



# 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](#_Toc5502)

[学生签到转换软件 1](#_Toc18544)

[摘 要 1](#_Toc2411)

[Abstract 1](#_Toc26998)

[1功能设计 3](#_Toc3856)

[1.1主要设计目的 3](#_Toc254)

[1.2辅助功能 3](#_Toc10036)

[2框架设计 3](#_Toc5127)

[2.1总体设计思路及分析 3](#_Toc29916)

[2.2程序封装与拓展 4](#_Toc19295)

[3模块设计及实现 4](#_Toc20478)

[3.1模块设计与思路 4](#_Toc4125)

[3.2模块实现 6](#_Toc29308)

[4调试及结果 8](#_Toc32023)

[3.1调试及改进 8](#_Toc14929)

[3.2运行结果 10](#_Toc21744)

[3.3 性能评估 11](#_Toc27057)

[5总结 12](#_Toc22427)

[附录 13](#_Toc2570)

[1. Sign-inTableConversion 签到表转换程序 13](#_Toc10476)

[2. Sign-inTableConversion2.0 签到表转换程序 13](#_Toc27047)

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



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。



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，易得知在排序上次序与备注名易读性成正相关关系，故程序会在每个源文件中挑选出排序后前若干名学生的备注名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。

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:



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，当遇到学生名字重复或者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。

在程序完成后，会将源文件的名字复制到输出文件上，并且删除源文件以方便下次转换时直接使用。

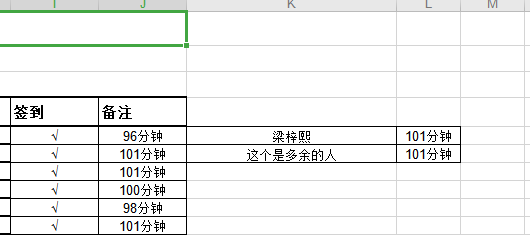


图 2 无法识别或重复的学生

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。

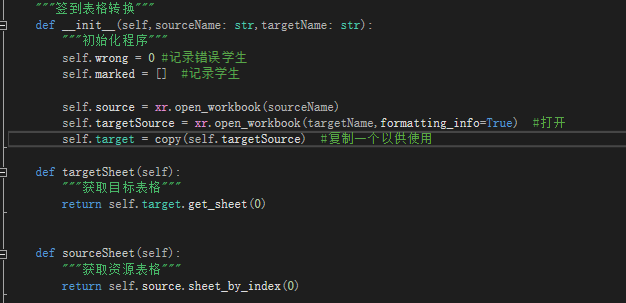


图 3 初始化及调用类方法

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，程序会用循环搜索到其中前4名学生的信息并进行匹配，若找到对应的模板文件则直接返回模板文件的名字；若未命中，则会返回第一个模板文件的名字。

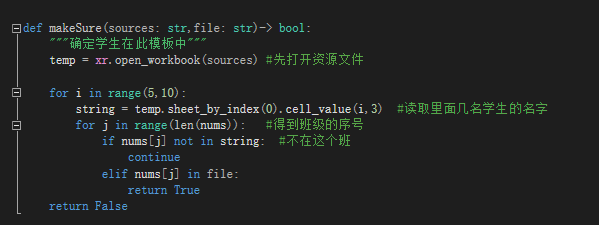


图 4 模板匹配函数

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。

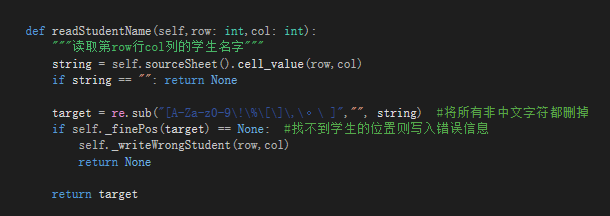


图 5 读取姓名的类方法

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。

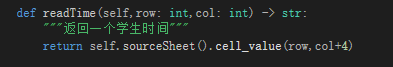


图 63 得到学生的在线时间

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。

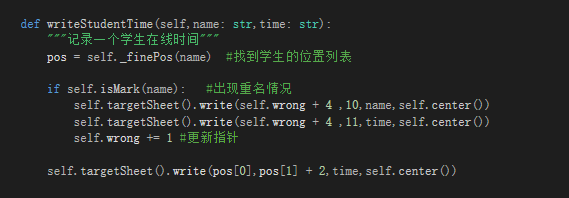


图 7 记录学生时间的类方法

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。

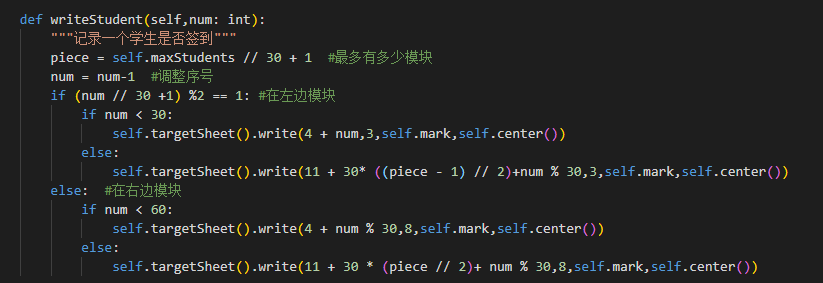


图 8 在常数级别下的信息录入

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。

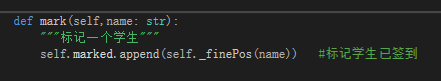


图 9 标记类方法(\_finePos为寻找学生在表格位置)

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。

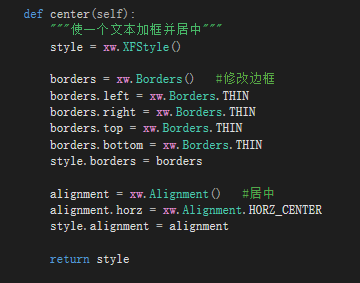


图 10 美化方法，返回一个风格变量

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。

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。



图 11 资源表格

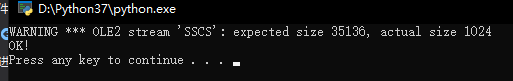


图 12 运行结果



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：



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。

在理论计算下，处理一份具有 100,000,000 份学生名单所时程序本身需要的内存为：



即只需要约92.16MB内存即可。

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