**实验报告**

本科生课程设计(实践)任务书、设计报告

**（Python程序设计基础课程设计）**

计算机基础教学实验中心

2022年5月30日

**运动会积分管理系统**

**摘要** 通过编写运动会积分管理系统，巩固对Python语言(包括tk库)的理解，提升程序设计、资料查询、报告写作等综合能力。最终版的系统完成了要求的所有功能，并带有用户界面，完成度较高。

**关键词** Python，数据结构，字典，图形用户界面(GUI)

# 开发设计思想

## 函数模块化、命名规范化

由于整个系统较为庞大，且要求功能1~8即开即用、独立运行，为了方便重复调用以及后期修改维护，程序整体采用模块化函数编写。没有缩进的主程序只包含一个语句，即绘制开始菜单函数的入口。每个函数有独立的功能，可以随时独立运行。这样，即使用户不希望运行某个模块（或者某个模块报错），其他功能也能正常使用。

因此，这就要求在开发时，所有函数、变量的命名都有严格标准，能做到“望文生义”；所有变量必须做到全局、局部严格区分。

变量用英文名词命名。函数命名将在后文介绍。

## 存储安全化

为保证数据安全和程序的可移植性，本系统中，每个局部变量的修改都会及时同步到全局变量，每个全局变量的修改都会及时写入文件，然后再进行下一步操作。这样会牺牲运行时间，但是能确保数据的安全性（用户随时终止程序都不会丢失数据），而且经实测，这样的牺牲时间并不会带来肉眼可见的卡顿。

3个表格分3个文件存储，均采用CSV格式、GBK编码。CSV格式的好处是，程序读取CSV时，可以简单地通过换行符、英文逗号进行字符串分割，非常方便；而Windows默认采用Excel打开CSV，这样一来，用户不需要通过本系统也能直观地查看、修改源数据，进而通过截图、云文档等方式备份数据，进一步提升数据安全性。

## 数据结构化

全局变量主要指sports, faculties, stus三个列表。下面列出数据存储结构：

**列表 字典 字符串**

图示

描述已自动生成

图示

描述已自动生成

## 设计现代化

为获取较高分数，我为系统写了用户界面（GUI）。但是原生TK库包含的组件外观较为远古、且不能适配不同电脑，因此程序采用多种方式实现现代化外观（详见后文“GUI现代化问题”）。GUI在整个程序设计中占了一定分量。

## 时间合理化

为合理分配时间，本系统按以下时间线编写：

## 其他

系统环境：Windows 10+Python 3.10与Windows 7+Python 3.5环境下测试正常。

# 功能详细设计

## 主菜单、主功能概述

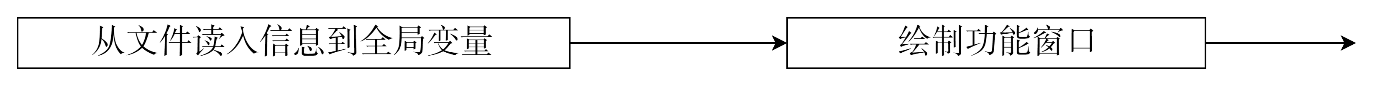
运行即进入主菜单。主菜单仅包含9个按钮（8个主功能+数据初始化），按钮绑定主功能函数。为避免窗口过多造成的不良用户体验，进入主功能后，主菜单隐藏。关闭主功能窗口后，主菜单才再次显示。

主功能函数以“fileop数字(功能主题色)”命名，如fileop2("#5E35B1")。

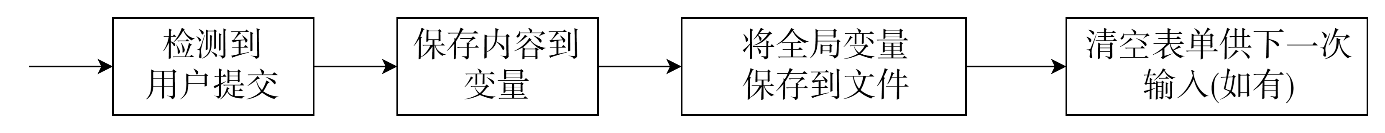
## 通用模块函数

### 初始化与保存

每次修改都会涉及文件操作。下列所有流程图中，BEGIN指代以下功能：



END指代以下功能：



### 绘制表格

8个主功能中，有5个功能需要用到表格，因此定义new\_table()函数，统一绘制所有表格。传入canvas地址、表头、数据等信息到new\_table()函数，它就会调用tk库中的ttk模块绘制Treeview表格。这样做的好处是，表格样式基本统一，且相同代码不重复出现，提高程序简洁性和可读性。

## 增添、读取功能

为符合课设要求，功能1、2分别是增加、查询功能。以功能1为例：



## 修改、删除功能

功能3的示意图如下：

图标

低可信度描述已自动生成

## 查询功能

功能4~7为查询功能，均遵循以下过程：



### 绘制输入表单

部分功能的“绘制输入表单”包含下拉列表，需要借助其他函数实现，比如功能6的GUI代码中，就调用了array\_sports()函数，以获取包含所有运动会项目的列表。

