第八章
图形用户界面设计

早期的计算机工作界面就如现在的“控制台应用”一般，并无图形用户界面，甚至不是多线程多任务的工作环境。计算机处理器技术和显示技术的发展，推动了图形用户界面（Graphical User Interface）的蓬勃发展。Windows视窗操作系统就是一个典型的图形用户界面系统，图形用户界面使计算机操作变得友好且直观。

第八章 图形用户界面设计

图形用户界面按其作用不同主要分为两类：

其一，作为程序的控制和显示界面而存在的图形用户界面，例如，窗体和控件等；

其二，作为计算结果而存在的图形用户界面，例如，画布和图形等。

Python语言中，实现图形用户界面设计的包称为Tkinter，而且Tkinter是Python自带的一个标准GUI包。本章主要介绍借助于Tkinter进行图形用户界面设计的技巧。

本章将分别用六节来介绍借助于Tkinter进行图形用户界面设计的技巧：

8.1 视窗设计

8.2 界面布局设计

8.3 “复数计算器”程序算法设计

8.4 常用控件

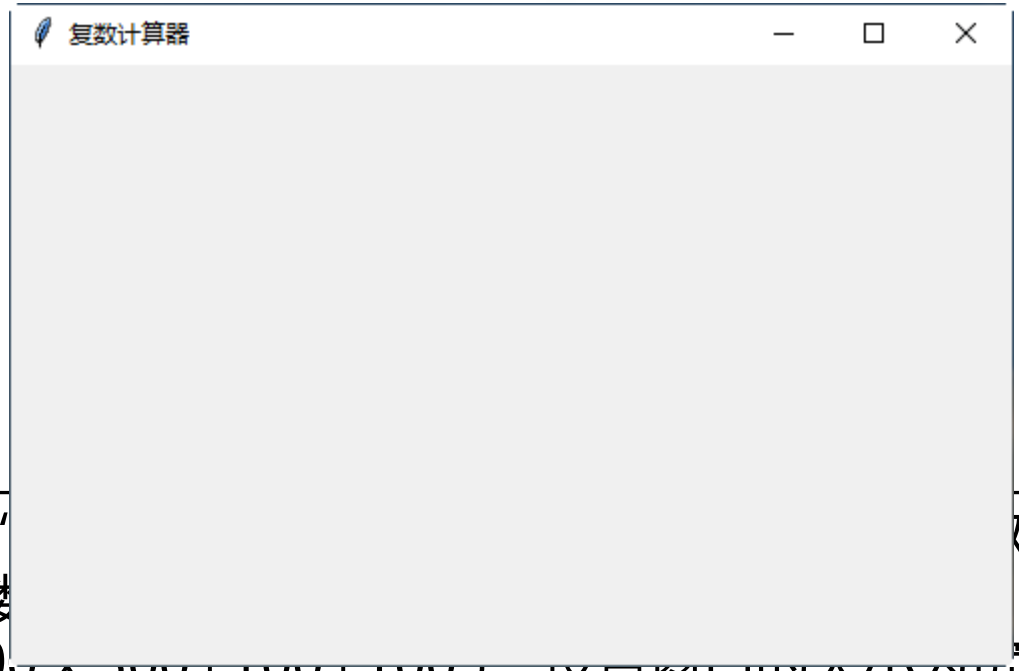
8.5 画布与绘图技术

8.6 事件绑定与自定义事件

8.1 视窗设计

下面是使用Tkinter进行图形用户界面设计，创建一个空的窗体的程序：

```
1    import tkinter as tk
2    if __name__=='__main__':
3        mainform=tk.Tk()
4        mainform.title('复数计算器')
5        mainform.geometry('500x300+100+100')
6        mainform.mainloop()
```



第1行装载tkinter包，并赋以别名tk。第3行“ 对象，赋给mainform。第4行“mainform.title('复数 复数计算器”。第5行“mainform.geometry('500x300+100+100')”设置窗口的大小为宽500像素点、高300像素点、窗口左上角(x,y)为（100，100），大小格式为“宽度x高度+左上角x坐标+左上角y坐标”（其中，“宽度x高度”中间的“x”为小写的字母x）。第6行“mainform.mainloop()”调用mainloop函数启动窗口，等待系统发送事件。

