**第十二章 图形用户界面**

**姓名**：胡浩龙，**学号**：2018050576

**第一题**：参照例12.4创建grid集合布局程序2。

**源代码**：

from tkinter import \*

root=Tk();root.title("登陆")

Label(root,text="用户名").grid(row=0,column=0)

Entry(root).grid(row=0,column=1,columnspan=2)

Label(root,text="密码").grid(row=1,column=0)

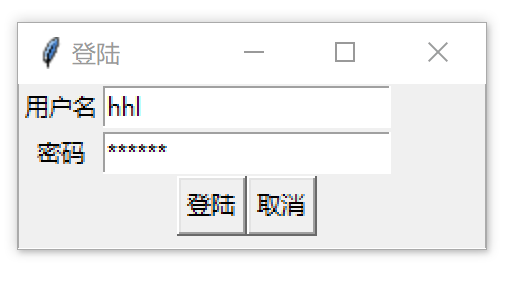
Entry(root,show="\*").grid(row=1,column=1,columnspan=2)

Button(root,text="登陆").grid(row=3,column=1,sticky=E)

Button(root,text="取消").grid(row=3,column=2,sticky=W)

root.mainloop()

**输出结果：**



**第二题**：参照例12.5创建grid集合布局程序，实现按钮布局界面。

**源代码**：

from tkinter import \*

root=Tk()

Button(root,text="1").grid(row=0,column=0)

Button(root,text="2").grid(row=0,column=1)

Button(root,text="3").grid(row=0,column=2)

Button(root,text="4").grid(row=1,column=0)

Button(root,text="5").grid(row=1,column=1)

Button(root,text="6").grid(row=1,column=2)

Button(root,text="7").grid(row=2,column=0)

Button(root,text="8").grid(row=2,column=1)

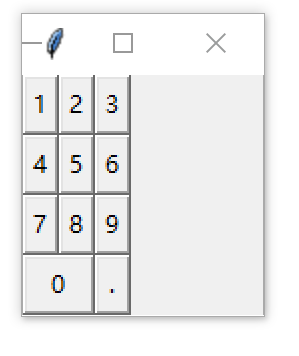
Button(root,text="9").grid(row=2,column=2)

Button(root,text="0").grid(row=3,column=0,columnspan=2,sticky=E+W)

Button(root,text=".").grid(row=3,column=2,sticky=E+W)

root.mainloop()

**输出结果：**

****

**第三题**：参照例12.11利用Label和Button组件，创建简易图片浏览器程序。

**源代码**：

import tkinter as tk, os

class Application(tk.Frame):

def \_\_init\_\_(self, master=None):

self.files = os.listdir(r'd:\pic')

self.index = 0

self.img = tk.PhotoImage(file=r'd:\pic' + '\\' + self.files[self.index])

tk.Frame.\_\_init\_\_(self, master)

self.pack()

self.createWidgets()

def createWidgets(self):

self.lblImage = tk.Label(self, width=300, height=300)

self.lblImage['image'] = self.img

self.lblImage.pack()

self.f = tk.Frame()

self.f.pack()

self.btnPrev = tk.Button(self.f, text='上一张', command=self.prev)

self.btnPrev.pack(side=tk.LEFT)

self.btnNext = tk.Button(self.f, text='下一张', command=self.next) #创建按钮组件

self.btnNext.pack(side=tk.LEFT)

def prev(self): self.showfile(-1)

def next(self): self.showfile(1)

def showfile(self, n): self.index += n

if self.index < 0: self.index = len(self.files) - 1

if self.index > len(self.files) - 1: self.index = 0

self.img = tk.PhotoImage(file=r'd:\pic' + '\\' + self.files[self.index])

