信息安全课程设计报告

题 目： 防伪码设计与实现

专 业： 18智能科学与技术

姓 名： 潘俊安

学 号： 2018327100146

指导教师： 肖荣

2021年 6 月 29 日

目 录

[1、前言 1](#_Toc75885820)

[2、系统分析 1](#_Toc75885821)

[2.1、基本要求 1](#_Toc75885822)

[2.2、设计内容 1](#_Toc75885823)

[3、功能设计 2](#_Toc75885824)

[3.1功能介绍 2](#_Toc75885825)

[3.2项目流程图 2](#_Toc75885826)

[4、关键技术 4](#_Toc75885827)

[4.1 主界面及各个子页面 4](#_Toc75885828)

[4.2生成数字、混合防伪编码 4](#_Toc75885829)

[4.3 生成含数据分析功能的防伪码 5](#_Toc75885834)

[4.4 智能批量生成防伪码 5](#_Toc75885838)

[4.5 后续补加生成防伪码 5](#_Toc75885839)

[4.6 条形码与二维码 6](#_Toc75885844)

[4.7 粉丝抽奖 6](#_Toc75885849)

[4.8 项目打包 6](#_Toc75885852)

[5、测试 6](#_Toc75885854)

[6、结论 14](#_Toc75885855)

[6.1完成情况 14](#_Toc75885856)

[6.2心得体会 14](#_Toc75885857)

[参考文献 15](#_Toc75885859)

# 1、前言

随着改革开放的深入发展，我国从一个发展滞后的国家逐渐成为一个生产制造业大国，各行各业生产的产品遍布全世界，并涌现了华为、格力、海尔等一批世界级的制造企业。但是，一些不法企业为了获取不义之财，假冒正规企业的产品进行仿造生产，这给企业和消费者都带来了巨大的损失。如何保证用户买到正规的产品和企业健康地发展，除了政府加强监管和惩罚外，使用防伪码对产品进行产品验证已成为一种比较实用的方式。同时随着互联网、物联网的快速发展，企业生产的产品不但需要高品质，还需要追踪用户需求，吸引用户，为用户提供有价值的服务。通过企业产品的编码（防伪码、条形码、二维码)标识验证系统可以很好地实现用户和企业的信息对接。防伪码的生成与研究也是一门与信息安全技术紧密联系的技术，本次课程设计，也想简单编写一个企业防伪码系统。 Python丰富的库系统使得GUI界面设计和一些特定的功能实现变得十分地方便，却又不会丧失研究过程中学习的乐趣与对内在技术的探究。通过对于这一项目的实现，无疑会收获颇丰，对于今后的学习和开发也是十分有帮助的。

# 2、系统分析

2.1、基本要求

1. 具有不可逆性。

2. 具有不可预测性。即不容易有意构造一个码属于防伪认证码系列。

3. 抵抗生日攻击，具有稀疏的特征。

4. 防止碰撞发生，即同样的防伪认证码出现。

5. 在保证安全性的同时具有实用性，包括码长合理，不能太长。

6. 防止根据已有的防伪认证码来预测，或者推测新的有效防伪认证码。

2.2、设计内容

该项目主要的目的是实现基于Python的防伪码生成系统：自行设计一款基于Qt的GUI界面设计的防伪码生成系统，实现防伪码、二维码和条形码生成和保存。主要包含三个功能模块：防伪码生成功能、条形码和二维码生成功能、粉丝抽奖功能。三个功能在一个GUI界面实现，不同功能展示各自不同的GUI子界面。

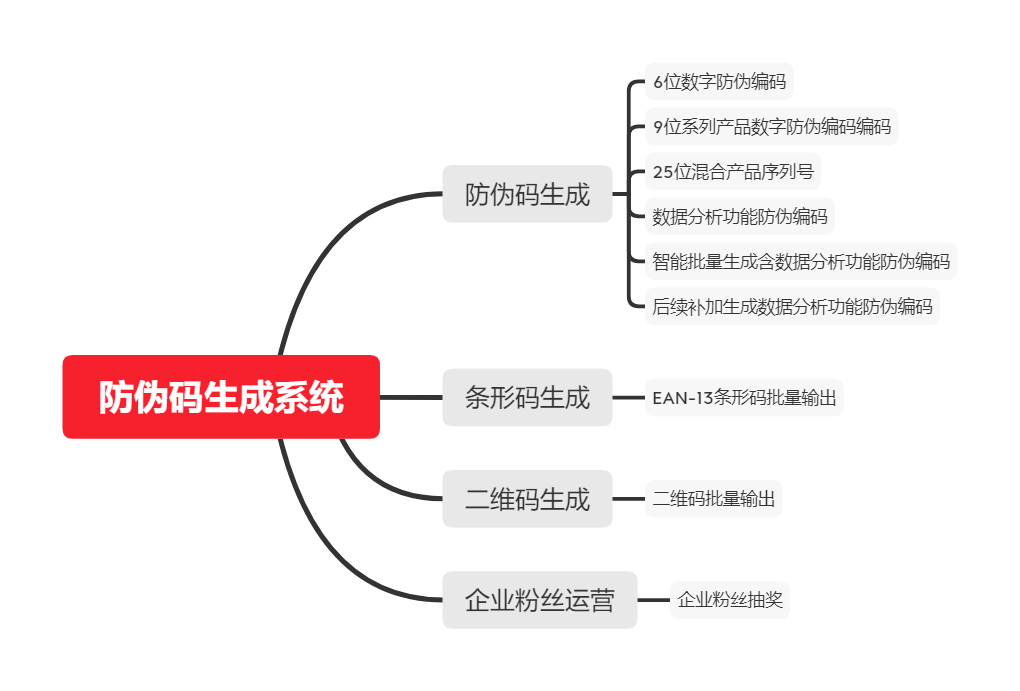
# 3、功能设计

3.1功能介绍

该项目的主要功能就是生成企业不同要求和标准的6位、9位、24位、12位可数据分析的防伪码，按照企业配置文件智能批量生成防伪码，后续补充已生成的防伪码、条形码和二维码生成、企业粉丝抽奖。

