

基于 Python 的砝码检定证书自动生成系统

□中国计量科学研究院 陈杭杭 钟瑞麟

【摘要】针对砝码检定中需手动进行大量数据处理和证书编辑等问题，基于Python编程，提出了一种面向Word的砝码检定证书自动生成方法，实现了包括检定数据自动修约、结果评判、证书模板识别等功能在内的证书自动生成系统。

【关键词】Python；砝码；检定证书；自动化

文献标识码：B 文章编号：1003-1870（2023）03-0013-4

引言

砝码检定证书和检定结果通知书是证明砝码计量检定合格的法定技术文件，由法定计量技术机构具有资质的检定员根据实际检定结果予以出具。由于检定过程中，数据处理、结果评判、证书录入和编辑等往往依赖人工，过程繁琐且容易出错，制约了检定工作效率。为此，考虑使用计算机代替人工进行检定中的数据处理、合格判定、数据录入和证书编辑，从而提高工作效率，降低出错风险。

作为近年快速兴起的高级编程语言，Python语法简单，功能强大，除了可以快速搭建用户操作面板、采集和处理数据外，还可以编辑和处理各类Microsoft Office文件，实现自动办公。此外，Python还具有良好的可移植性，经编译后可在Windows，Mac和Linux等各类操作系统下直接运行，非常方便。

本文以砝码检定为例，基于Python的PySimpleGUI，docx，time和decimal等模块，搭建了简洁实用的用户操作界面，实现了Word格式砝码检定证书的自动生成。

1 砝码检定证书自动生成原理

证书的录入和编辑是检定工作的最后一步。以中国计量科学研究院质量室为例，检定员需要在判定砝码合格后，下载Word格式的检定证书模板，根据化整后的砝码质量修正值计算折算质量实际值，将其与砝码编号、允许误差等录入模板数据页（第三页），将砝码规格型号、送检单位、生产厂家、检定结论、检定时间、有效期等录入模板封面（第二页），将标准器信息、检定地点和实验室温湿度等录入模板第二页。如果检定不合格，则需要下载检定结果通知书模板，再将上述信息录入。

将上述过程简化为流程图，如图1所示。其中，获取检定周期、获得砝码允许误差和有效数字位数、对修正值进行修约并判定是否超差以及生成检定证书或检定结果通知书都可以通过Python程序自动完成。由于本程序针对的是手写原始记录，砝码的修正值需要手动填写，但不必对修正值进行修约。如果采用Excel格式的原始记录，砝码修正值也可以通过Openpyxl模块自动读取，效率更高。

