模板并同时打开源文件与模板文件的备份并进行处理：读取源文件中每个学生的名字并在备份中找到对应位置，然后进行输出处理。

在输出模块中，程序将会复制一份模板文件，然后依次在文件中寻找对应学生所在的位置并进行修改更新。每个录入的学生将会被标记，当遇到找不到该学生的情况后，将会在表格的额外部分输出无法被识别的信息，以方便老师手动修改。若有学生使用多个备注，则会在该学生备注后依次添加所有的在线时间。

在程序完成后，会将源文件的时间、模板文件的课程、班级整合到一起作为输出文件，并且将源文件转移到历史记录文件中以方便下次转换时直接使用。

签到	备注		
√	96分钟	梁梓熙	101分钟
√	101分钟	这个是多余的人	101分钟
√	101分钟		
√	100分钟		
√	98分钟		
√	101分钟		

图 2 无法识别的学生

## 3.2 模块实现

### 3.2.1 初始化

程序将会根据用户的设定，将资源表格和模板文件分别通过 Python 的 xlrd 与 xlwt 库打开，然后通过 xlutils 库中的 copy() 函数得到模板文件的副本，并作为最终输出文件。除此以外，程序还制作了 targetSheet(), targetSheet() 方法快捷打开资源表格和输出文件。

```
"""签到表格转换"""
def __init__(self, sourceName: str, targetName: str):
    """初始化程序"""
    self.wrong = 0 #记录错误学生
    self.marked = [] #记录学生

    self.source = xr.open_workbook(sourceName)
    self.targetSource = xr.open_workbook(targetName, formatting_info=True) #打开
    self.target = copy(self.targetSource) #复制一个以供使用

def targetSheet(self):
    """获取目标表格"""
    return self.target.get_sheet(0)

def sourceSheet(self):
    """获取资源表格"""
    return self.source.sheet_by_index(0)
```

图 3 初始化及调用类方法

### 3.2.2 输入模块

程序首先在源文件夹中进行循环，对每一个资源表格进行处理。在选定了资源表格后，程序会用搜索课程的信息并进行匹配，若找到对应的模板文件则直接返回模板文件的名称；若未命中，则会返回第一个模板文件的名称。若有多个对应模板，则会提示并不做处理。

```
def getTargetName(sources: str) -> str:
    """获得对应的目标模板"""
    temp = xr.open_workbook(sources) #先打开资源文件
    string = temp.sheet_by_index(0).cell_value(6,2)[:4] #读取模板里面课程

    for root, dirs, files in os.walk('templates'): #在模板文件中查找
        if len(files) == 0:
            return None

        for i in range(len(files)):
            #判断后缀为xls的
            if os.path.splitext(files[i])[1] == '.xls':
                if string in files[i]:
                    for j in range(i+1, len(files)):
                        if string in files[j]:
                            return None
                    return files[i]

    for root, dirs, files in os.walk('templates'): #找不到时，默认使用第一个
        return files[0]
```

图 4 模板匹配函数



程序将会打开资源表格，然后在设定位置开始暴力输入：程序会将通过 Python 的 re 库中的 sub()函数将会删除输入中的所有非中文数字(如：8 班梁梓熙 将会得到资源字符串 班梁梓熙)，从而能够保证程序能够在无法预测的学生备注中尽可能地提高识别率。随后将该字符串返回到程序主函数中缓存。

```
def readStudentName(self,row: int,col: int):
    """读取第row行col列的学生名字"""
    string = self.sourceSheet().cell_value(row,col)
    if string == "": return None

    target = re.sub("[A-Za-z0-9!\@%\[\]\.\,\*\ ]","", string) #将所有非中文字符都删掉
    if self._finePos(target) == None: #找不到学生的位置则写入错误信息
        self._writeWrongStudent(row,col)
        return None

    return target
```