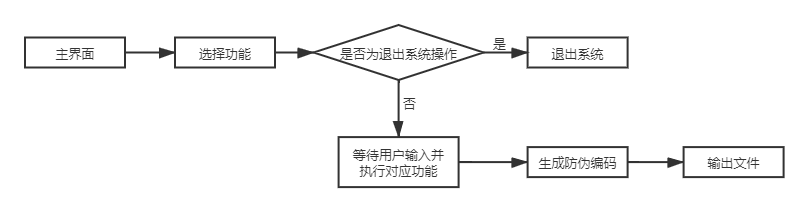
3.2项目流程图

下面是项目具体的流程图，从中也可以体现出系统的各个功能：

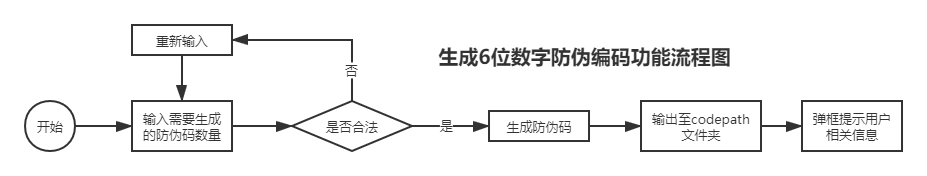


**图3-1 项目流程图**

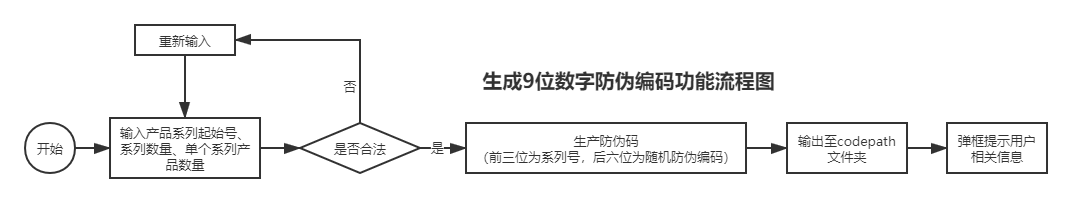
④功能流程图



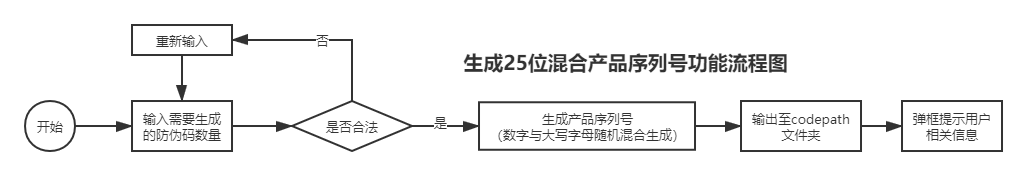
**图 3-2 总体功能流程图**



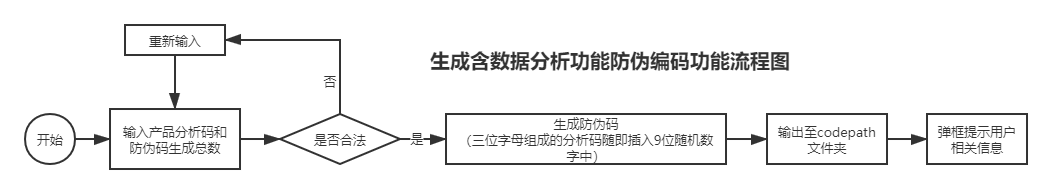
**图 3-3 生成6位数字防伪编码功能流程图**



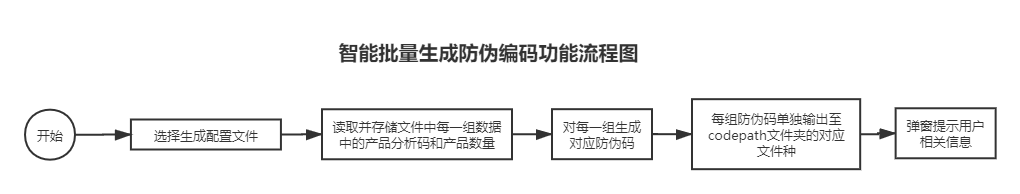
**图 3-4 生成9位数字防伪编码功能流程图**



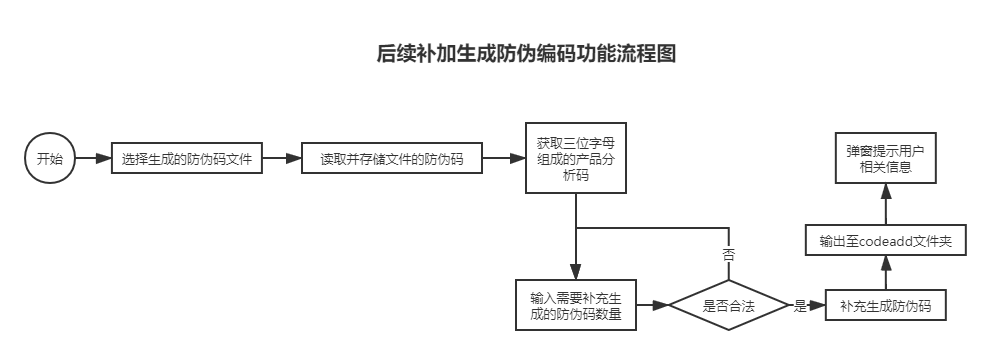
**图 3-5 生成29位混合产品序列号功能流程图**



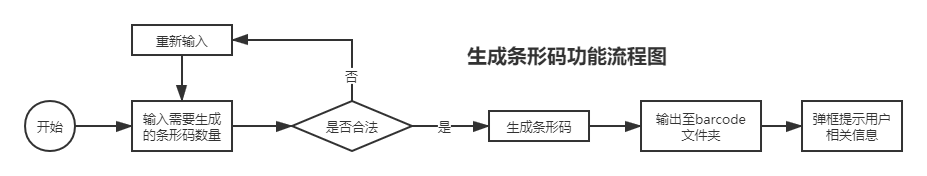
**图 3-6 生成含数据分析功能防伪编码功能流程图**



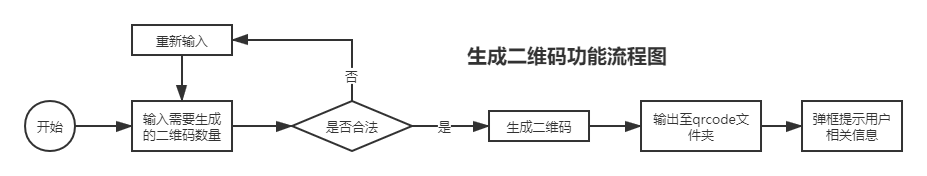
**图 3-7 智能批量生成防伪编码功能流程图**



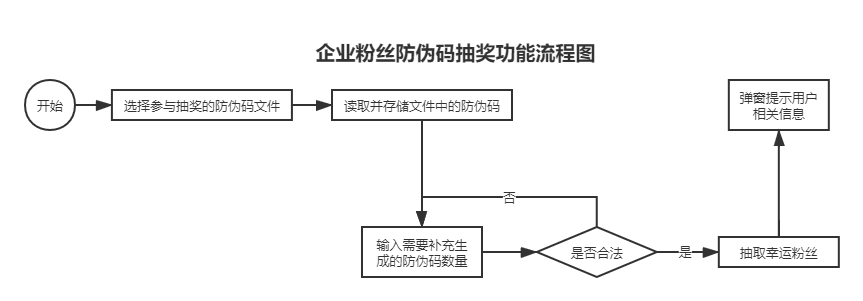
**图 3-8 后续补加生成防伪编码功能流程图**



**图 3-9 生成条形码功能流程图**



**图 3-10 生成条形码功能流程图**



**图 3-11 企业粉丝防伪码抽奖功能流程图**

# 4、关键技术

4.1 主界面及各个子页面

4.1.1．在Qt designer软件中新建一个widget文件用于设计各个子页面；

4.1.2． 在主页面的menubar中加入Qmenu对象和QAction对象，用于用户选择系统功能。对于没有子菜单的，仅用于跳转的Qmenu对象可以在py文件中修改对象类型来实现QAction对象的功能；

