一：代码中已有功能简述

1.读取配置文件信息（获取教师进入队列时的时间对比参数，作为提出教师考勤时的依据，只有在配置文件中存在的班级才可以进行签到。）

2.时间计时器的建立（建立时间计时器，此功能在框架代码中已经实现，可在代码中研究学习）

3.模拟主菜单的界面（主菜单调用各种函数实现功能）

附：代码中时间窗口的实现方法：在第一个教师进入队列时，开启一个线程开始倒计时，时间是45×2+10分钟。

假设：之间没有错误。就是第一个大时间内没问题，直到时间结束，队首的教师被提出队列，此时计时器被销毁，开始下一个计时器

即：队首的教师和第二个教师的时间差值，重置一个计时器。之后，重复此类操作。

二：现有代码的结构

读取配置文件信息(\_init\_.py)

1. 类readini()
2. def \_\_init\_\_(self):
3. def readIni(self)：读取配置文件中的时间区间信息，教师进入队列时在配置文件中更新信息开始考勤，退出队列时删除信息且教师名下的班级同时无法进行考勤。

时间窗口(Timewindows.py)：

1.类 Timer():调用时间窗口模块的时间是班级正确进入考勤列表。第一节课的时候第一个老师考勤的时候，直接计时，计时结束后，将考勤的班，踢出列表；第二节课，另外一个教师再次考勤，会销毁上一个时间窗口，然后重新计时。也就是说时间窗口只是记录考勤列表里的第一个班级的结束考勤时间，当考勤列表的第一个班被踢出列表后，计时器重新计时，记录新的列表首的班级。

2.def timeCheck(self): 独立的踢出队列函数，即列表为空的话直接提出队列。

3.def startCheckTime(self): 教师被踢出队列时需要重新计时,或者自动到时间后教师被提出队列，此处只计算与下一位的差值，并重新计时。

4.def stopCheckIn(self): 通过计算将已经达到时间限制的教师踢出队列。

run函数(run.py)

1. 类SystemRun(object):

2.def \_\_init\_\_(self)

3.def Form(self)：存储程序的模拟主菜单，其中包含有各个功能模块。

4.def dform(self)：查看信息窗口衍生的子菜单。

5.def cform(self)：教师的手工考勤功能子菜单。

6.if \_\_name\_\_ =="\_\_main\_\_" :run文件中存放着菜单和各个类的方法。

三：文档与代码的关系

1. setting.ini文件：ini文件中存放课程的上课和下课时间，python文件中的readini的类中读取配置文件中的时间区间，作为系统时间规则的标准。大部分代码都会用到这个时间，用正则表达式来读取文件中有用的信息，比如说进入考勤队列的时候，需要判断进入考勤队列的时间是否为下课时间。
2. course.csv文件：文件中放置课程ID、课程名称、教师ID、班级名称(班级名称格式当前未处理)。
3. courseInfo.csv文件：文件中保存课程处理信息，将班级信息处理后存入，teamanualcheckin.py中的def manualCheckInAdd(self,teacherWechatID)教师进行手工考勤的时候读取文件信息通过教师输入课程号找到课程信息，生成新的seq表新的行列，并将新的考勤详细信息表生成，将班级的全部信息写入列表中。
4. studentInfo.csv: stucommonmethod中checkStudent(self,\_stuWechatID)学生身份信息验证时使用。
5. teacherInfo.csv: teacommonmethod中checkTeacher(self,\_teaWechatID)教师身份信息验证时使用。
6. seq.csv:教师每次考勤就增加一次考勤次序表的记录，手工考勤时可手动增加考勤记录，但抽查时不生成新的seq表，只在detail表中追加即可。
7. example.csv: 示例文件未使用。

**需要大家去做的事情**

0：浏览已有的文件

1：根据源代码的注释提示将各个文件的方法补充完整。

2：将导入文件模块加入到框架之中。

3：需要完成的模块有：

教师与学生的普适方法，即基本的信息身份验证。

教师开启自助考勤与学生进行自助考勤。（具体表现为文件信息的创建、读出、写入、修改等功能，需要注意功能在程序中的具体实现顺序来对表格进行操作）

教师手工考勤（先前代码中的实现思路是：教师通过身份验证后找到所授课的班级课程名称，并创建seq与detail表，将全部学生的信息加入，并默认未出勤状态。然后调用教师修改考勤状态的功能对于未出勤的同学的考勤结果进行修改）