图 5 读取姓名的类方法

得到学生在线时间也是用类似的方法完成，因得到学生在线时间的前提是该学生必定存在于签到表之中，故得到时间的方法去掉了判断学生信息是否存在的步骤。同时因腾讯课堂软件中的在线时间字符串已经完善，故也省去了处理字符串的过程。

```
def readTime(self,row: int,col: int) -> str:
    """返回一个学生时间"""
    return self.sourceSheet().cell_value(row,col+4)
```

图 63 得到学生的在线时间

### 3.2.3 输出模块

输出分为三个步骤：找到学生位置，表格指针偏移，将目标字符串输入到表格中。

找到学生的位置是程序的内循环部分，代码使用一个循环来读取输出文件中的每行内容，并依次用资源字符串进行对比，如果发现目标字符串位于资源字符串中（即名字已配对）后则调用 writeStudent()、writeStudentTime()方法依次将学生的签到信息与在线时间记录在输出文件中。记录完毕后的学生将会被标记（程序采用的是标记该学生在输出表格中的位置），以检测到后续的重名/重复签到情况。

```
def writeStudentTime(self, name: str, time: str):
    """记录一个学生在线时间"""
    pos = self._finePos(name) #找到学生的位置列表

    if self.isMark(name): #出现重名情况
        self.targetSheet().write(self.wrong + 4, 10, name, self.center())
        self.targetSheet().write(self.wrong + 4, 11, time, self.center())
        self.wrong += 1 #更新指针

    self.targetSheet().write(pos[0], pos[1] + 2, time, self.center())
```

图 7 记录学生时间的类方法

在程序完成所有的信息转录后，程序将会输出 OK! 字符串来表示已经成功转换完毕，并会在代码文件夹中生成输出文件。

## 4 调试及结果

### 3.1 调试及改进

在第一次制作的时候，程序中处理学生签到信息是采用记录其前面的特殊代号来进行信息录入。这样的优点是能够将运行时间减少至常数级别，但是与之带来的后果是得学生不得不按照规定的格式进行改名。在与老师商量之后，抛弃了这一版本。

```
def writeStudent(self, num: int):
    """记录一个学生是否签到"""
    piece = self.maxStudents // 30 + 1 #最多有多少模块
    num = num - 1 #调整序号
    if (num // 30 + 1) % 2 == 1: #在左边模块
        if num < 30:
            self.targetSheet().write(4 + num, 3, self.mark, self.center())
        else:
            self.targetSheet().write(11 + 30 * ((piece - 1) // 2) + num % 30, 3, self.mark, self.center())
    else: #在右边模块
        if num < 60:
            self.targetSheet().write(4 + num % 30, 8, self.mark, self.center())
        else:
            self.targetSheet().write(11 + 30 * (piece // 2) + num % 30, 8, self.mark, self.center())
```

图 8 在常数级别下的信息录入

在第二次制作的时候，决定采取适用性更强的检测学生姓名进行匹配的方式完成信息录入。不过也带来了一定的缺点：无法检测重名情况和姓名相似的情况(如：张三和张三三会被认为是同一个人)

为了减少此类 BUG 带来的影响，程序加入了一个布尔型列表来记录学生是否已经完成信息的录入，当发现信息早已被录入的时候，此时缓存中的信息将会被判定为问题信息从而放入异常区域，并由老师来进行手动纠错。

```
def mark(self, name: str):  
    """标记一个学生"""  
    self.marked.append(self._finePos(name))    #标记学生已签到
```

图 9 标记类方法(\_finePos 为寻找学生在表格位置)

在后一次维护中，加入了修改格式框的类方法，以保证表格在被修改后依旧能不破坏原有的格式。

```
def center(self):  
    """使一个文本加框并居中"""  
    style = xw.XFStyle()  
  
    borders = xw.Borders()    #修改边框  
    borders.left = xw.Borders.THIN  
    borders.right = xw.Borders.THIN  
    borders.top = xw.Borders.THIN  
    borders.bottom = xw.Borders.THIN  
    style.borders = borders  
  
    alignment = xw.Alignment()    #居中  
    alignment.horz = xw.Alignment.HORZ_CENTER  
    style.alignment = alignment  
  
    return style
```