4.1.3． 设计完成后将界面文件保存为ui文件，在PyCharm软件中使用预先配置好的PyUIC工具，将ui文件转换为py文件，界面便成为python中的一个类；

4.2生成数字、混合防伪编码

4.2.1． 首先检测用户输入的相关信息是否有误，如防伪码数量是否为0、系列起始编号是否为3位数字、系列数量等；如果信息存在错误，使用Qt中QtWidgets类下的QMessageBox对象的warning提示框提示用户错误信息和错误原因；

4.2.2． 对于6位数字和29位混合防伪码，分别使用random库中的randrange函数或者choice函数，从各自防伪码每位可能出现的字符中随机抽取一位，通过遍历生成对应位数的防伪码；例如6位数字防伪码的每一位都是从0-9的数字中抽取的，29位混合防伪码每一位可能是数字也可能是大写字母，所以应该从“ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890”中随机抽取。

4.2.3． 对于9位数字编码，需要用户输入产品系列起始产品号、系列数量和单个系列产品数量，需要检验用户输入的每一个数据是否合法；生成9位防伪码的本质与6位防伪码相同，只需要生成6位防伪码后在防伪码前加上3位系列号；

4.2.4． 完成防伪码的生成后，首先使用os库的path.exists方法检验程序同一目录下是否存在用于输出防伪码的“codepath”文件夹，如果不存在，使用os的mkdir方法建立codepath文件夹；检测输出文件夹是否存在完成后，是由open方法以“写”模式打开对应的输出文件（txt文件），以一个防伪码一行的形式将防伪码写入对应文件中；

4.3 生成含数据分析功能的防伪码

4.3.1． 用户输入这一批产品的分析码，分析码由三位字母组成，分别代表 产地、颜色和批次；用户输入后判断输入的分析码是否符合规则，如果不符合提示用户再次输入，符合规则则使用upper()方法将分析码三位字母转换为大写存储；

4.3.2． 对于用户输入的防伪码数量，按照一样的方法检测、提醒

4.3.3． 分析码与防伪码结合，需要在生成的9位数字的防伪码中间随机按顺序插入分析码；通过分析可得，9位数字的防伪码一共有10个位子可供分析码的三个字母插入，使用random库的sample函数，在0-9的数字中抽取三个位子用于分析码插入；由此组成一个带数据分析功能的12位混合防伪码

4.4 智能批量生成防伪码

4.4.1．读取用户预先编辑好的mri文件；使用QtWidgets.QFileDialog.getOpenFileName调用windows的选择文件框，当用户选择完文件后该函数返回选择文件的绝对路径；

4.4.2． 使用python自带的open函数打开用户选择的文件，文件中是用户预先编辑好的需要批量生成防伪码的信息——分析码和产品数量；两个数据之间提醒用户使用英文“,”分隔；

4.4.3． 打开文件后，对文件中的数据使用readlines()方法进行按行读取，对每一行数据使用split(‘,’)方法分出分析码和产品数量，分别存储；

4.4.4． 遍历存储并对应好的数据，按照生成带数据分析功能的防伪码的方法生成防伪码，并对不同分析码产生的防伪码进行分别存储，文件名按照“分析码+scode-feature5.txt”的格式；

4.4.5. 最后提示用户一共生成了多少个防伪码，存储在哪一个文件夹下；

4.5 后续补加生成防伪码

4.5.1． 与智能批量生成类似，通过QtWidgets.QFileDialog.getOpenFileName调用windows的选择文件框，让用户选择已生成的防伪码文件；

4.5.2． 使用open函数读取文件中的防伪码，存储在一个集合中，使用集合的原因是便于检验是否补充生成了与已生成的防伪码重复的防伪码；对已生成的防伪码中的一个进行分析，提取防伪码中的字母，按照顺序存储，这便是这一批已生成的防伪码的分析码；

4.5.3． 再读取用户输入的需要补充生成的防伪码数量，检测是否符合规则，如果符合则按照生成带数据分析功能的防伪码的方法生成一个防伪码，加入之前的集合中，如果集合长度产生变化说明新生成的防伪码没有重复，是一个符合规则的防伪码，存储在一个列表中；

4.5.4． 重复步骤3直至新生成的防伪码数量符合用户的需求，判断输出文件夹“codeadd”是否存在，将新生成的防伪码输出至该文件夹下的文件中，弹窗提示用户基本信息；

4.6 条形码与二维码

4.6.1． 条形码的生成使用pystrich.ean13中的EAN13Encoder函数，判断用户输入3位数字的国家代码和4位数字的企业代码以及条形码数量符合规则后，使用random库补充生成5位随机码之后，按照“10减去偶数位数之和的3倍和奇数位之和的个位数的个位数”获取次防伪码的校验位，加在12位防伪码的最后；

4.6.2． 使用pystrich.ean13中的EAN13Encoder类生成防伪码对应的条形码图片，调用类函数save保存条形码图片至“barcode”文件夹下；

4.6.3． 二维码的生成与条形码类似，使用qrcode库中的make函数将随机生成的12位防伪码生成对应的二维码图片，保存至“qrcode”文件夹下；

使用pyinstaller库打包程序，打包代码为pyinstaller -F -w main.py

4.7 粉丝抽奖

4.7.1． 用户事先编辑好可以参与抽奖活动的防伪码，并将防伪码保存在一个ini文件中，点击选择文件按钮选择该文件后系统读取文件中的防伪码并存储在列表中

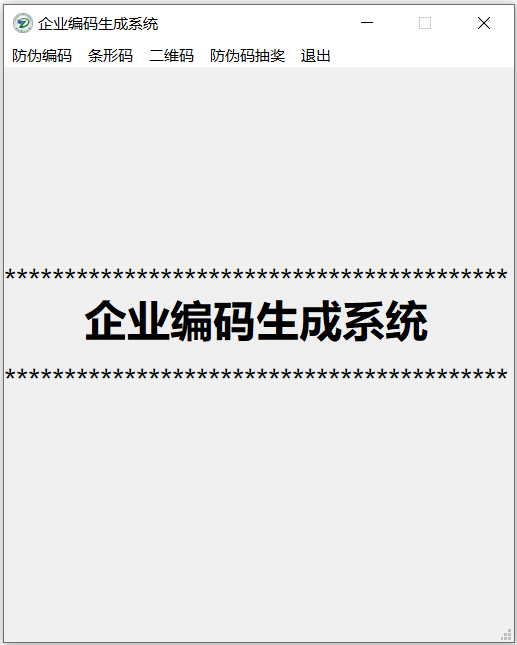
4.7.2． 读取用户输入的中奖用户的数量，判断是否符合规则，如果不符合规则弹窗提示用户；符合规则则使用random库中的sample函数抽取幸运用户，弹窗提示用户中将粉丝对应的防伪码；

