广西科技大学

《软件测试与质量保证》测试报告

学 院: 启迪数字学院

班 级：软件Q163班

姓 名：李远树

学 号：201609601133

指导老师：胡伟平

日 期：2019年 5月18日

### 1. 测试背景

#### 测试内容

本次测试的软件对象是一个小型的电话查询系统。测试内容主要有功能测试和性能测试两个方面。

在功能测试方面，只为用户提供一个查询界面，针对用户的查询习惯和系统的查询功能，主要包含以下几个测试内容：

1. 号码的位数。本次的电话号码查询系统包括3位、5位、7位、8位、11位这几种类型的号码。
2. 号码的首位数。三位的紧急服务号码开头只能为1或9，五位的服务号码开头是1或9，七位或者八位数的号码(本地号码)开头不能为0、1、9，十一位数的电话号码(外地号码)区号长度为三的开头只能为0或8，十一位数的电话号码(外地号码)区号长度为四的开头只能为0，十一位数的手机号码只能为1.
3. 电话号码的字符是否合法。十一位数的电话号码只能包括纯数字。
4. 号码段的合法性。座机号码的区号必须是各个城市的区号子集，并且座机号码区号后面的第一位不能为0、1、9（即若座机号码的区号为3位，则第4位不能是0,1,9；若座机号码的区号为4位，则第5位不能是0,1,9）。手机号码的前三位必须属于中国移动，中国电信，中国联通这三大服务商之一。手机号码的前位必须要在中国归属地的号码集中。

在性能测试方面：主要测试号码的从点击“查询”按钮到查询结果出现所需要的

即查询速度。本次的电话查询系统利用到数据库，在性能测试的过程中可以通过测试一些查询语句或者创建视图之类的，来测试电话系统的性能。

#### 1.2测试方法

电话归属地查询系统主要是用黑盒测试和白盒测试两种方法。

在黑盒测试中，先对号码进行等价类划分，再建立决策表，这样可以相对减少测试用例的个数，提高测试效率；最后再设定测试用例，利用测试用例来检查电话号码归属地查询系统的功能实现情况。

在白盒测试中，只对11位数的号码进行测试，首先根据程序代码画出流程图，再根据流程图画出控制流图，这样就可以得出11位数的号码查询时出现全部可能。再根据控制流图的全部覆盖路径，决定每一个测试用例，完成白盒测试。