4：删掉提示性的注释

5：遇到问题百度或咨询同学

**可能遇到的问题：**

1. cmd菜单的维护方法：

由于利用pycharm开发环境时，代码编码注释为utf-8，故在cmd中直接运行程序会导致中文乱码。故需要在命令行中输入：chcp65001 将默认编码格式转换为utf-8后，运行菜单，可以正常显示代码中的中文菜单。（若想要将命令行回归gbk编码需要输入：chcp936）

1. 如何添加新的模块到菜单中 ：如 import csv 模块

pycharm的开发环境中，如果需要使用新的模块，就需要将新的模块在代码的始端表明。

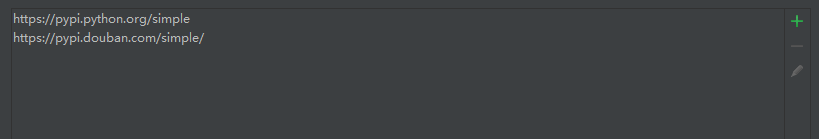
用import语句导入模块，当使用在被导入模块中定义的函数时，必须包含模块的名字。

如果不小心忘记了导入，那么在你使用模块中的方法的时候，pycharm会弹出一个窗口来提醒你导入新的模块，按下Alt+Enter，采取快捷菜单中的建议，此时import命令会被添加到导入模块的代码部分，并且输入光标仍留在原位，方便我们继续输入并且无需重定位。

1. 如何快速的在pycharm中加入新的第三方模块（以csvkit为例）：

打开pycharm，依次点击file--->settings--->project--->project interpreter--->在右边黑框中随便选择一个包进行双击--->在上部搜索框中输入你想要添加第三方包的具体名称（csvkit）--->选中你所要添加的包并点击install package进行安装。

当进行搜索包的时候，可以点击Manage Repositories进行源的更改，使用豆瓣源会大大加快下载速度。（即如下图）



附加：如希望命令行pip安装模块，百度由很多的相应教程，可自行搜索学习。

# 

# 附加信息：

json模块：JSON(JavaScript Object Notation) 是一种轻量级的数据交换格式，易于人阅读和编写。

|  |  |
| --- | --- |
| json.dumps | 将 Python 对象编码成 JSON 字符串 |
| json.loads | 将已编码的 JSON 字符串解码为 Python 对象 |

sys模块：sys是system的缩写，用来获取[操作系统](https://www.baidu.com/s?wd=%E6%93%8D%E4%BD%9C%E7%B3%BB%E7%BB%9F&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1dhmHnLPvD3nyP9ryuWnWm0IAYqnWm3PW64rj0d0AP8IA3qPjfsn1bkrjKxmLKz0ZNzUjdCIZwsrBtEXh9GuA7EQhF9pywdQhPEUiqkIyN1IA-EUBtkn16vnWmLnH6zPHmdPj0Ln10s" \t "https://zhidao.baidu.com/question/_blank)和编译器的一些配置，设置及操作。

如判断文件和文件夹是否存在，创建文件文件夹，获取系统版本之类的操作,sys模块获得脚本的参数、处理模块、使用sys模块操作模块搜索路径、使用sys模块查找内建模块、使用sys模块查找已导入的模块等作用。

os模块：Python os模块包含普遍的操作系统功能，如果你希望你的程序能够与平台无关的话，这个模块是尤为重要的。

csv模块：csv文件格式是一种通用的电子表格和数据库导入导出格式。

time模块中常用函数：

time.localtime([secs])：将一个时间戳转换为当前时区的struct\_time。secs参数未提供，则以当前时间为准。

ConfigParser模块:

1.1.读取配置文件

-read(filename) 直接读取ini文件内容  
-sections() 得到所有的section，并以列表的形式返回  
-options(section) 得到该section的所有option  
-items(section) 得到该section的所有键值对  
-get(section,option) 得到section中option的值，返回为string类型  
-getint(section,option) 得到section中option的值，返回为int类型

1.2.写入配置文件

-add\_section(section) 添加一个新的section  
-set( section, option, value) 对section中的option进行设置  
需要调用write将内容写入配置文件。

# 