图 10 美化方法，返回一个风格变量

在 2020/3/24 的更新中，依据老师的建议，做出了较大的修改，主要更新在文件处理中。

程序现在从单纯的单文件输入输出，已经修改成了更加灵活的可以同时处理多个班级多个签到文件的程序。但代价为引入了较多的循环结构，造成了一定程度上性能的下降。

后根据老师的要求，依次更新了历史记录、可以重复统计重名学生、以课程名来匹配模板文件和更加清晰直观的输出文件名字功能。

至此，程序的初步制作已经完成，剩下的便是后续的更新与维护。

### 3.2 运行结果

程序能够保存在用户可接受的时间内完成对所有数据的输入与输出。值得注意的是，输出文件是会在每次运行程序后进行更新并覆盖掉原有的输出文件。因此当不关闭输出文件并尝试进行数据转换时，会发生越权报错。

A	B	C	D	E	F	G	H	I	J
22	徐荣华	程序设计	137 七班 庄泽洋	1.44E+17	是	2020/03/17 08:29:41	01分钟	否	0分钟
24	徐荣华	程序设计	138-7班-李婧曦	1.44E+17	是	2020/03/17 08:29:41	08分钟	否	0分钟
41	徐荣华	程序设计	139 8班 包志耀	1.44E+17	是	2020/03/17 08:29:51	01分钟	否	0分钟
74	徐荣华	程序设计	142 8班 邓楷翰 311900	1.44E+17	是	2020/03/17 08:34:09	06分钟	否	0分钟
25	徐荣华	程序设计	143 八班 冯世峰	1.44E+17	是	2020/03/17 08:29:41	01分钟	否	0分钟
77	徐荣华	程序设计	144 8班 郝亿康3119001	1.44E+17	是	2020/03/17 08:35:19	01分钟	否	0分钟
47	徐荣华	程序设计	145 8班 何超钱	1.44E+17	是	2020/03/17 08:29:51	01分钟	否	0分钟
78	徐荣华	程序设计	146 8班 何奕龙	1.44E+17	是	2020/03/17 08:35:49	05分钟	否	0分钟
52	徐荣华	程序设计	147 自8班 胡楷泽	8.48E+08	是	2020/03/17 08:29:59	09分钟	否	0分钟
71	徐荣华	程序设计	148 8班 黄永亮	1.44E+17	是	2020/03/17 08:32:19	03分钟	否	0分钟
2	徐荣华	程序设计	149 8 黄云浩	1.44E+17	是	2020/03/17 08:29:31	01分钟	否	0分钟
34	徐荣华	程序设计	150 8班 黄泽安	1.44E+17	是	2020/03/17 08:29:51	01分钟	否	0分钟
60	徐荣华	程序设计	153 八班 李鸿高	1.93E+09	是	2020/03/17 08:30:41	00分钟	否	0分钟
40	徐荣华	程序设计	154 八班 梁帆	1.44E+17	是	2020/03/17 08:29:51	01分钟	否	0分钟
75	徐荣华	程序设计	155 自8 梁林峰	2.73E+09	是	2020/03/17 08:34:28	08分钟	否	0分钟
18	徐荣华	程序设计	157 8班 林洪标	1.44E+17	是	2020/03/17 08:29:41	01分钟	否	0分钟
29	徐荣华	程序设计	159 八班 刘建发	1.44E+17	是	2020/03/17 08:29:41	01分钟	否	0分钟
70	徐荣华	程序设计	160 刘士瑜 AU8	1.44E+17	是	2020/03/17 08:32:19	04分钟	否	0分钟