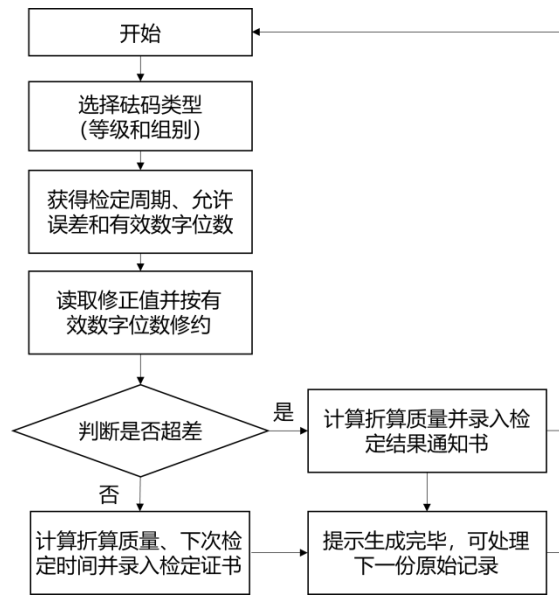


图1 程序流程图

程序设计包含以下要点：

- (1) 计算不能因浮点运算引入误差；
- (2) 数据修约需依据国家标准和/或实验室作业指导书；
- (3) 程序设计应尽量遍历各类错误，防止用户界

面闪退或程序卡死。

2 基于Python 编程的砝码证书自动生成实例

本程序针对手写原始记录，利用PySimpleGUI 模块设计了用户操作界面，如图2 所示。

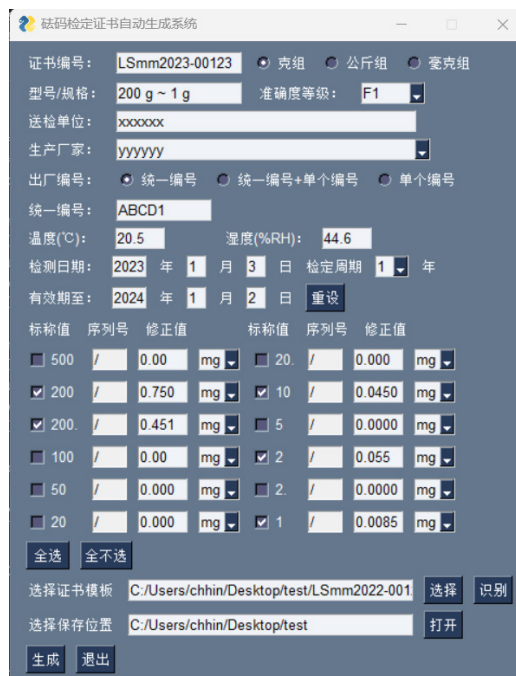


图2 用户操作界面

其中证书编号、型号/规格、送检单位、生产厂家、准确度等级和统一编号等信息可以直接从Word格式的证书模板自动识别。检定员可在用户界面中检查和修改。代码如下：

```
import PySimpleGUI as sg
.....
if event in ('识别'):
    try:
        doc = Document(values['-Cer-'])
        window.Element('-SN-').update(doc.paragraphs[2].
text[5:])
        .....
    except:
        sg.popup('请选择正确的证书模板!')
```

其中'-Cer-'是模板存储路径对应的键，对应“选择证书模板”文本框中的地址。'-SN-'是证书编号对应的键。doc.paragraphs[2].text[5:]是模板中证书编号对应的字段。

一旦界面上的“生成”键被触发，程序将根据输入的砵码修正质量值自动计算折算质量实际值，并对二者进行修约。

为了规避浮点运算误差对高等级砵码带来的风险，程序将所有需要参加运算的数据都保存为字符串格式，并调用decimal模块实现了十进制数的准确运算。

其中，修约采用了GB/T 8170-2008推荐的规则。除“四舍六入五成双”外，当拟舍弃数字的最左一位数字是5，且其后有非0数字时，无论其前后是奇是偶，都向前进位。

以g组F₁砵码为例，实现代码如下：

```
from decimal import *
getcontext().rounding = 'ROUND_HALF_EVEN'
.....
i = 0
for Nom in nom_values:
    Nom_mass = Decimal(Nom.split('.')[0])# 如果标称值带点则只取点前的字符串
    if Nom_mass >= 500:
        corr.append(Decimal(values['-Nom-']).
quantize(Decimal('0.0')))
```

```
        cal.append(Nom_mass+Decimal(corr[i]/1000).
quantize(Decimal('0.0000')))
    else:
        corr.append(Decimal(values['-Nom-']).
quantize(Decimal('0.00')))
        cal.append(Nom_mass+Decimal(corr[i]/1000).
quantize(Decimal('0.00000')))
    i += 1
```

其中第二行代码通过配置decimal模块的参数，实现了符合GB/T 8170-2008的修约规则。为了保证数据的可靠性，也可在修约时采取较为保守的做法，即只要拟舍去的位数非0，就向前进一位。若采取这种修约规则，需要将decimal的rounding参数配置为“ROUND_UP”。

“corr”是保存砵码质量修正值的列表，“cal”是保存砵码折算质量实际质量值的列表，值都保存为Decimal型。

计算完成后，根据砵码类型和标称值查表，可得到对应的允许误差、绝对值|IMPEI|。如果砵码修正值的绝对值小于等于|IMPEI|，则读取砵码检定证书Word模板，将合格砵码信息以表格的形式录入。否则，读取砵码检定结果通知书模板将不合格砵码信息以表格形式录入。

以砵码检定合格的情况为例，利用docx模块，将检定结果生成表格并插入到文档规定位置的关键代码如下：

```
def move_table(table,para):
    tbl, p = table._tbl, para._p
    p.addnext(tbl)
    .....
    copy_tbl = Target_file.add_table(rows=1,cols=len(judge),style = 'Table Grid')
    .....
    for item in Content_list:
        row_cells = copy_tbl.add_row().cells
        i=0
        for num in judge:
            row_cells[i].text = item[num]
            .....
        i+=1
```

move_table(copy_tbl,Target_file.tables[3].cell(0,0).paragraphs[0])

其中第一个函数的作用是将传入的表格table插入到段落para之后，这主要是为了实现在表格中插入表格。“judge”是一个数字列表，用于判断砝码有无单独序列号。如果砝码有单独序列号，表格会多添加一列序列号信息。

实际程序中还有设置表格高度、对齐方式以及字体字号的语句，语法可在docx模块的说明文档中查询，此处不作赘述。

检定证书的封面和第二页信息也可利用docx模块Document对象实现自动插入。这一过程并不需要预先在模板中相应位置插入“锚点”，只需要分析模板的段落位置或表格的行列信息即可实现。在特殊情况下，为了不改变模板字体，还需要分析段落(paragraphs)中的字段(runs)信息，再插入相应内容。