self.lblImage['image'] = self.img

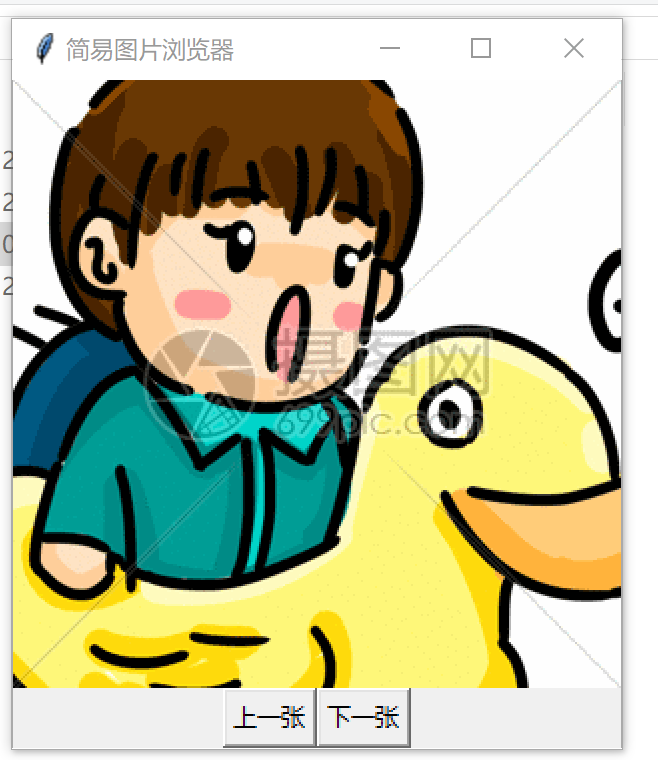
root = tk.Tk()

root.title('简易图片浏览器')

app = Application(master=root)

app.mainloop()

**输出结果：**

****

**第四题**：参照例12.15利用Entry和Text组件，创建用户注册程序。

**源代码**：

import tkinter as tk

from tkinter import messagebox

class Application(tk.Frame):

def \_\_init\_\_(self,master=None):

tk.Frame.\_\_init\_\_(self,master)

self.grid()

self.creatWidgets()

def creatWidgets(self):

self.lblEmail=tk.Label(self,text="用户名")

self.lblPass1=tk.Label(self,text="密码")

self.lblPass2=tk.Label(self,text="确认密码")

self.lblDesc=tk.Label(self,text="自我介绍")

self.lblEmail.grid(row=0,column=0,sticky=tk.E)

self.lblPass1.grid(row=1,column=0,sticky=tk.E)

self.lblPass2.grid(row=2,column=0,sticky=tk.E)

self.lblDesc.grid(row=3,column=0,sticky=tk.NE)

self.entryEmail=tk.Entry(self)

self.entryPass1=tk.Entry(self,show="\*")

self.entryPass2=tk.Entry(self,show="\*")

self.textDesc=tk.Text(self,width=20,height=5)

self.entryEmail.grid(row=0,column=1,columnspan=2)

self.entryPass1.grid(row=1,column=1,columnspan=2)

self.entryPass2.grid(row=2,column=1,columnspan=2)

self.textDesc.grid(row=3,column=1,columnspan=2)

self.btnOK=tk.Button(self,text="注册",command=self.funcOK)

self.btnOK.grid(row=4,column=1,sticky=tk.E)

self.btnCancel=tk.Button(self,text="取消",command=root.destroy)

self.btnCancel.grid(row=4,column=2,sticky=tk.W)

def funcOK(self):

pw1=self.entryPass1.get()

pw2=self.entryPass2.get()

if(pw1==pw2):

str1='欢迎注册\n'

str1+="您的账户为："+self.entryEmail.get()+'\n'

str1+="您的特长为：\n"+self.textDesc.get(0.0,tk.END)

tk.messagebox.showinfo("注册",str1)

else:

str2='两次密码不同，请重新输入！\n'

tk.messagebox.showinfo("Error!",str2)

