## 实验五Tkinter图形界面设计

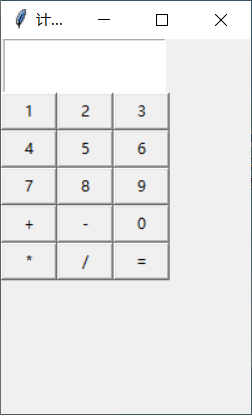
1. **实验目的**

熟练Tkinter库，能熟练运用Tkinter库中的各类控件开发图形界面

**（二）实验内容**

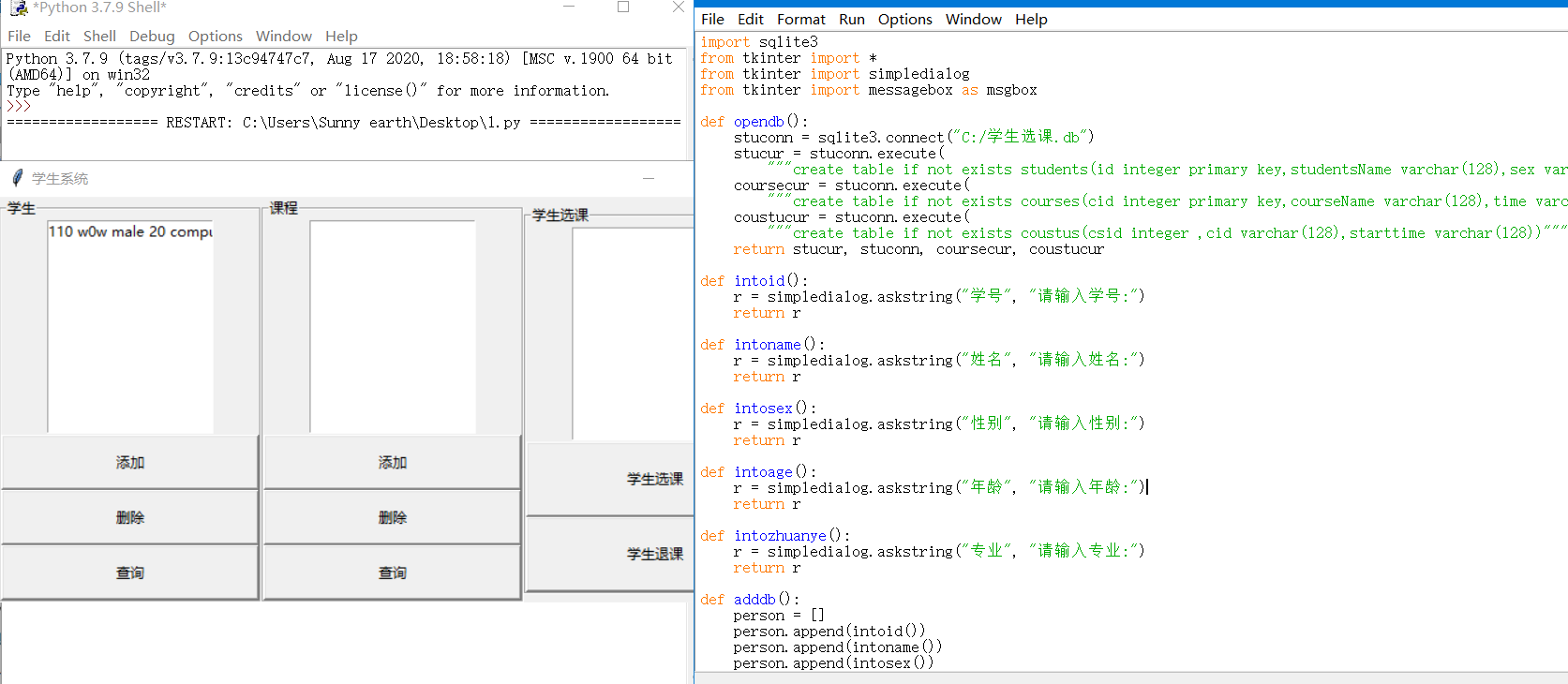
1. 设计并实现一个简单计算器，计算机器能实现基本的算术运算，包含如下键：0~9个数字键，加减乘除键，退格键，等号键。