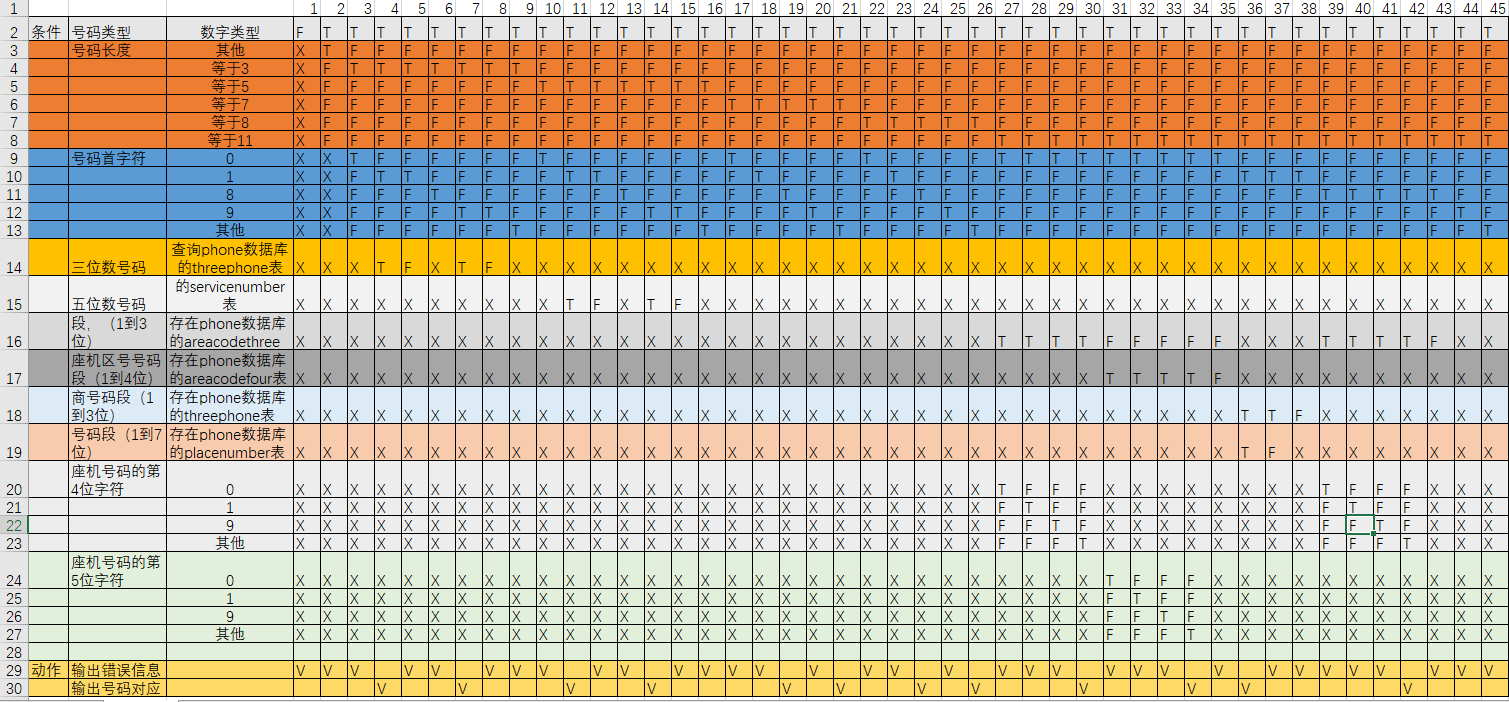
#### 1.3测试软硬件环境

华硕笔记本电脑一台，python开发环境，Pycharm编码器，Mysql数据库

### 2. 黑盒测试

#### 2.1决策表

根据第一部分的测试内容，可以为电话号码查询系统划分出六个判断条件，若满足其中的某些条件可以输出正确的号码查询结果，否则输出错误的提示信息。根据这些条件和执行动作，做出的决策表如下（表格行数和列数太多，无法直接在word文档中插入，具体请查看软件测试.xlsx——>决策表）：



#### 2.2 用例表

用例表包含四个内容：用例编号，用例输入，预期结果，实际结果，结果是否一致。根据决策表可以设计出45个用例。其中43个测试用例的实际结果和预期结果一致；2个测试用例的实际结果与预期结果不一致，原因为数据库中没有包含足够多的数据，导致本来正确的号码查询不到结果。