4.8 项目打包

使用pyinstaller库打包程序，打包代码为pyinstaller -F -w main.py

# 5、测试

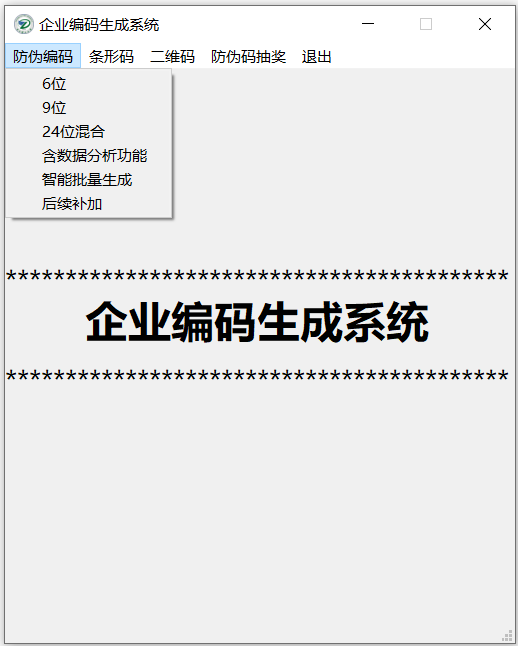
5.1系统启动



**图 5-1 系统启动**

5.2菜单栏

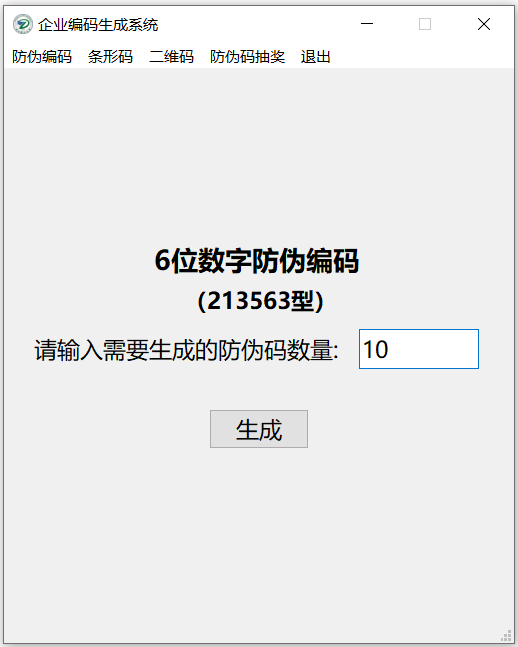
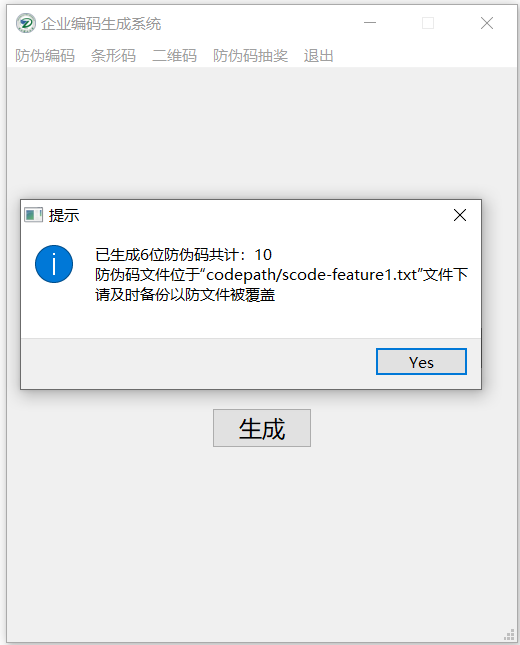
菜单栏中除第一个“防伪编码”具有展开菜单栏外，其余在代码中已设置为QAction对象，以实现单击跳转的功能。



**图 5-2 菜单栏层级关系**

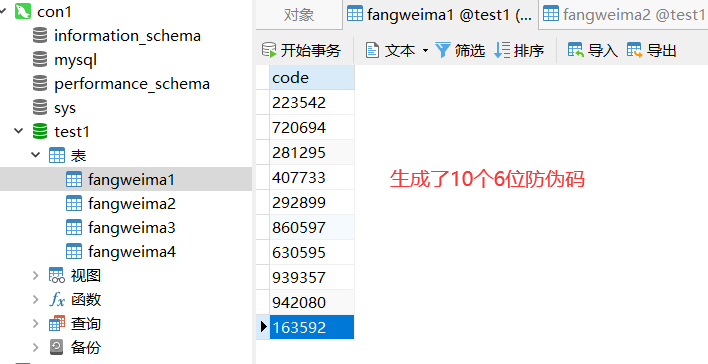
5.3生成各类防伪编码

5.3.1生成6位数字防伪编码

**图 5-3 功能子界面及数据输入**   **图 5-4 生成成功后的提示**

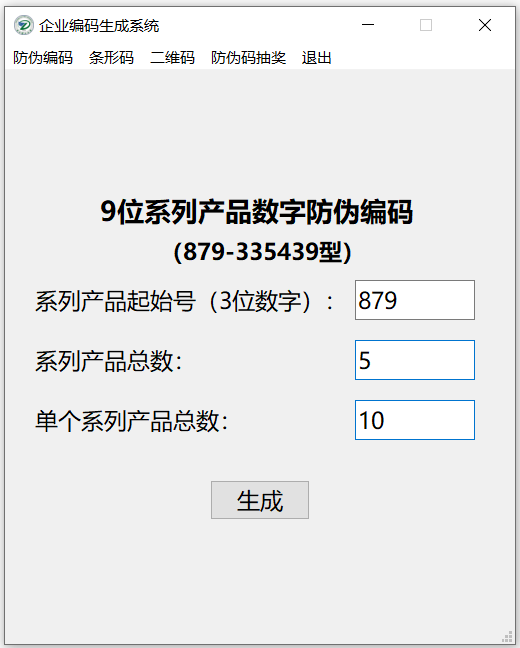
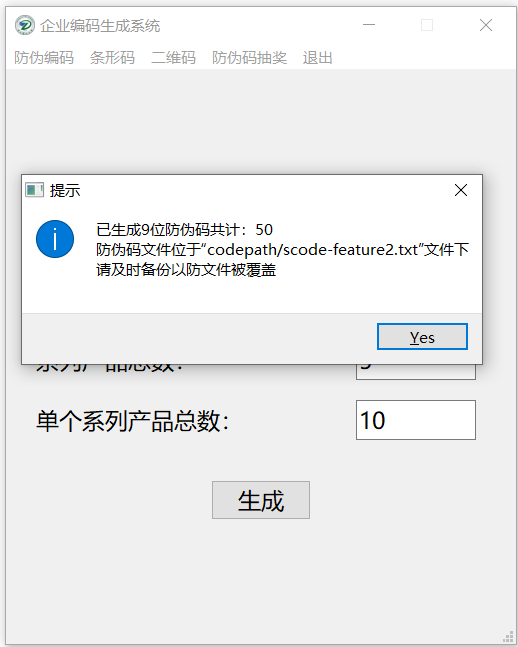
查看对应数据库中生成情况：