35	徐荣华	程序设计	161 龙锦森8	1.44E+17	是	2020/03/17 08:29:51	01分钟	否	0分钟
3	徐荣华	程序设计	162 八班 罗杨茂	1.44E+17	是	2020/03/17 08:29:31	01分钟	否	0分钟
73	徐荣华	程序设计	163 八班 邱俊文	2.53E+09	是	2020/03/17 08:32:29	01分钟	否	0分钟
15	徐荣华	程序设计	164 自动化8班 苏元晟	1.44E+17	是	2020/03/17 08:29:41	01分钟	否	0分钟
14	徐荣华	程序设计	165 自动化8班 吴世聪	1.44E+17	是	2020/03/17 08:29:41	01分钟	否	0分钟
33	徐荣华	程序设计	166-八班-吴世楠	1.44E+17	是	2020/03/17 08:29:51	01分钟	否	0分钟
81	徐荣华	程序设计	167 8班 向国臣	1.44E+17	是	2020/03/17 09:01:06	44分钟	否	0分钟
68	徐荣华	程序设计	168 8班 肖博文	1.44E+17	是	2020/03/17 08:31:29	03分钟	否	0分钟
76	徐荣华	程序设计	169 廖孝耀3119001165	2.24E+09	是	2020/03/17 08:35:09	02分钟	否	0分钟
62	徐荣华	程序设计	170 8班 杨奕邦	1.44E+17	是	2020/03/17 08:30:49	09分钟	否	0分钟
23	徐荣华	程序设计	173 自八 钟子健	1.48E+09	是	2020/03/17 08:29:41	01分钟	否	0分钟
27	徐荣华	程序设计	174 自8 周扬希	2.94E+09	是	2020/03/17 08:29:41	01分钟	否	0分钟
10	徐荣华	程序设计	175 自8 朱洁霖	1.44E+17	是	2020/03/17 08:29:31	01分钟	否	0分钟
51	徐荣华	程序设计	177 8班 朱映丞3119001	1.44E+17	是	2020/03/17 08:29:41	01分钟	否	0分钟
67	徐荣华	程序设计	178 八班 邹耀鸿	1.44E+17	是	2020/03/17 08:31:29	09分钟	否	0分钟
26	徐荣华	程序设计	179 8班 黄玮娟	1.44E+17	是	2020/03/17 08:29:41	01分钟	否	0分钟
58	徐荣华	程序设计	180 八班 赖文希	1.44E+17	是	2020/03/17 08:30:31	00分钟	否	0分钟
50	徐荣华	程序设计	1八班雷文通311900114	1.01E+09	是	2020/03/17 08:29:51	00分钟	否	0分钟
59	徐荣华	程序设计	1 蔡志浩 3119001136	1.44E+17	是	2020/03/17 08:30:39	00分钟	否	0分钟
80	徐荣华	程序设计	1 赖文希	3.65E+08	是	2020/03/17 08:45:08	05分钟	否	0分钟
11	徐荣华	程序设计	1 梁梓熙 3119001152	1.44E+17	是	2020/03/17 08:29:31	01分钟	否	0分钟
11	徐荣华	程序设计	1 梁梓熙 3119001152	1.44E+17	是	2020/03/17 08:29:31	01分钟	否	0分钟
11	徐荣华	程序设计	1 这个是多余的人	1.44E+17	是	2020/03/17 08:29:31	01分钟	否	0分钟