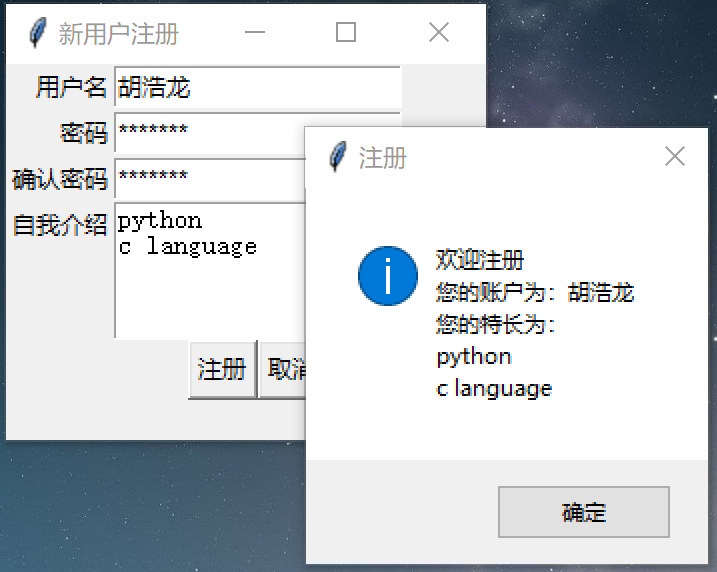
root=tk.Tk()

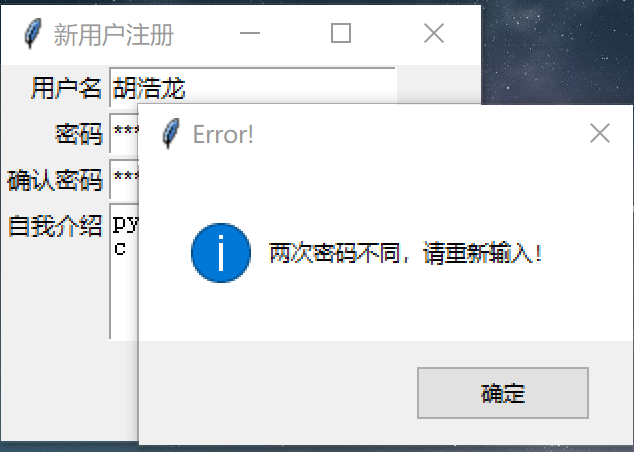
root.title("新用户注册")

app=Application(master=root)

app.mainloop()

**输出结果：**





**第五题**：参照例12.18利用Radibutton和Checkbox组件，创建Questionnaire调查个人信息程序。

**源代码**：

import tkinter as tk

from tkinter import messagebox

class Application(tk.Frame):

def \_\_init\_\_(self,master=None):

tk.Frame.\_\_init\_\_(self,master)

self.grid()

self.creatWidget()

def creatWidget(self):

self.lblTitle=tk.Label(self,text="个人信息调查")

self.lblName=tk.Label(self,text="姓名")

self.lblSex=tk.Label(self,text="性别")

self.lblHobby=tk.Label(self,text="爱好")

self.lblTitle.grid(row=0,column=0,columnspan=4)

self.lblName.grid(row=1,column=0)

self.lblSex.grid(row=2,column=0)

self.lblHobby.grid(row=3,column=0)

#文本框

self.entryName=tk.Entry(self)

self.entryName.grid(row=1,column=1,columnspan=3)

#单选按钮

self.vSex=tk.StringVar()

self.vSex.set('M')

self.radioSexM=tk.Radiobutton(self,text="男",value='M',variable=self.vSex)

self.radioSexF=tk.Radiobutton(self,text="女",value='F',variable=self.vSex)

self.radioSexM.grid(row=2,column=1)

self.radioSexF.grid(row=2,column=2)

#复选框

self.vHobbyMusic=tk.IntVar()

self.vHobbySports=tk.IntVar()

self.vHobbyTravel=tk.IntVar()

self.vHobbyMovie=tk.IntVar()

self.checkboxMusic=tk.Checkbutton(self,text="音乐",variable=self.vHobbyMusic)

self.checkboxSports=tk.Checkbutton(self,text="运动",variable=self.vHobbySports)

self.checkboxTravel=tk.Checkbutton(self,text="旅游",variable=self.vHobbyTravel)

self.checkboxMovie=tk.Checkbutton(self,text="影视",variable=self.vHobbyMovie)

self.checkboxMusic.grid(row=3,column=1)

self.checkboxSports.grid(row=3,column=2)

self.checkboxTravel.grid(row=3,column=3)

self.checkboxMovie.grid(row=3,column=4)