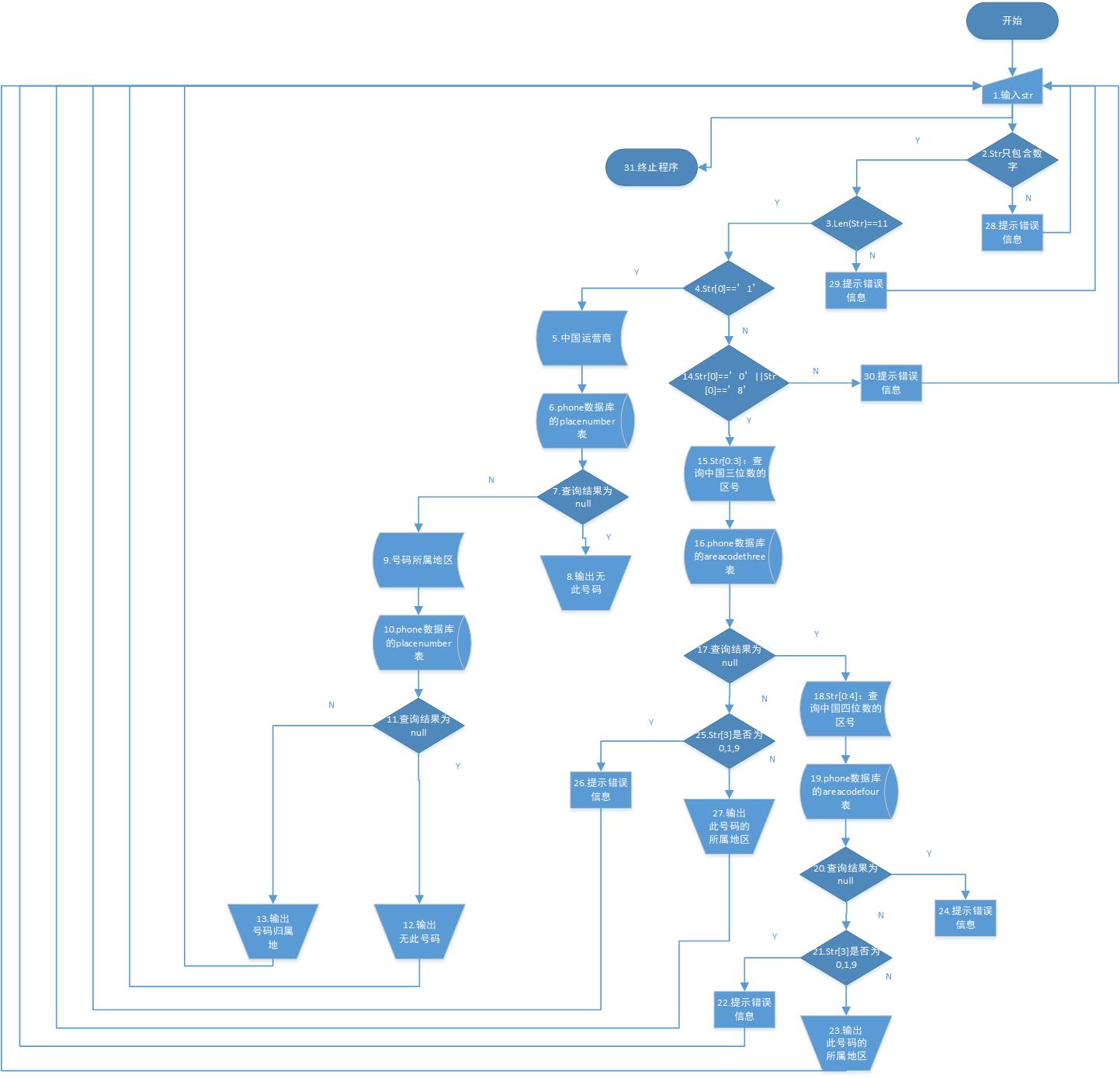
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 用例（输入号码） | 预期结果 | 实际结果 | 是否一致 |
| 1 | 123a | 号码只能包含数字哦，请重新输入 | 号码只能包含数字哦，请重新输入 | 是 |
| 2 | 12 | 号码的长度不对哦,请重新输入 | 号码的长度不对哦,请重新输入 | 是 |
| 3 | 012 | 号码开头只能是1或9，请重新输入 | 号码开头只能是1或9，请重新输入 | 是 |
| 4 | 110 | 报警服务台 | 报警服务台 | 是 |
| 5 | 123 | 无法查询到您输入的号码信息 | 无法查询到您输入的号码信息 | 是 |
| 7 | 889 | 号码开头只能是1或9，请重新输入 | 号码开头只能是1或9，请重新输入 | 是 |
| 8 | 999 | 红十字会急救台 | 红十字会急救台 | 是 |
| 9 | 911 | 无法查询到您输入的号码信息 | 无法查询到您输入的号码信息 | 是 |
| 10 | 520 | 号码开头只能是1或9，请重新输入 | 号码开头只能是1或9，请重新输入 | 是 |
| 11 | 01369 | 号码开头只能是1或9，请重新输入 | 号码开头只能是1或9，请重新输入 | 是 |
| 12 | 10086 | 中国移动服务电话 | 中国移动服务电话 | 是 |
| 13 | 12348 | 法律服务 | 无法查询到您输入的号码信息 | 否 |
| 14 | 88888 | 号码开头只能是1或9，请重新输入 | 号码开头只能是1或9，请重新输入 | 是 |
| 15 | 95119 | 森林火警 | 森林火警 | 是 |
| 16 | 99999 | 无法查询到您输入的号码信息 | 无法查询到您输入的号码信息 | 是 |
| 17 | 23232 | 号码开头只能是1或9，请重新输入 | 号码开头只能是1或9，请重新输入 | 是 |
| 18 | 0123456 | 号码开头不能是0或1或9，请重新输入 | 号码开头不能是0或1或9，请重新输入 | 是 |
| 19 | 1123456 | 号码开头不能是0或1或9，请重新输入 | 号码开头不能是0或1或9，请重新输入 | 是 |
| 20 | 8765432 | 号码为本地座机号码，当前地区为广西柳州 | 号码为本地座机号码，当前地区为广西柳州 | 是 |
| 21 | 9987123 | 号码开头不能是0或1或9，请重新输入 | 号码开头不能是0或1或9，请重新输入 | 是 |
| 22 | 2323677 | 号码为本地座机号码，当前地区为广西柳州 | 号码为本地座机号码，当前地区为广西柳州 | 是 |
| 23 | 01234567 | 号码开头只不能是0或1或9，请重新输入 | 号码开头只不能是0或1或9，请重新输入 | 是 |
| 24 | 11234567 | 号码开头只不能是0或1或9，请重新输入 | 号码开头只不能是0或1或9，请重新输入 | 是 |
| 25 | 89891889 | 号码错误，柳州本地号码只能为7位数 | 号码错误，柳州本地号码只能为7位数 | 是 |