在图2所示情况下，由本程序自动生成的砝码检定证书如图3所示。

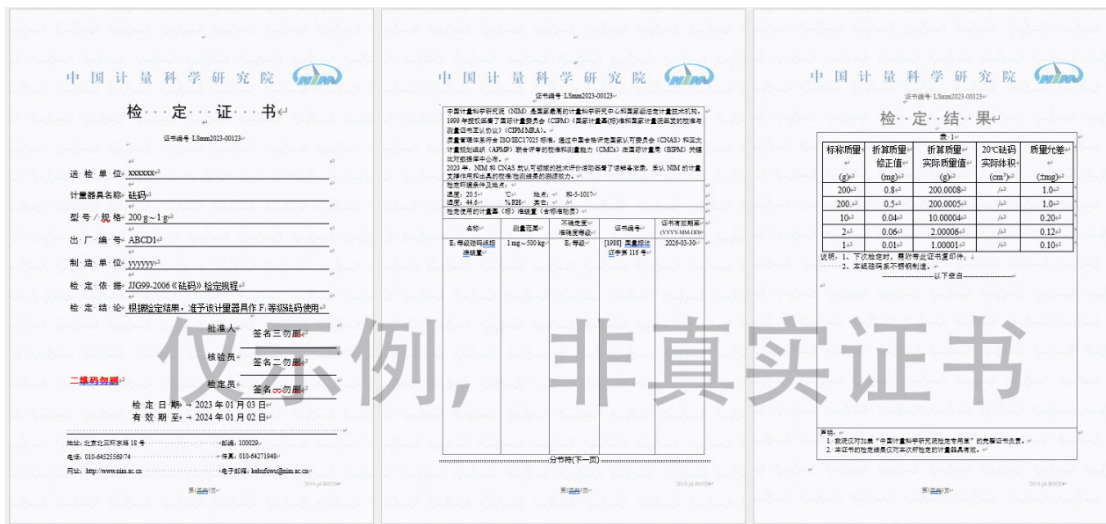


图3 自动生成的砝码检定证书示例

此外，“检定日期”“有效期至”两行是程序自动计算并录入模板封面的。程序利用time模块，根据砝码组别、准确度等级两个选项自动判断检定周期，再根据检定日期计算得到检定证书的有效日期。本程序的检定日期默认为用户计算机当前时间的日期，也可手动在用户界面进行更改。

为了提高程序的可靠性，防止证书系统因为用户误操作而出现卡死或闪退的情况，本程序还尽可能分析了所有可能出现的错误类型，如：修正值不能转化为Decimal型（如为英文字母、汉字或其他符号）、准确度等级输入有误（如不在“E₁”~“M₂”范围内）、检定日期或有效期填写有误、未选择证书模板、未选择保存位置、识别了错误的检定证

书或检定结果通知书模板路径等，并在编程时一一给出了出现这些错误时程序的应对方式——弹出对话框，用中文提示错误类型，返回继续识别操作界面更新。这一做法不仅提高了程序后期使用的可靠性，在程序本身的调试过程中也发挥了关键作用。

最后，为了在不同终端使用这一程序，还应将编写完成的Python脚本编译为可执行文件。以windows系统为例，使用pyinstaller模块，从Python脚本所在的目录进入MSDOS环境，并执行命令pyinstaller -F -wFormatrans.py即可。

经测试，生成的可执行文件Formatrans.exe可在windows 7及以上版本系统运行。这是因为windows XP及以下版本的windows操作系统不支持docx模块。

3 结论

论文充分利用了Python强大的功能模块和面向对象编程能力,用简洁的代码实现了数据处理、结果判定和检定证书自动生成,在一定程度上提高了砝码检定工作效率,避免了人为因素造成的错误。论文提供的相关方法也可以推广到其他计量检定领域。

鉴于实际工作中还涉及砝码标称值可能不为 $x \cdot 10^k$ (x 为5、2或1, k 为整数)的情况以及需计算和报告不确定度的情况,程序将为此进行改进。

参考文献

[1] 刘中文,基于VBA编程的砝码检定证书自动生成方法[J]. 衡器,2021,50(11).

[2] 陈玉忠,于振凡,冯士雍等. GB/T 8170-2008 数值修约规则与极限数值的表示和判定[S].

[3] 叶德培,赵峰,施昌彦等. JJF1059.1-2012 测量不确定度评定与表示[S].

[4] R 111-2004. Weights of classes E1,E2,F1,F2, M1, M1-2, M2, M2-3 and M3[S]. OIML.

作者简介

陈杭杭,硕士,中国计量科学研究院力声所质量室工程师。

钟瑞麟,博士,中国计量科学研究院力声所质量室副研究员。