#按钮

self.btnOk=tk.Button(self,text="提交",command=self.funcOk)

self.btnOk.grid(row=4,column=1,sticky=tk.E)

self.btnCancel=tk.Button(self,text="取消",command=root.destroy)

self.btnCancel.grid(row=4,column=2,sticky=tk.W)

def funcOk(self):

strSex='男' if (self.vSex.get()=='M') else '女'

strMusic=self.checkboxMusic['text'] if (self.vHobbyMusic.get()==1) else''

strSports=self.checkboxSports['text'] if (self.vHobbySports.get()==1) else''

strTravel=self.checkboxTravel['text'] if (self.vHobbyTravel.get()==1) else''

strMovie=self.checkboxMovie['text'] if (self.vHobbyMovie.get()==1) else''

str1=self.entryName.get()+'您好:\n'

str1+="您的性别是："+strSex+'\n'

str1+="您的爱好是："+strMusic+' '+strSports+' '+strTravel+' '+strMovie

tk.messagebox.showinfo("个人信息",str1)

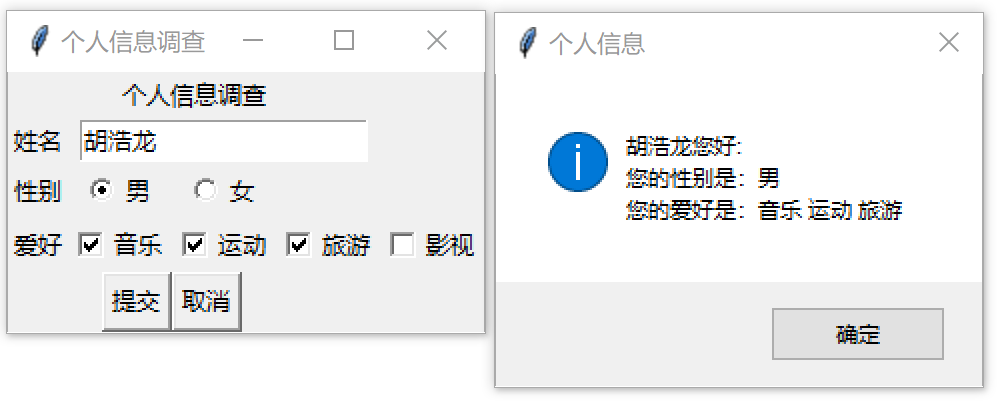
root=tk.Tk()

root.title('个人信息调查')

app=Application(master=root)

app.mainloop()

**输出结果：**

****

**第六题**：参照例12.22利用OptionMenu选择项程序，从组合框中选择字体大小，然后单击“改变字体”按钮，改变标签文本的字体大小。

**源代码**：

import tkinter as tk

class Application(tk.Frame):

def \_\_init\_\_(self,master=None):

tk.Frame.\_\_init\_\_(self,master)

self.grid()

self.createWidgets()

def createWidgets(self):

optionList=range(10,61,4)

self.vFont=tk.StringVar()

self.vFont.set(14)

self.optionMenuFont=tk.OptionMenu(self,self.vFont,\*optionList)

self.optionMenuFont.pack(side=tk.LEFT)

self.buttonFont=tk.Button(self,text="改变字体",command=self.changefont)

self.buttonFont.pack(side=tk.LEFT)

self.lblTitle=tk.Label(self,text="hello",font=('Helvetica',14,'bold'))

self.lblTitle.pack(side=tk.LEFT)

def changefont(self):

fontNew=('Helvetica',self.vFont.get(),'bold')

self.lblTitle.config(font=fontNew)

root=tk.Tk()

root.title('设置字体大小')

root['width']=400

root['height']=50

app=Application(master=root)

app.mainloop()

**输出结果：**

****

**第七题**：参照例12.23创建Scale程序，通过移动滑块，改变字体大小。

**源代码**：

import tkinter as tk

class Application(tk.Frame):

def \_\_init\_\_(self,master=None):

tk.Frame.\_\_init\_\_(self,master)

self.grid()

self.creatWidgets()

def creatWidgets(self):

self.scaleFont=tk.Scale(self,from\_=10,to=60,length=400,orient=tk.HORIZONTAL,command=self.changefont)

self.scaleFont.set(20)

self.scaleFont.pack()

self.lblTitle=tk.Label(self,text="Hello",font=('Helvetica',20,'bold'))

self.lblTitle.pack()

def changefont(self,value):

fontNew=('Helvetica',self.scaleFont.get(),'bold')

self.lblTitle.config(font=fontNew)

root=tk.Tk()

root.title('设置字体大小')

root['width']=400

root['height']=50

app=Application(master=root)

app.mainloop()