**图 5-5 在数据库中验证结果**

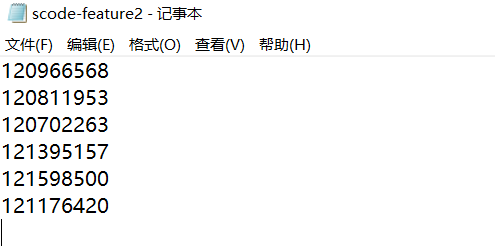
在数据库中的防伪码一方面起到了保存的作用，另一方面也在生成过程中达到了判重，避免重复性的作用。表fangweima1保存的是6位数的防伪码，后面的fangweima2，fangweima3和fangweima4表分别对应保存9位、29位混合以及含数据分析功能的防伪码，后续生成结果就不再贴图展示，与以上类似。

5.3.2生成9位数字防伪编码

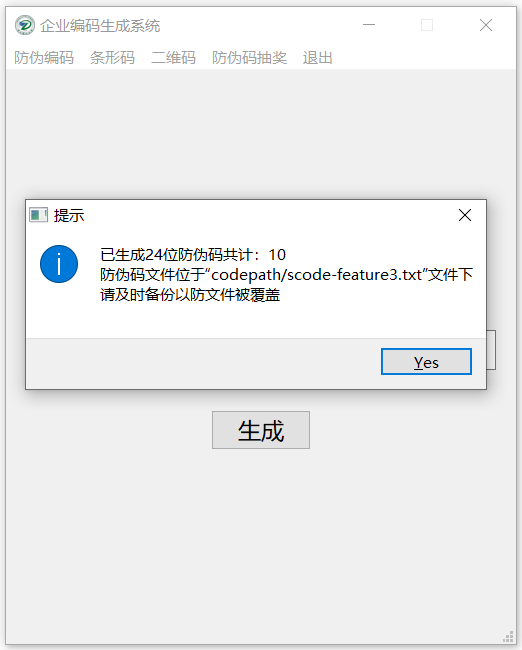
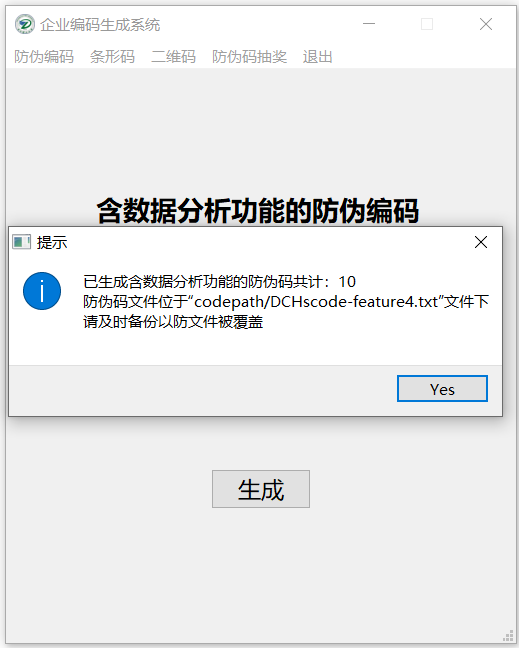
**图 5-6 功能子界面及数据输入 图 5-7 生成成功后的提示**

生成文件scode-feature2.txt，其他生成的防伪码也会类似保存在文件中，即数据库和普通文件中都进行存放。



**图 5-8 在临时文件中验证结果**

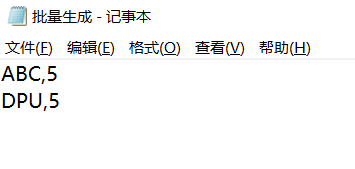
5.3.3生成29位数字防伪编码与带数据分析功能的防伪编码

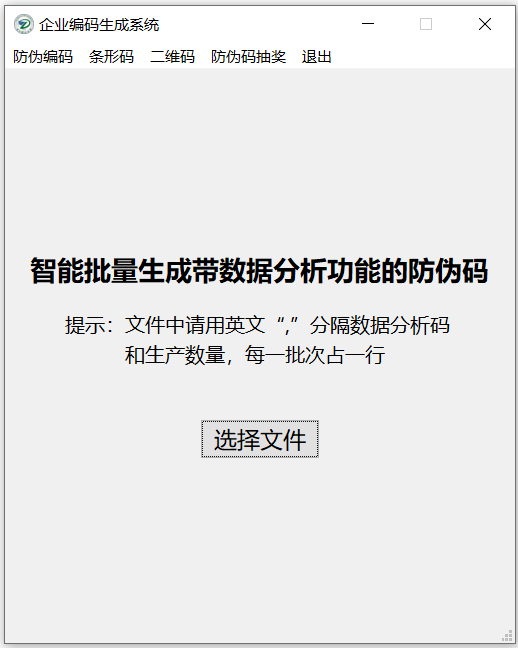
**图 5-9 功能子界面及数据输入 图 5-10 生成成功后的提示**

其他功能与上面类似，为节省篇幅就不截图展示了。

5.3.4智能批量生成防伪编码

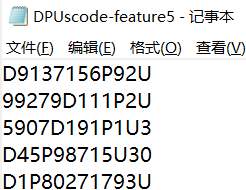
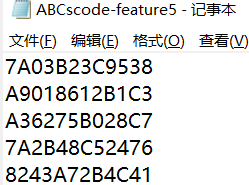