8.1 视窗设计

图形用户界面程序的设计方法为：

（1）设为主窗口，并为主窗口设为标题（和图标）；

（2）在主窗口上放置各类控件，有些控件本身是容器类控件（用于摆放其他控件），然后，为这些控件编写事件响应程序；

（3）启动窗口后，等待操作系统向窗口发送消息或触发特定的事件，程序收到事件后执行相应的任务。

因此，图形用户界面程序启动后，一直处于等待外部事件和输入（包括键盘和鼠标输入等）的状态，直接收到外部输入（事件），将执行相应的功能。

8.1 视窗设计

接下来的实例，将在类的基础上实现空窗体的创建（功能与上一个实例相同）：

```
1    import tkinter as tk
2    class MainForm(tk.Tk):
3        def __init__(self):
4            super().__init__()
5            self.title('复数计算器')
6            self.geometry('500x300+100+100')
7    if __name__=='__main__':
8        mainform=MainForm()
9        mainform.mainloop()
```

第1行装载包tkinter，并赋以别名tk。

第2~6行定义类MainForm，继承了父类tk.Tk，MainForm作为主窗口控件（也称主窗体）。第3~6行为类MainForm的构造方法。第4行“super().__init__()”调用父类的构造方法；第5行“self.title('复数计算器')”设置主窗口的标题为“复数计算器”。第6行“self.geometry('500x300+100+100')”设置主窗口的大小为500×300，左上角的坐标为(100, 100)，这里的“+100+100”可以省略，由系统确定窗口的位置。

第8行“mainform=MainForm()”调用MainForm类定义窗口对象mainform。第9行“mainform.mainloop()”启动图形用户界面程序，等待用户输入或系统事件。

8.1 视窗设计

视窗常用的方法还有：

（1）configure

用于设置窗口样式，其中最常用的[illegible]窗口背景色，例如：

mainform.configure(bg='lig[illegible]

将在上述实例代码中插入该语[illegible]背景设为淡蓝色。

（2）iconbitmap

用于设置窗口的图像，例如：

mainform.iconbitmap('fly.ico')