**输出结果：**

****

**第八题**：参照例12.28创建通用对话框应用程序，实现简易文本编辑器。

**源代码**：

import tkinter as tk

import tkinter.scrolledtext as tst

from tkinter import filedialog

from tkinter import colorchooser

class Application(tk.Frame):

def \_\_init\_\_(self,master=None):

tk.Frame.\_\_init\_\_(self,master)

self.grid()

self.creatWidgets()

def creatWidgets(self):

self.textEdit=tst.ScrolledText(self,width=80,height=20)

self.textEdit.grid(row=0,column=0,rowspan=6)

self.btnOpen=tk.Button(self,text="打开",command=self.funcOpen)

self.btnOpen.grid(row=1,column=1)

self.btnSave=tk.Button(self,text="保存",command=self.funcSave)

self.btnSave.grid(row=2,column=1)

self.btnColor=tk.Button(self,text="颜色",command=self.funcColor)

self.btnColor.grid(row=3,column=1)

self.btnQuit=tk.Button(self,text="退出",command=self.funcQuit)

self.btnQuit.grid(row=4,column=1)

def funcOpen(self):

self.textEdit.delete(1.0,tk.END)

fname=tk.filedialog.askopenfilename(filetypes=[('python源文件','.py')])

with open(fname,'r',encoding='UTF-8') as f1:

str1=f1.read()

self.textEdit.insert(0.0,str1)

def funcSave(self):

str1=self.textEdit.get(1.0,tk.END)

fname=tk.filedialog.asksaveasfilename(filetypes=[('python源文件','.py')])

with open(fname,'w',encoding='UTF-8') as f1:

f1.write(str1)

def funcColor(self):

t,c=tk.colorchooser.askcolor(title='askcolor')

self.textEdit.config(bg=c)

def funcQuit(self):

root.destroy()

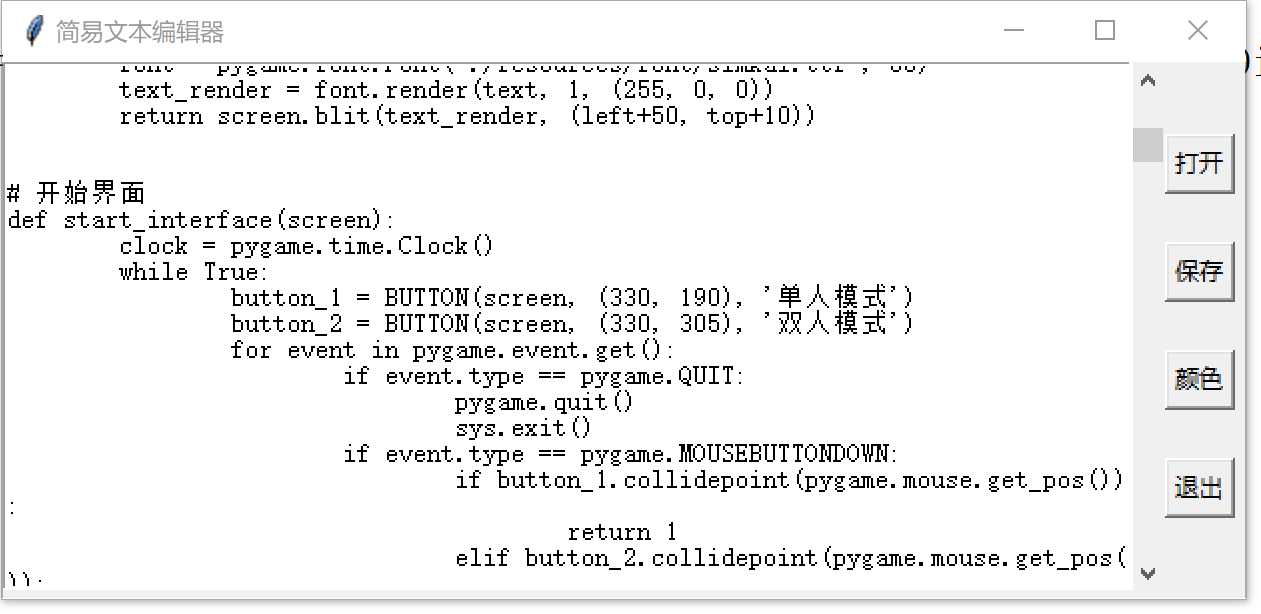
root=tk.Tk()

root.title('简易文本编辑器')

app=Application(master=root)

app.mainloop()