**图 5-11 用于批量生成的文件**



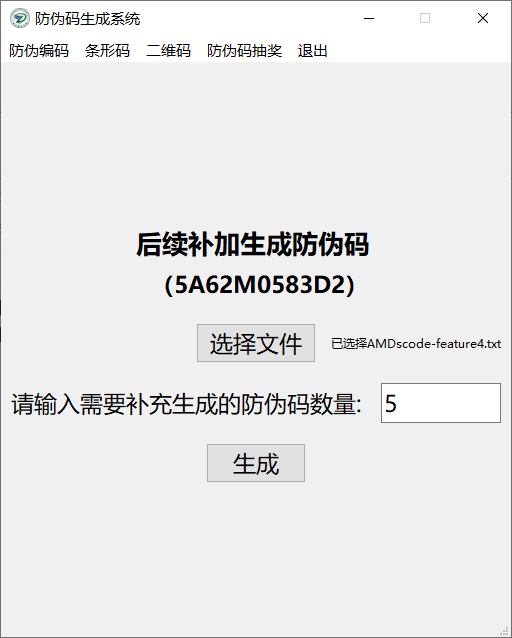
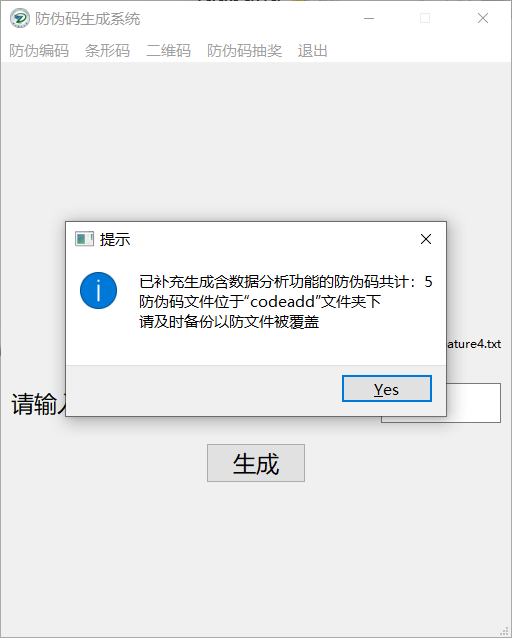
**图 5-12 打开文件并成功生成后的提示**

点击“选择文件“，打开”批量生成.txt“就可以根据格式自动生成如下结果，其中带有ABC的和DPU的数据分析功能的文件结果如下：



**图 5-13 批量生成的文件内容**

5.3.4后续补充生成防伪编码

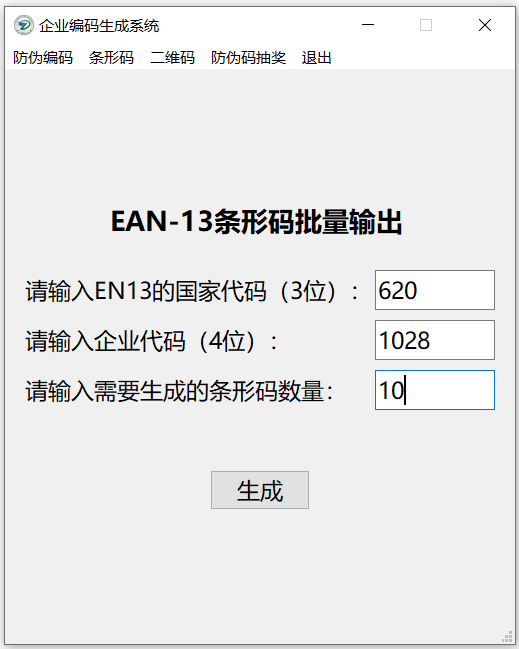
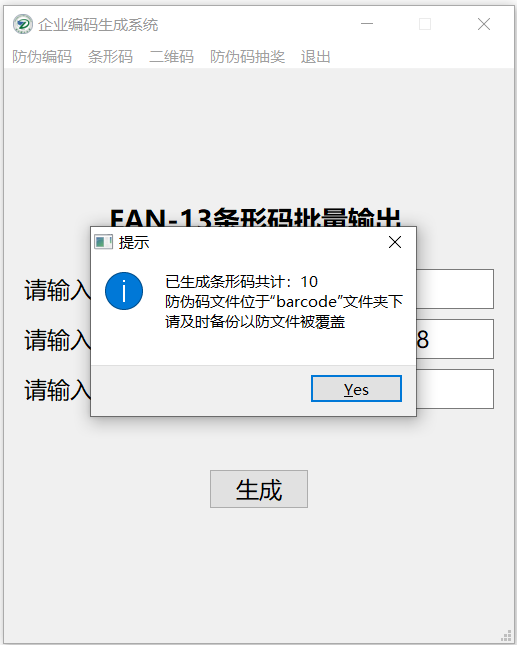
**图 5-14 文件选择结束后的子界面 图 5-15 成功生成后的提示**