图 11 资源表格

```
D:\Python37\python.exe
WARNING *** OLE2 stream 'SSCS': expected size 35136, actual size 1024
OK!
Press any key to continue . . .
```

图 12 运行结果

广东工业大学考试学生签到表									
课程名称: 程序设计 课程编号: TMP0801 序号: 1									
学年学期: 2020 春季 任课教师: 徐家华 考核方式: 考试									
序号	学号	姓名	签到	备注	序号	学号	姓名	签到	备注
1	3119001093	陈洁佳	√	100分钟	31	3119001154	阮志	√	95分钟
2	3119001094	陈瑞岳	√	100分钟	32	3119001155	阮国豪	√	100分钟
3	3119001095	陈伟生	√	100分钟	33	3119001156	阮步云	√	100分钟
4	3119001096	陈文福	√	100分钟	34	3119001158	阮国瑞	√	100分钟
5	3119001097	陈世华	√	100分钟	35	3119001159	阮明周	√	98分钟
6	3119001098	陈洁	√	100分钟	36	3119001131	阮文良	√	100分钟
7	3119001099	陈力	√	98分钟	37	3119001132	阮国洋	√	100分钟
8	3119001100	陈梓俊	√	94分钟	38	3119001133	阮国瑞	√	95分钟
9	3119001101	陈志斌	√	99分钟	39	3119001135	阮国瑞	√	100分钟
10	3119001102	廖仕强	√	100分钟	40	3119001136	廖志浩	√	95分钟
11	3119001103	廖远昕	√	100分钟	41	3119001137	李富平	√	100分钟
12	3119001104	梁东柱	√	99分钟	42	3119001138	李耀楠	√	96分钟
13	3119001105	梁耀楠	√	100分钟	43	3119001139	阮国瑞	√	100分钟
14	3119001106	王嘉华	√	100分钟	44	3119001140	阮国瑞	√	95分钟
15	3119001107	阮志平	√	100分钟	45	3119001141	阮国瑞	√	100分钟
16	3119001108	梁嘉豪	√	100分钟	46	3119001142	阮志平	√	95分钟
17	3119001109	梁智豪	√	100分钟	47	3119001143	阮国瑞	√	95分钟
18	3119001110	阮志信	√	100分钟	48	3119001144	黄志豪	√	95分钟
19	3119001111	阮耀楠	√	100分钟	49	3119001145	黄志豪	√	100分钟
20	3119001112	阮耀	√	100分钟	50	3119001146	黄志豪	√	100分钟
21	3119001113	阮耀文	√	100分钟	51	3119001147	阮国瑞	√	100分钟
22	3119001114	梁国瑞	√	100分钟	52	3119001148	阮国瑞	√	100分钟
23	3119001115	阮国瑞	√	96分钟	53	3119001149	李国豪	√	100分钟
24	3119001117	梁国瑞	√	100分钟	54	3119001150	阮国瑞	√	100分钟
25	3119001118	王仙	√	94分钟	55	3119001151	阮国瑞	√	88分钟
26	3119001119	王耀楠	√	100分钟	56	3119001152	阮国瑞	√	100分钟
27	3119001120	阮国瑞	√	100分钟	57	3119001153	阮国瑞	√	100分钟
28	3119001121	李力安	√	100分钟	58	3119001154	阮国瑞	√	87分钟
29	3119001122	梁国瑞	√	100分钟	59	3119001155	阮国瑞	√	100分钟
30	3119001123	梁国瑞	√	100分钟	60	3119001156	阮国瑞	√	94分钟
考试日期:				监考人数:	监考人数:				

图 13 输出文件

### 3.3 性能评估

程序的内循环部分为搜索学生名字位置，假设班级有  $N$  位同学，每个签到表板块为  $M$ ，则单次搜寻运行时间的总成本约为：

$$\frac{N \bmod M}{4} \times \left( \frac{2N + 4N}{2} \right) \times N \sim N^2$$

且其他循环处理皆能在常数时间内完成，故能有较大的概率保证程序总体运行时间为平方级别。因数据不足的缘故，故无法进行有效的程序测试。

在内存方面，程序采用的是能够节省大量成本的“输入-输出”模型，仅仅维护的只是三份 Excel 表格与一个 Conversion 类数据结构(包含三个变量与一个储存布尔型变量的列表)，因此程序所占内存主要与用户提供的资源相关，对于程序自身内存消耗几乎可以忽略。

## 5 总结

虽然这个程序对于我来说并不是特别困难，但是里面仍然有着许多值得去思考，去学习的地方。我也在制作程序的这段时间里了解到了许多新的知识，例如 Python 对于 Excel 表格的转换，以及如何才能去更好地进行代码重构，包装代码以及设计好这份小小的程序。此外，设计的过程中不同方法之间的对比和对程序时间、空间性能的追求也巩固了我所学的数据结构与算法。当我看到程序能够顺利地按照我的预期去运行的时候，一种喜悦感油然而生。

因设计经验尚缺，故该程序依旧可能会出现一些不可预知的情况，请多多包涵。

## 附录

**1. Sign-inTableConversion 签到表转换程序**

**2. Sign-inTableConversion2.0 签到表转换程序**