| 26 | 98765432 | 号码开头只不能是0或1或9，请重新输入 | 号码开头只不能是0或1或9，请重新输入 | 是 |
| 27 | 52013142 | 号码错误，柳州本地号码只能为7位数 | 号码错误，柳州本地号码只能为7位数 | 是 |
| 28 | 01001234567 | 号码区号：北京市;但号码第4位不能是0或1或9，请重新输入 | 号码区号：北京市;但号码第4位不能是0或1或9，请重新输入 | 是 |
| 29 | 01011234567 | 号码区号：北京市;但号码第4位不能是0或1或9，请重新输入 | 号码区号：北京市;但号码第4位不能是0或1或9，请重新输入 | 是 |
| 30 | 01091234567 | 号码区号：北京市;但号码第4位不能是0或1或9，请重新输入 | 号码区号：北京市;但号码第4位不能是0或1或9，请重新输入 | 是 |
| 31 | 02161234567 | 上海市 | 上海市 | 是 |
| 32 | 03110123456 | 号码区号：河北省石家庄;但号码第5位不能是0或1或9，请重新输入 | 号码区号：河北省石家庄;但号码第5位不能是0或1或9，请重新输入 | 是 |
| 33 | 03121123457 | 号码区号：河北省保定市;但号码第5位不能是0或1或9，请重新输入 | 号码区号：河北省保定市;但号码第5位不能是0或1或9，请重新输入 | 是 |
| 34 | 03139123458 | 号码区号：河北省张家口;但号码第5位不能是0或1或9，请重新输入 | 号码区号：河北省张家口;但号码第5位不能是0或1或9，请重新输入 | 是 |
| 35 | 05724123459 | 浙江省湖州市 | 浙江省湖州市 | 是 |
| 36 | 13014886666 | 广西柳州联通 | 广西柳州联通 | 是 |
| 37 | 15277470258 | 广西梧州移动 | 查询不到结果，号码错误或者数据库无记录 | 否 |
| 38 | 19952011314 | 错误，中国无此运营商，请重新输入 | 错误，中国无此运营商，请重新输入 | 是 |
| 39 | 85201234567 | 号码区号：香港;但号码第4位不能是0或1或9，请重新输入 | 号码区号：香港;但号码第4位不能是0或1或9，请重新输入 | 是 |
| 40 | 85218764567 | 号码区号：香港;但号码第4位不能是0或1或9，请重新输入 | 号码区号：香港;但号码第4位不能是0或1或9，请重新输入 | 是 |
| 41 | 85298764567 | 号码区号：香港;但号码第4位不能是0或1或9，请重新输入 | 号码区号：香港;但号码第4位不能是0或1或9，请重新输入 | 是 |
| 42 | 85356898593 | 澳门 | 澳门 | 是 |
| 43 | 98765432100 | 号码开头只能是0或1或8，请重新输入 | 号码开头只能是0或1或8，请重新输入 | 是 |
| 44 | 66666666666 | 号码开头只能是0或1或8，请重新输入 | 号码开头只能是0或1或8，请重新输入 | 是 |