可以看到，对于给入的文件，先识别出具体的三位字母，再根据要求生成相应数量的防伪码，起到了后续补加的功能。

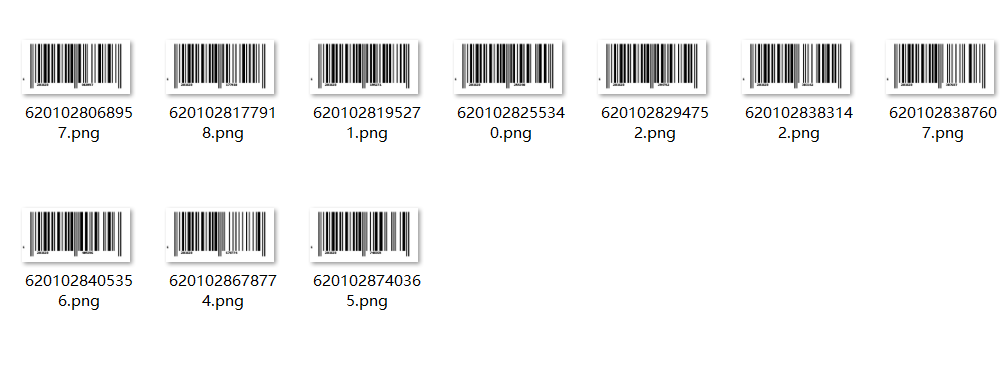


**图 5-16 文件内容**

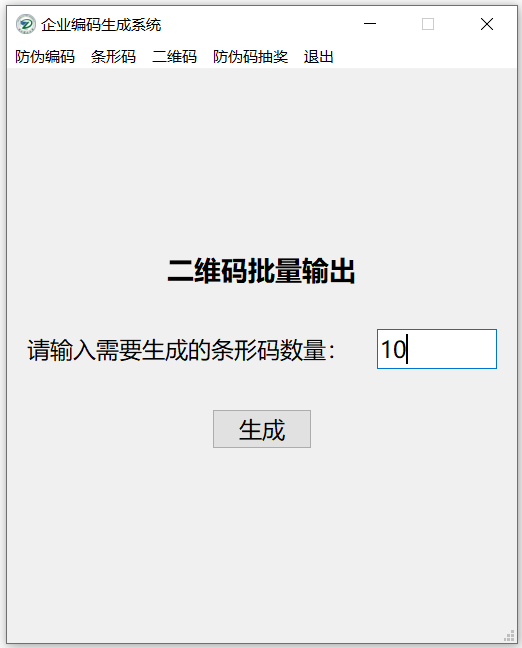
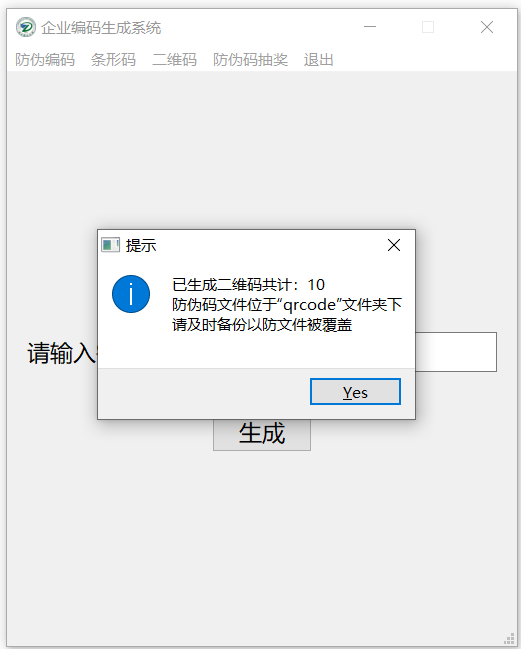
5.3.5生成条形码和二维码

**图 5-17 条形码功能子界面 图 5-18 生成成功后的提示**



**图 5-19 生成的条形码**

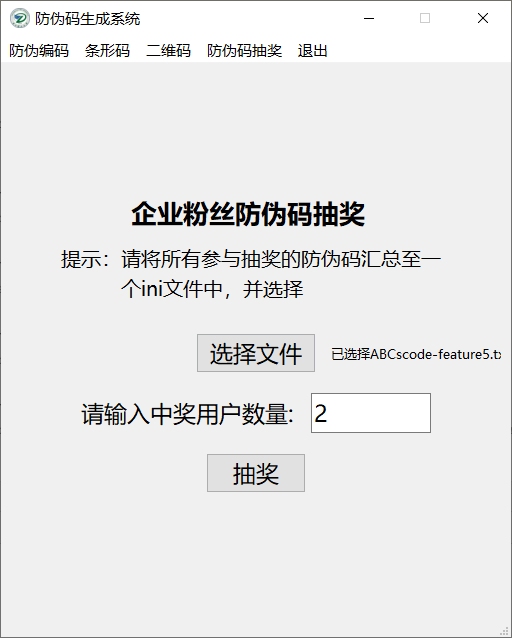
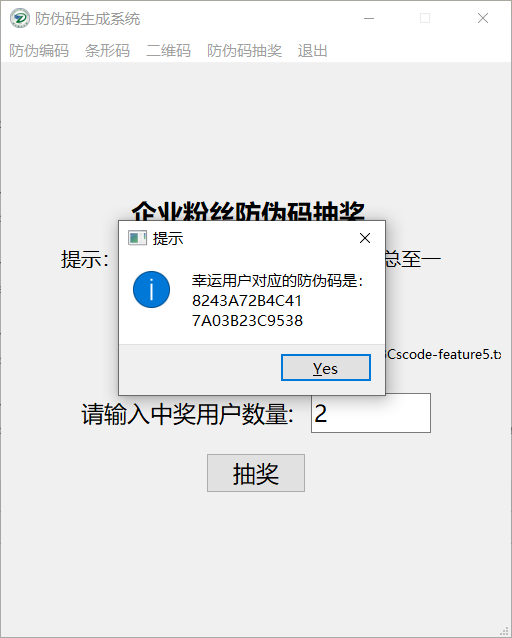
 

**图 5-20 二维码功能子界面 图 5-21 成功生成后的提示**



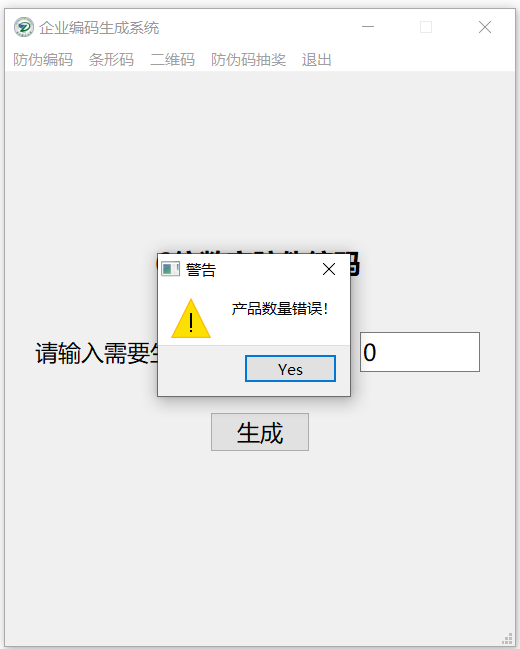
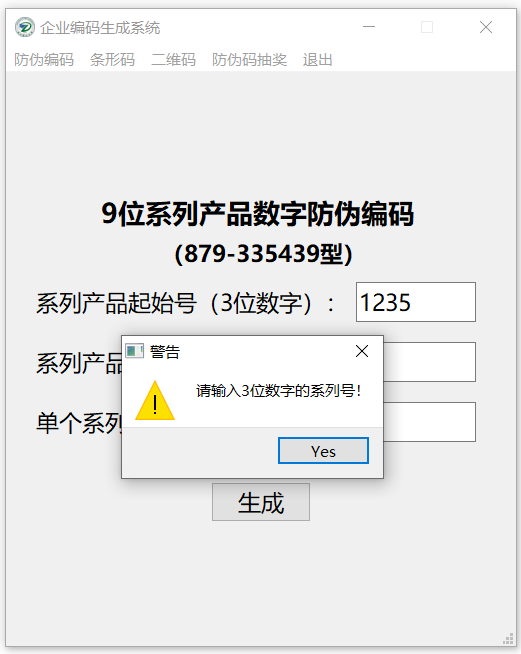
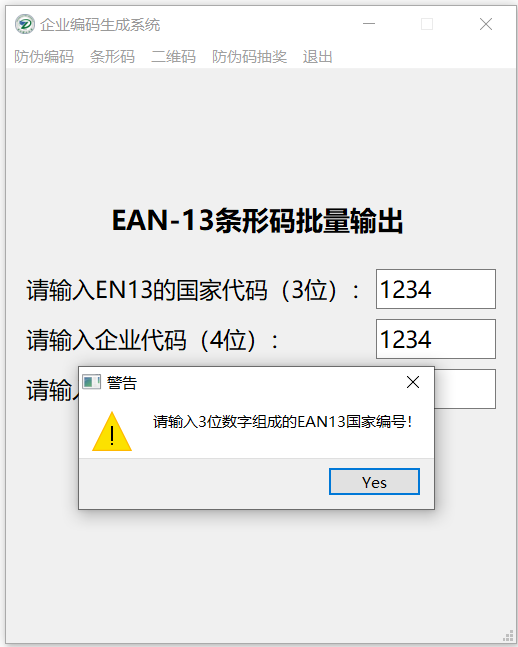
**图 5-22 生成的二维码**