**输出结果：**

****

**第九题**：参照例12.33创建简单的文本编辑器程序。

**源代码**：

import tkinter as tk

import tkinter.scrolledtext as tst

from tkinter import messagebox

from tkinter import filedialog

class Application(tk.Frame):

def \_\_init\_\_(self,master=None):

tk.Frame.\_\_init\_\_(self,master)

self.grid()

self.creatWidgets()

self.creatMenu()

root['menu']=self.menubar

root.bind('<Button-3>',self.f\_popup)

def creatWidgets(self):

self.textEdit=tst.ScrolledText(self,width=80,height=20)

self.textEdit.grid(row=0,column=0,rowspan=6)

def creatMenu(self):

#创建子菜单

self.menubar=tk.Menu(root)

self.menufile=tk.Menu(self.menubar)

self.menuedit=tk.Menu(self.menubar,tearoff=0)

self.menuhelp=tk.Menu(self.menubar,tearoff=0)

self.menubar.add\_cascade(label='File',menu=self.menufile)

self.menubar.add\_cascade(label='Edit',menu=self.menuedit)

self.menubar.add\_cascade(label='Help',menu=self.menuhelp)

#添加菜单项

self.menufile.add\_command(label='New',command=self.f\_new)

self.menufile.add\_command(label='Open',command=self.f\_open)

self.menufile.add\_command(label='Save',accelerator='^A',command=self.f\_save)

self.menufile.add\_separator

self.menufile.add\_command(label='Exit',command=root.destroy)

self.menuedit.add\_command(label='Cut',command=self.f\_cut)

self.menuedit.add\_command(label='Copy',command=self.f\_copy)

self.menuedit.add\_command(label='Paste',command=self.f\_paste)

self.menuhelp.add\_command(label='About',command=self.f\_about)

def f\_new(self):

self.textEdit.delete(1.0,tk.END)

def f\_open(self):

self.textEdit.delete(1.0, tk.END)

fname = tk.filedialog.askopenfilename(filetypes=[('Python源文件', '.py')])

with open(fname, 'r', encoding='utf-8') as f1:

str1 = f1.read()

self.textEdit.insert(0.0, str1)

def f\_save(self):

str1 = self.textEdit.get(1.0, tk.END)

fname = tk.filedialog.asksaveasfilename(filetypes=[('Python源文件','.py')])

with open(fname, 'w', encoding='utf-8') as f1:

f1.write(str1)

def f\_about(self):

tk.messagebox.showinfo('关于','版本V 1.0.1')

def f\_cut(self):

try:

str1 = self.textEdit.get(tk.SEL\_FIRST, tk.SEL\_LAST)

self.textEdit.clipboard\_clear()

self.textEdit.clipboard\_append(str1)

self.textEdit.delete(tk.SEL\_FIRST, tk.SEL\_LAST)

except:

pass

def f\_copy(self):

try:

str1 = self.textEdit.get(tk.SEL\_FIRST, tk.SEL\_LAST)

self.textEdit.clipboard\_clear()

self.textEdit.clipboard\_append(str1)

except:

pass

def f\_paste(self):

str1 = self.textEdit.selection\_get(selection='CLIPBOARD')

try:

self.textEdit.replace(tk.SEL\_FIRST, tk.SEL\_LAST, str1)

except:

self.textEdit.insert(tk.INSERT, str1)

def f\_popup(self, event):

self.menuedit.post(event.x\_root, event.y\_root)

root=tk.Tk()

root.title('简易文本编辑器')

app=Application(master=root)

app.mainloop()

**输出结果：**