### 3. 白盒测试

在电话号码归属地查询系统的白盒测试中，主要针对11位数的手机号码。会通过绘制这部分的流程图，控制流图，然后根据控制流图得出独立路径，再根据独立路径设计测试用例，完成白盒测试。

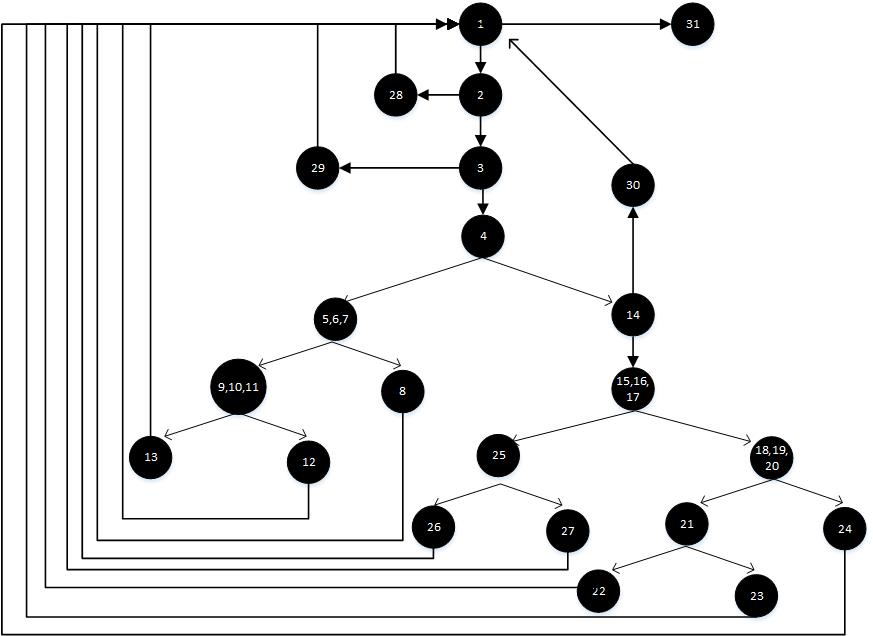
#### 3.1 流程图

在11位数的号码中，包括电话号码和手机号码。手机号码的首位字符必须为1，号码的前三位字符决定号码属于哪个运行商，号码的前七位字符决定号码属于哪个运行商和所在地。这两部分的对应信息分别存放于phone数据库的serviceprovide表和placenumber表。电话号码区号所属地由号码的前三位或者前四位决定，对应信息分别存放于phone数据库的areacodethree表和areacodefour表；若在数据库中能查到对应信息，还需要判断号码的特定位置的字符是否符合要求：若号码的区号为3位，则第4位不能是0,1,9；若号码的区号为4位，则第5位不能是0,1,9。最后号码符合要求，这输出号码归属地信息，否则输出错误提示信息。号码无论是否正确，都可以进行多次查询，直到关闭窗口，代表终止程序运行。由以上条件并且结合程序，可得出以下流程图（图片所属文件：流程图.jpg）