注意：图标文件的扩展名为.ico。

在当前的工程目录下要有图标fly.ico文件（可使用HyperSnap-7抓图软件随意抓[illegible]，在前面实例[illegible]象。

（3）resizable

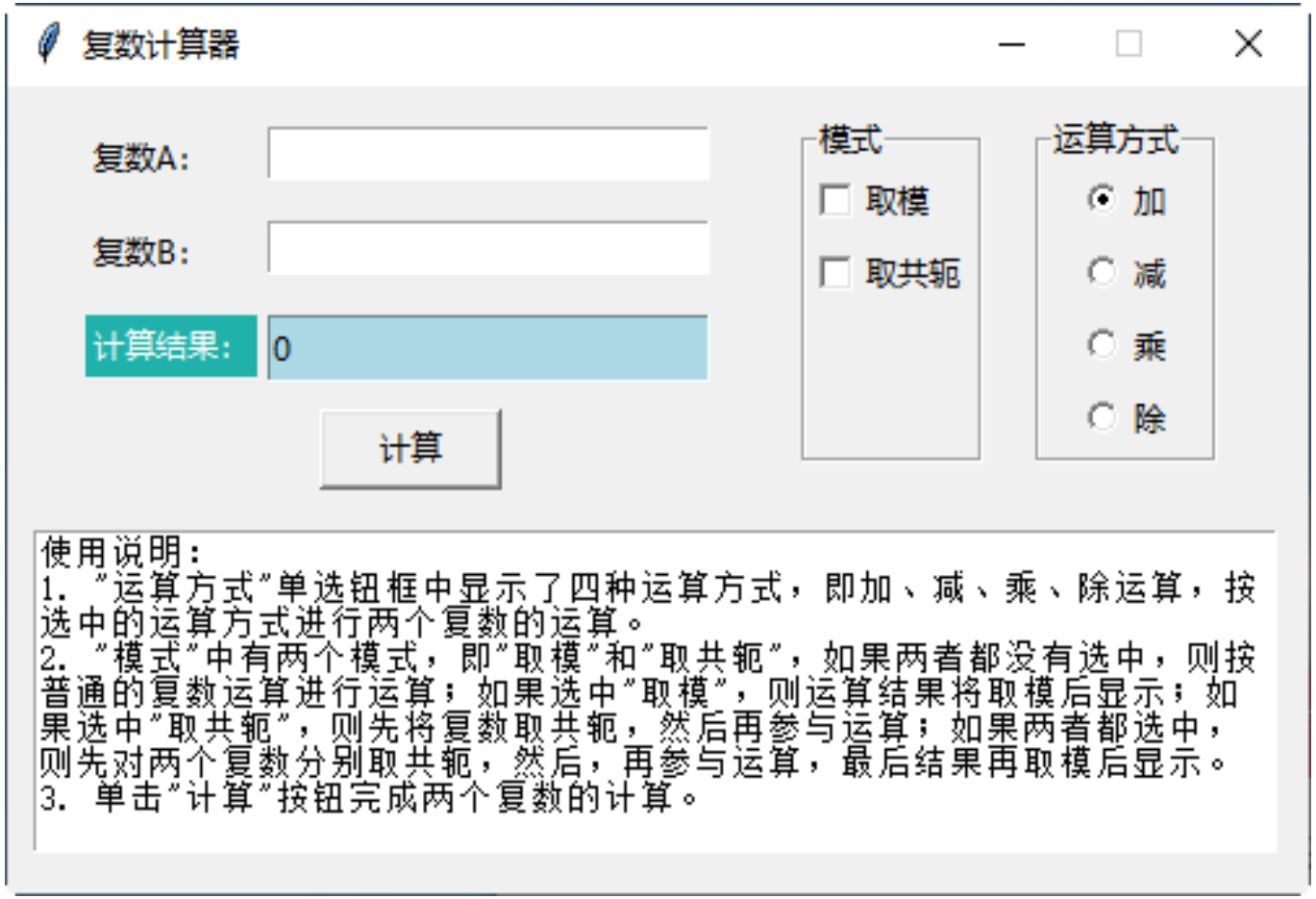
用于设置窗口是否可调整大小，具有两个参数，第一个参数为真，则窗口宽度可调大小，为假，则窗口宽度大小不可调；第二参数为真，则窗口高度可调节大小，为假，则高度不可调节大小。例如，

mainform.resizable(False,False)