|  |
| --- |
| from tkinter import \*  class Calc:  def \_\_init\_\_(self, root):  self.root = root  self.num1 = 0  self.num2 = 0  self.sign = None  self.if\_equal = None  self.init\_root()  def init\_root(self):  self.root.title('计算器')  self.root.geometry('200x300+10+10')  # 设置按钮  self.button\_1 = Button(self.root, text=1, width=5, command=lambda: self.show\_num(1))  self  .button\_1.grid(row=1, column=0)  self.button\_2 = Button(self.root, text=2, width=5, command=lambda: self.show\_num(2))  self.button\_2.grid(row=1, column=1)  self.button\_3 = Button(self.root, text=3, width=5, command=lambda: self.show\_num(3))  self.button\_3.grid(row=1, column=2)  self.button\_4 = Button(self.root, text=4, width=5, command=lambda: self.show\_num(4))  self.button\_4.grid(row=2, column=0)  self.button\_5 = Button(self.root, text=5, width=5, command=lambda: self.show\_num(5))  self.button\_5.grid(row=2, column=1)  self.button\_6 = Button(self.root, text=6, width=5, command=lambda: self.show\_num(6))  self.button\_6.grid(row=2, column=2)  self.button\_7 = Button(self.root, text=7, width=5, command=lambda: self.show\_num(7))  self.button\_7.grid(row=3, column=0)  self.button\_8 = Button(self.root, text=8, width=5, command=lambda: self.show\_num(8))  self.button\_8.grid(row=3, column=1)  self.button\_9 = Button(self.root, text=9, width=5, command=lambda: self.show\_num(9))  self.button\_9.grid(row=3, column=2)  self.button\_0 = Button(self.root, text=0, width=5, command=lambda: self.show\_num(0))  self.button\_0.grid(row=4, column=2)  self.button\_add = Button(self.root, text='+', width=5, command=lambda: self.sign\_num('+'))  self.button\_add.grid(row=4, column=0)  self.button\_add = Button(self.root, text='-', width=5, command=lambda: self.sign\_num('-'))  self.button\_add.grid(row=4, column=1)  self.button\_add = Button(self.root, text='\*', width=5, command=lambda: self.sign\_num('\*'))  self.button\_add.grid(row=5, column=0)  self.button\_add = Button(self.root, text='/', width=5, command=lambda: self.sign\_num('/'))  self.button\_add.grid(row=5, column=1)  self.button\_equal = Button(self.root, text='=', width=5, command=self.equal\_num)  self.button\_equal.grid(row=5, column=2)  self.text1 = Text(self.root, width=18, height=3)  self.text1.grid(row=0, column=0, columnspan=3)  def show\_num(self, num):  if self.if\_equal:  self.text1.delete(1.0, END)  self.if\_equal = False  self.text1.insert(2.0, num)  def sign\_num(self, sign):  self.sign = sign  self.num1 = self.text1.get(1.0, END)  self.text1.delete(1.0, END)  def equal\_num(self):  self.if\_equal = True  self.num2 = self.text1.get(1.0, END)  self.text1.delete(1.0, END)  count = 0  if self.sign == '+':  count = float(self.num1) + float(self.num2)  elif self.sign == '-':  count = float(self.num1) - float(self.num2)  elif self.sign == '\*':  count = float(self.num1) \* float(self.num2)  else:  count = float(self.num1) / float(self.num2)  self.text1.insert(1.0, count)  if \_\_name\_\_ == '\_\_main\_\_':  root = Tk()  c = Calc(root)  root.mainloop |



1. 设计一个数据库，包括三个数据表：学生表（Student）、课程表（Course）、学生选课表（stuCour）。学生表包含字段：学号（主键），姓名，性别，年级，专业。课程表：课程号（主键），课程名，学时数。学生选课表：学号，课程号，开课时间。设计图形界面实现：
2. 添加/删除/查询学生信息；
3. 添加/删除/查询课程信息；
4. 学生选课/退课。