#### 3.2控制流图

由11位号码的程序流程图，可以得出11位电话号码的控制流图，其中控制流图每一个数值都有对应流程图的步骤数值。（图片所属文件：控制流图.jpg）



#### 3.3 独立路径

独立路径可以由控制流图得出，首先依照公式可得出总的独立路径个数。计算公式为：

独立路径总个数=分支数+1。由图可得共有11个分支。即独立路径总个数为12。再根据控制流图可得出以下独立路径：

Path1：1，31

Path2：1，2，28，1，31

Path3：1，2，3，29，1，31

Path4：1，2，3，4，5，6，7，9，10，11，13，1，31

Path5：1，2，3，4，5，6，7，9，10，11，12，1，31

Path6：1，2，3，4，5，6，7，8，1，31

Path7：1，2，3，4，14，30，1，31

Path8：1，2，3，4，14，15，16，17，25，26，1，31

Path9：1，2，3，4，14，15，16，17，25，27，1，31

Path10：1，2，3，4，14，15，16，17，18，19，20，21，22，1，31

Path11：1，2，3，4，14，15，16，17，18，19，20，21，23，1，31

Path12：1，2，3，4，14，15，16，17，18，19，20，24，1，31

#### 3.4 用例表

用例表共包含六项内容：用例编号、覆盖路径、用例输入、预期结果、实际结果、结果是否一致。根据独立路径共设计出12个测试用例，这样可以覆盖了全部路径，确保程序能够带给用户好的使用体验。有一个测试用例的实际结果与预期结果不一致，原因为数据库中没有包含足够多的数据，导致本来正确的号码查询不到结果。测试用例表如下(或者查看软件测试.xlsx——>路径覆盖用例表)：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 用例编号 | 覆盖路径 | 用例(str的值) | 预期结果 | 实际结果 | 是否一致 |
| 1 | path1 | 无输入，关闭程序 | 退出程序 | 退出程序 | 是 |
| 2 | path2 | 198rr | 号码只能包含数字哦，请重新输入 | 号码只能包含数字哦，请重新输入 | 是 |
| 3 | path3 | 152774702581 | 号码的长度不对哦,请重新输入 | 号码的长度不对哦,请重新输入 | 是 |
| 4 | path4 | 13005930258 | 广西柳州联通 | 广西柳州联通 | 是 |
| 5 | path4 | 15277470258 | 广西梧州移动 | 查询不到结果，号码错误或者数据库无记录 | 否 |
| 6 | path5 | 13977470258 | 查询不到结果，号码错误或者数据库无记录 | 查询不到结果，号码错误或者数据库无记录 | 是 |
| 7 | path6 | 11977470258 | 错误，中国无此运营商，请重新输入 | 错误，中国无此运营商，请重新输入 | 是 |
| 8 | path7 | 71977470258 | 号码开头只能是0或1或8，请重新输入 | 号码开头只能是0或1或8，请重新输入 | 是 |
| 9 | path8 | 01012345678 | 号码区号：北京市;但号码第4位不能是0或1或9，请重新输入 | 号码区号：北京市;但号码第4位不能是0或1或9，请重新输入 | 是 |
| 10 | path9 | 01022345678 | 北京市 | 北京市 | 是 |
| 11 | path10 | 03101234567 | 号码区号：河北省邯郸市;但号码第5位不能是0或1或9，请重新输入 | 号码区号：河北省邯郸市;但号码第5位不能是0或1或9，请重新输入 | 是 |
| 12 | path11 | 03102234567 | 河北省邯郸市 | 河北省邯郸市 | 是 |

### 4. 小结

《软件测试》课程的课程设计作业为设计一个电话号码查询系统。主要包含两大部分的工作：程序的设计和代码实现；对系统进行测试和编写测试文档。

程序的设计和代码实现这部分内容，在程序设计方面主要是先定义一个函数实现对输入号码的检查，若符合条件返回想对应的值，不符合条件则返回相对应的错误信息；再实现一个GUI界面，若用户输入的号码符合条件，就连接到数据库相对于的数据表进行查询，并且返回对应的结果给用户。在代码实现方面使用的是python实现，主要功能都已经实现，但在代码规范方面严重不足：代码注释不多，函数和变量命名不符合规范，希望以后能够改正这些缺点。