本程序中，所有以array\_开头的函数，需要传入模式参数，返回列表，供绘制GUI使用。

### 获取

“获取”模块是每个功能都不一样的，这个模块函数被嵌套在功能函数里面，一般命名为search()。有的search()函数比较复杂，那么它会嵌套handle()子模块。这些模块函数的实现主要依赖“信息转换”——“遍历、记录”的过程实现。

“信息转换”依赖get\_函数。这些以get\_开头的函数，大多是全局函数，函数的定义即表明其作用。比如，get\_sport\_name(sport\_id)函数表示传入项目编号，获取项目名称；get\_falcuty\_fname(fid)表示传入学院编号，获取学院名称；其他依此类推。单独定义这些函数，有利于使主功能函数更加精简，同时方便不同的模块重复调用。

这些get\_函数，大都遵循以下方式执行：



get\_函数返回数值或字符串。

“遍历、记录”也是遍历并记录符合条件的值，不过一般以列表方式存储，方便输出到GUI表格。

# 问题与解决方案

## 文件空行问题

前文提到，文件存储格式为CSV，并允许用户修改。由此，文件末尾的空行数量可能为0或多行，在此基础上，直接使用a+追加信息，会带来致命错误。file\_append()函数（第47行左右）应运而生。程序的所有a+追加操作均调用file\_append()函数。该函数会通过正则表达式先去除文件末尾多余的空行（没有空行的补一个），再追加内容，使得在追加信息前，文件末尾空行始终有且仅有一行。具体流程如下：

QR 代码

描述已自动生成

## 程序稳定性问题

上课曾提到程序鲁棒性问题。鲁棒性即稳定性，指程序在异常和危险情况下的应对能力。在程序编写完成后，程序在实际使用测试中表现不佳，比如无法应对空值输入、重复编号、无初始化文件会报错等问题。之后，程序增加了多个verification\_\*()函数，一一解决以上问题。目前，程序已经可以正确识别并警告不合法的输入，并在部分节点询问用户对文件的操作方式。

## GUI现代化问题

使用原始tk库编写的界面，充斥着20年前Windows XP那种远古、沧桑、老旧的历史感。作为新时代的大学生，我不能接受。另外，在不同分辨率、不同大小的屏幕下，原始tk库也有些许排版问题。为了使最终程序实现现代化，我做了以下努力：

### 使用tk.ttk而不是tk

查阅文献得知，在Python 3中，Tkinter 库新增了ttk组件。TTK中包含了和TK组件有相同功能的组件，如Entry，RadioButton，OptionMenu等，但是TTK中的组件调用了原生Windows的外观属性，使得整个程序与Windows一体化。

此外，Treeview表格是ttk中独有的控件。因此绘制表格也用到了ttk。

### 去除浮雕样式

由于某些原因，ttk中的Button组件存在bug。这是我翻遍国内外论坛帖子，最终在Stack Overflow得到的结论。因此只能使用原生tk库中的按钮组件。

造成原生tk库丑陋的终极原因，是它采用了浮雕设计。在Button属性中，增加relief= "flat"可以使浮雕样式变为扁平，初具现代程序的模样。

### 固定化所有元素

TK中提供了pack()和grid()函数。这两个函数可以根据窗口大小自动排列控件元素，但是效果不尽人意。例如，在窗口最大化时，元素间距过大；窗口缩小至一定比例时，元素有可能重叠。因此，我拒绝使用它们，转而采用place()函数。

Place()函数需要指定元素的四要素（左上角x、y，长，宽，部分缺省亦可），才能绘制元素。因此，程序的大部分窗口都锁死了窗口大小（亦不允许最大化）。

为了让界面更加好看，我先使用绘图软件画出各个界面的所有元素，再把他们一一转成place语句参数。尽管这样工作量庞大，但是在绘图软件里面，元素的排列对齐有更多样化的选择，元素位置的修改也更方便随意。绘图软件为美化界面提供了更多的可能性。至今，我的项目工程文件夹内，还存有9个界面的原始设计稿。

### 其他

为了应对不同分辨率的显示器，查阅Python官方文档发现，把GUI字体大小改为负值，可以把计量单位从“像素”改为“points”，从而解决此问题。

为了增加了程序和Windows的亲和度，程序中所有的提示信息，包括信息类、警告类、错误类和询问类信息，都通过Windows原生对话框弹出。

为了方便用户区分不同的功能，各功能区都有独立的主题颜色，这些颜色贯穿在从主菜单到功能界面的每个角落。颜色选取了Google推出的Material Design调色板中的色值，整体繁而不乱，多而不杂。

# 课设总结

通过本次课设，我学会了使用Python写一个多模块的系统程序，经历了程序从框架搭建到界面绘制，再到细节优化的全周期，提升了代码能力和文献查阅能力。

此外，课设过程中，通过与同学的交流，我了解了Python语言的易错点，并提升了沟通表达能力。