|  |
| --- |
| import sqlite3  from tkinter import \*  from tkinter import simpledialog  from tkinter import messagebox as msgbox  def opendb():  stuconn = sqlite3.connect("C:/学生选课.db")  stucur = stuconn.execute(  """create table if not exists students(id integer primary key,studentsName varchar(128),sex varchar(10),age integer ,zhuanye varchar(128))""")  coursecur = stuconn.execute(  """create table if not exists courses(cid integer primary key,courseName varchar(128),time varchar(128))""")  coustucur = stuconn.execute(  """create table if not exists coustus(csid integer ,cid varchar(128),starttime varchar(128))""")  return stucur, stuconn, coursecur, coustucur  def intoid():  r = simpledialog.askstring("学号", "请输入学号:")  return r  def intoname():  r = simpledialog.askstring("姓名", "请输入姓名:")  return r  def intosex():  r = simpledialog.askstring("性别", "请输入性别:")  return r  def intoage():  r = simpledialog.askstring("年龄", "请输入年龄:")  return r  def intozhuanye():  r = simpledialog.askstring("专业", "请输入专业:")  return r  def adddb():  person = []  person.append(intoid())  person.append(intoname())  person.append(intosex())  person.append(intoage())  person.append(intozhuanye())  hel = opendb()  hel[1].execute("insert into students(id,studentsName,sex,age ,zhuanye)values(?,?,?,?,?)",  (person[0], person[1], person[2], person[3], person[4]))  hel[1].commit()  listb.insert(END, person)  hel[1].close()  def deldb():  delchoice = simpledialog.askstring("学号", "请输入要删除的学号:")  hel = opendb()  if hel[1].execute("delete from students where id=" + delchoice).rowcount >0:  listb.delete(0, listb.size())  list1 = hel[0].execute("select \* from students")  for item in list1:  listb.insert(END, item)  else:  msgbox.showwarning("Warning","没有查找到此学生")  hel[1].commit()  hel[1].close()  def sea():  seachoice = simpledialog.askstring("学号", "请输入要查询的学号:")  hel = opendb()  cur = hel[1].cursor()  cur.execute("select\*from students where id=" + seachoice)  hel[1].commit()  for row in cur:  str1 = "学号:" + str(row[0]) + "\n姓名:" + str(row[1]) + "\n性别:" + str(row[2]) + "\n年级:" + str(  row[3]) + "\n专业:" + str(row[4])  msgbox.showinfo("学生信息", str1)  cur.close()  hel[1].close()  def intocid():  r = simpledialog.askstring("课程号", "请输入课程号:")  return r  def intocoursename():  r = simpledialog.askstring("课程名", "请输入课程名:")  return r  def intotime():  r = simpledialog.askstring("学时数", "请输入学时数:")  return r  def addcourse():  person = []  person.append(intocid())  person.append(intocoursename())  person.append(intotime())  hel = opendb()  hel[1].execute("insert into courses(cid,courseName ,time )values(?,?,?)",  (person[0], person[1], person[2]))  hel[1].commit()  hel[1].close()  listb1.insert(END, person)  def delcourse():  delchoice = simpledialog.askstring("课程号", "请输入要删除的课程号:")  hel = opendb()  hel[1].execute("delete from courses where cid=" + delchoice)  listb1.delete(0, listb.size())  list1 = hel[0].execute("select \* from courses")  for item in list1:  listb1.insert(END, item)  hel[1].commit()  hel[1].close()  def seacourse():  seachoice = simpledialog.askstring("课程号", "请输入要查询的课程号:")  hel = opendb()  cur = hel[1].cursor()  cur.execute("select\*from courses where cid=" + seachoice)  hel[1].commit()  for row in cur:  str1 = "课程号:" + str(row[0]) + "\n课程名:" + str(row[1]) + "\n学时数:" + str(row[2])  msgbox.showinfo("课程信息", str1)  cur.close()  hel[1].close()  def intostarttime():  r = simpledialog.askstring("开课时间", "请输入开课时间:")  return r  def addcurstu():  person = []  person.append(intoid())  person.append(intocid())  person.append(intostarttime())  hel = opendb()  hel[1].execute("insert into coustus(csid ,cid ,starttime )values(?,?,?)",  (person[0], person[1], person[2]))  hel[1].commit()  hel[1].close()  listb2.insert(END, person)  def delcoustu():  delchoice = simpledialog.askstring("学号号", "请输入要删除的学号号:")  choice = simpledialog.askstring("课程号", "请输入要删除的课程号:")  hel = opendb()  hel[1].execute("delete from coustus where csid={} and cid={}".format(delchoice, choice))  listb2.delete(0, listb.size())  list1 = hel[0].execute("select \* from coustus")  for item in list1:  listb2.insert(END, item)  hel[1].commit()  hel[1].close()  if \_\_name\_\_ == "\_\_main\_\_":  flag = 1  root = Tk()  root.title("学生系统")  hel = opendb()  cur = hel[1].cursor()  cur1 = hel[1].cursor()  cur2 = hel[1].cursor()  cur.execute("select \* from students")  cur1.execute("select \* from coustus")  cur2.execute("select \* from courses")  ff = LabelFrame(root, width="100", text="学生")  ff.pack(side=LEFT)  listb = Listbox(ff)  for item in cur:  listb.insert(END, item)  listb.pack()  b1 = Button(ff, text="添加", command=adddb, height=2, width=30)  b2 = Button(ff, text="删除", command=deldb, height=2, width=30)  b3 = Button(ff, text="查询", command=sea, height=2, width=30)  b1.pack()  b2.pack()  b3.pack()  ff2 = LabelFrame(root, text="学生选课")  ff2.pack(side=RIGHT)  listb2 = Listbox(ff2)  for item in cur1:  listb2.insert(END, item)  listb2.pack()  b4 = Button(ff2, text="学生选课", command=addcurstu, height=3, width=30)  b5 = Button(ff2, text="学生退课", command=delcoustu, height=3, width=30)  b4.pack()  b5.pack()  ff1 = LabelFrame(root, text="课程")  ff1.pack()  listb1 = Listbox(ff1)  for item in cur2:  listb1.insert(END, item)  listb1.pack()  bb1 = Button(ff1, text="添加", command=addcourse, height=2, width=30)  bb2 = Button(ff1, text="删除", command=delcourse, height=2, width=30)  bb3 = Button(ff1, text="查询", command=seacourse, height=2, width=30)  bb1.pack()  bb2.pack()  bb3.pack()  cur2.close()  hel[1].close()  root.mainloop() |



**（三）实验结果**

要求提交word文档，文档命名规则：姓名+学号+实验次数.doc

内容包含：

* 实验题目
* 源代码
* 运行结果截屏。