系统测试和编写测试文档这部分内容，在黑盒测试过程中设计决策表时，不会划分等价类，经过阅读课本和网上查阅治疗已经解决这个问题，并且已完成黑盒测试的决策表和测试用例部分内容；白盒测试部分比较简单，只有根据代码画出11位号码的流程图后，就可以根据流程图完成白盒测试的所有内容。

本次的课程设计作业因为开始做得比较晚，所以完成得比较匆忙，但所幸的是目前为止已完成全部内容。希望以后的作业完成能做得不拖拉，提前开始做老师布置的作业，写出一份更完美的作业。

### 附：程序源码

**import** pymysql  
**import** sys  
**from** PyQt5.QtWidgets **import** QWidget, QApplication, QGridLayout, QLabel, QLineEdit, QPushButton  
  
*# 检查号码是否符合要求***def** checkPhone(str):  
 str\_error=**''** *# 判断str是否全部为数字* **if** (str.isdigit()):  
 length=len(str)  
 **if** length == 3 :  
 **if** str[0]==**'1' or** str[0]==**'9'**:  
 **return** 3  
 **else**:  
 str\_error = **'号码开头只能是1或9，请重新输入'  
 return** str\_error  
 **elif** length == 5 :  
 **if** str[0]==**'1' or** str[0]==**'9'**:  
 **return** 5  
 **else**:  
 str\_error = **'号码开头只能是1或9，请重新输入'  
 return** str\_error  
 **elif** length ==7:  
 **if** str[0] != **'1' and** str[0] != **'9' and** str[0] != **'0'**:  
 **return** 7  
 **else**:  
 str\_error = **'号码开头不能是0或1或9，请重新输入'  
 return** str\_error  
 **elif** length == 8:  
 **if** str[0] != **'1' and** str[0] != **'9' and** str[0] != **'0'**:  
 **return** 8  
 **else**:  
 str\_error = **'号码开头只不能是0或1或9，请重新输入'  
 return** str\_error  
 **elif** (length == 11):  
 *# 手机号码* **if** str[0] == **'1'**:  
 **return** 12  
 *# 座机号码* **elif** str[0] == **'0' or** str[0]==**'8'**:  
 **return** 11  
 **else**:  
 str\_error = **'号码开头只能是0或1或8，请重新输入'  
 return** str\_error  
 **else**:  
 str\_error = **'号码的长度不对哦,请重新输入'  
 return** str\_error  
 **else**:  
 str\_error = **'号码只能包含数字哦，请重新输入'  
 return** str\_error  
  