5.3.6粉丝抽奖

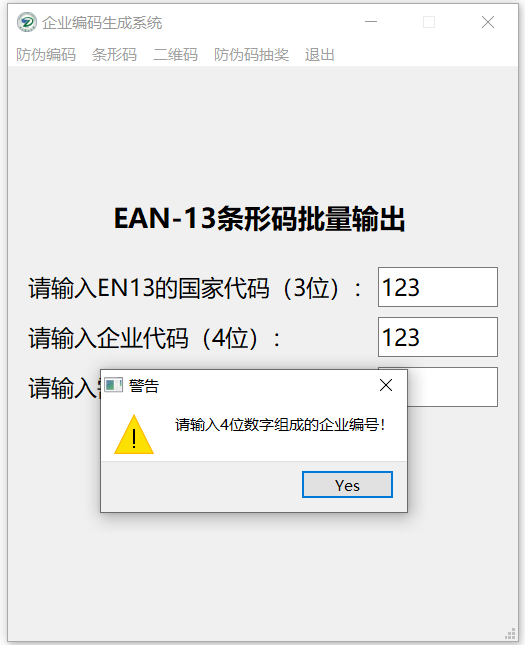
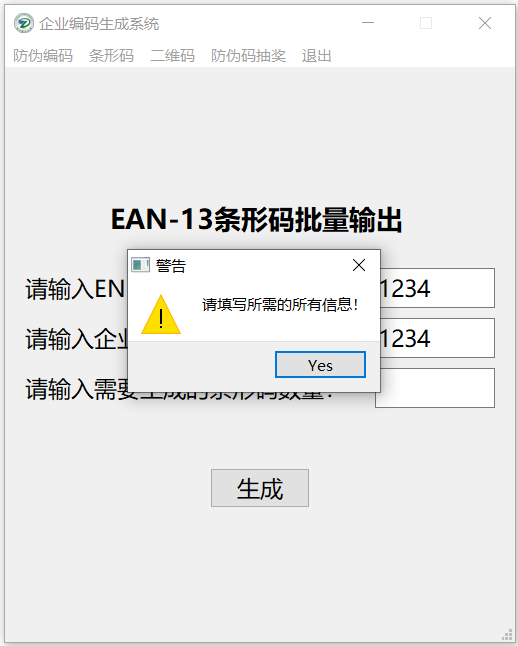
 

**图 5-23 文件选择结束后的子界面 图 5-24 抽奖结果**

5.3.7一些增强程序健壮性的错误提示一览

**图 5-25 数量错误 图 5-26 信息不符合要求1 图 5-27 信息不符合要求2**

**图 5-28 信息不符合要求-3 图 5-29 信息缺失**

# 6、结论

6.1完成情况

本次课程设计选择的是防伪码设计与实现，我从防伪码的不同种类入手，分析查阅得到市面上常见的防伪码类型，并分别动手实操了这些常见防伪码的生成过程，虽然我使用的方法大多采取随机生成等简单策略，但所谓大道至简，由随机生成的防伪码依然具有不可逆性、不可预测性等特质。同时，我是用python连接数据库，对于生成的防伪码进行自我保存，这样一方面可以对防伪码进行认证，另一方面也可以在生成新防伪码的同时防止碰撞的发生，可谓一举两得。

为了做到功能尽可能全面与展示效果美观，我也使用了pyqt5来制作可视化可交互的UI界面，做到让功能全面而有条理，使用起来也方便且容易上手。在我们的生活中，防伪码、条形码和二维码随着技术的发展变得越来越普遍，这也使得这次项目的内容十分地贴近我们的生活。至于有待改进的地方，我想就是请教和探讨工业界实际所使用的防伪码生成技术，理解他们的生成原理和效果评价，在本次课程设计的基础上进行完善，争取做出更全面化、多样化、智能化和更加健壮的防伪认证码。

6.2心得体会

本次课程设计与课程大作业分同时进行，时间紧任务重，于是我充分考虑了对数据信息安全相关知识的掌握和自身知识储备后选择了这样的一个题目。其实一开始我对于防伪码的生成也是一知半解，甚至都不知道防伪认证码的作用是什么，后来经过不断学习以及多方查找，我明白和认清了防伪认证码的意义并在此基础展开。因为我毕竟没有特别深入和系统的去学习过一些防伪认证码的生成算法，自己也担心它太难无法保证投入的时间一定可以做出成果。于是我打算从另一个角度出发，多方面了解了不同的防伪码，有一定深度基础之后在横向上进行扩充，并由此设计了不同位数、类型的结果。同时，为了进一步扩展功能，我还还突发奇想添加了条形码、二维码等操作，虽然只是调用了python强大的外部库来进行的生成，但充分增强了程序功能的完整性和多样性，也算是锦上添花之笔。

在实验过程中，也产生遇到过不少问题。比如我明明用python成功连接到了数据库，也没有产生报错，却始终无法将数据真正写入。后来经过查阅资料和单步调试等操作，我发现是因为上一步连入数据库查询时创建的游标cursor没有完成提交和关闭，造成了事务未能充分完成。经过改正之后，我终于做到了自如的使用python操作MySQL，进一步完成查找和更新等操作，也感受到了方便和成功的喜悦。

通过本次实验，我重新加深了对防伪认证码等在数据信息安全方面知识的理解和掌握，也有对曾经学习过的如pyqt5可视化、python连接数据库等操作复习和利用。我充分意识到曾经学习过的知识的益处与重要性，也希望将本学期所学融会贯通，做到在今后的学习生活中举一反三、充分利用好所学，让知识为自己所用。

# 参考文献

[1] Eric Matthes. Python编程：从入门到实践[M].人民邮电出版社:北京,2016

[2] 明日科技. Python项目开发案例集锦[M]. 机灵大学出版社:长春,2019

[3] 郑秋生,夏敏捷. Python项目案例开发从入门到实践[M].清华大学出版社:北京,2019

[4]姜华林.基于PyQt5界面的词云制作软件设计[J].电脑知识与技术,2021,17(13):74-76+92.

[5] 亦文.数字防伪认证系统的运作理念[J].中国防伪,2003(03):56-57.

[6] 魏超. 基于混沌加密的点阵防伪码研究及应用[D].哈尔滨工业大学,2017.