**class** Demo(QWidget):  
 **def** \_\_init\_\_(self, parent=**None**):  
 *# 设计一个窗口查询界面* super().\_\_init\_\_()  
 self.setWindowTitle(**'电话号码查询系统'**)  
 self.Label1 = QLabel(**'电话号码'**)  
 self.Label2 = QLabel(**'查询结果'**)  
 self.LineEdit1 = QLineEdit()  
 self.LineEdit2 = QLineEdit()  
 self.LineEdit2.setReadOnly(**True**)  
 self.translateButton1 = QPushButton()  
 self.translateButton1.setText(**'查询'**)  
 self.grid = QGridLayout()  
 self.grid.setSpacing(12)  
 self.grid.addWidget(self.Label1, 1, 0)  
 self.grid.addWidget(self.LineEdit1, 1, 1)  
 self.grid.addWidget(self.Label2, 2, 0)  
 self.grid.addWidget(self.LineEdit2, 2, 1)  
 self.grid.addWidget(self.translateButton1, 1, 2)  
 self.setLayout(self.grid)  
 self.resize(800, 350)  
 self.translateButton1.clicked.connect(**lambda**: self.translate())  
 *# 监听“查询”按钮事件，并且响应* **def** translate(self):  
 str = self.LineEdit1.text()  
 conn = pymysql.connect(host=**'127.0.0.1'**, port=3306, user=**'root'**, passwd=**'root'**, db=**'phone'**, charset=**'utf8'**)  
 cur = conn.cursor()  
 results = **"无法查询到您输入的号码信息"** *# 输入号码类型或者长度错误后，返回相应的错误结果* **if** type(checkPhone(str))==type(**'111'**):  
 str\_error=checkPhone(str)  
 self.LineEdit2.setText(str\_error)  
 **return  
 if not** str:  
 **return** *# 号码长度为3并且首位字符符合要求，查询phone数据库的threePhone表* **if** (checkPhone(str) == 3):  
 cur.execute(**'select** *\** **from threePhone where phone='** + str)  
 rows = cur.fetchall()  
 **if** (rows == ()):  
 self.LineEdit2.setText(results)  
 **else**:  
 **for** row **in** rows:  
 results = row[1]  
 **try**:  
 self.LineEdit2.setText(results)  
 **except** Exception **as** e:  
 print(**'---->'**, e)  
 *# 号码长度为3并且首位字符符合要求，查询phone数据库的servicenumber表* **elif**(checkPhone(str)==5):  
 cur.execute(**'select** *\** **from servicenumber where phone='** + str)  
 rows=cur.fetchall()  
 **if** (rows == ()):  
 self.LineEdit2.setText(results)  
 **else**:  
 **for** row **in** rows:  
 results = row[1]  
 **try**:  
 self.LineEdit2.setText(results)  
 **except** Exception **as** e:  
 print(**'---->'**, e)  
 **elif**(checkPhone(str)==7):  
 results=**"号码为本地座机号码，当前地区为广西柳州"** self.LineEdit2.setText(results)  
 **elif** (checkPhone(str) == 8):  
 results = **"号码错误，柳州本地号码只能为7位数"** self.LineEdit2.setText(results)  
 **elif** (checkPhone(str) == 11):  
 cur.execute(**'select** *\** **from areacodethree where phone='** + str[0:3])  
 rows = cur.fetchall()  
 **if** (rows == ()) **and** str[0]!=**'8'**:  
 cur.execute(**'select** *\** **from areacodefour where phone='** + str[0:4])  
 rows = cur.fetchall()  
 **if** (rows == ()):  
 self.LineEdit2.setText(results)  
 **else**:  
 **for** row **in** rows:  
 results = row[1]  
 **if** str[4] == **'1' or** str[4] == **'9' or** str[4] == **'0'**:  
 self.LineEdit2.setText(**"号码区号："** + results + **';但号码第5位不能是0或1或9，请重新输入'**)  
 **else**:  
 self.LineEdit2.setText(results)  
 **else**:  
 **for** row **in** rows:  
 results = row[1]  
 **if** str[3] == **'1' or** str[3] == **'9' or** str[3] == **'0'**:  
 self.LineEdit2.setText(**"号码区号："** + results + **';但号码第4位不能是0或1或9，请重新输入'**)  
 **else**:  
 self.LineEdit2.setText(results)  
  
 *# 号码长度为11并且首位字符符合要求，查询phone数据库的serviceprovide表* **elif** (checkPhone(str) == 12):  
 cur.execute(**'select** *\** **from serviceprovide where phone='** + str[0:3])  
 rows = cur.fetchall()  
 **if** (rows == ()):  
 self.LineEdit2.setText(**'错误，中国无此运营商，请重新输入'**)  
 **else**:  
 **for** row **in** rows:  
 results = row[1]  
 cur.execute(**'select** *\** **from placenumber where phone='** + str[0:7])  
 rows = cur.fetchall()  
  
 **if** (rows == ()):  
 self.LineEdit2.setText(**'查询不到结果，号码错误或者数据库无记录'**)  
 **else**:  
 **for** row **in** rows:  
 results = row[1]  
 self.LineEdit2.setText(results)  
 cur.close()  
 conn.commit()  
 conn.close()  
  
**if** \_\_name\_\_ == **'\_\_main\_\_'**:  
 app = QApplication(sys.argv)  
 demo = Demo()  
 demo.show()  
 sys.exit(app.exec\_())