在前面的实例中插入上述语句，即可将窗口大小设置为不可调节。

8.2 界面布局设计

现在拟在前两个实例的基础上，设计一个“复数计算器”，其界面设计如下图：



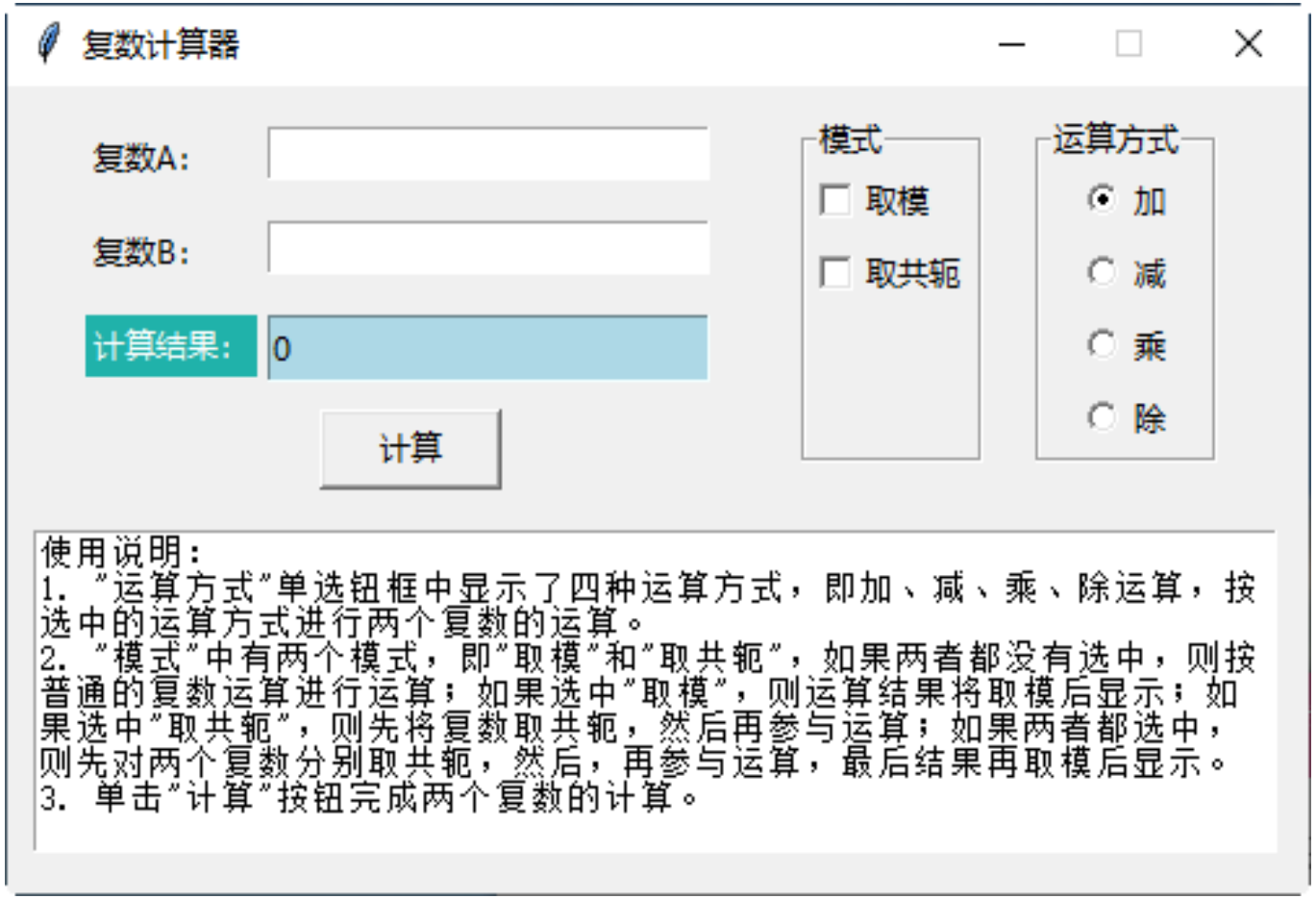
8.2 界面布局设计

首先来了解复数计算器工作界面包含的控件：

序号	控件名	类名	属性及内容
1	mainform	MainForm	继承 tk.Tk 类，为主窗口，标题为“复数计算器”
2	label1	Label	静态文本框（也可以显示图像），显示“复数 A：”
3	stxt1	Entry	（单行 入，输
4	label2	Label	静态文 B：”
5	stxt2	Entry	（单行 入，输
6	label3	Label	静态文 结果：
7	stxt3	Entry	（单行 显示两
8	checkboxgroup	LabelFrame	带标签 选框，这两个复选框的值保存在 checkboxvalue 列表中
9	cb	Checkbutton	复选框控件，用于 for 结构中，共有 2 个复选框
10	radiogroup	LabelFrame	带标签的框架控件，用作容器，存放四个单选钮（也称收音机控件），这四个单选钮只有一个处于选中状态，其值保存在 radiovalue 中
11	rb	Radiobutton	单选钮控件，用于 for 结构中，共有 4 个单选钮
12	btn1	Button	命令按钮，接收用户的单击事件后，将执行 btn1cal 方法
13	btxt	Text	多行文本框控件，设为“不使能”状态，即不接收用户入，这里用于显示程序的功能和使用方法

8.2 界面布局设计

现在拟在前两个实例的基础上，设计一个“复数计算器”，其界面设计如下图：



8.2 界面布局设计

复数计算器实现代码（第一部分）：

```
1       import tkinter as tk
2       class MainForm(tk.Tk):
3           def __init__(self):
4               super().__init__()
5               self.title('复数计算器')
6               self.geometry('500x300+100+100')
7               self.resizable(False,False)
8               self.myinitgui()
```

第1行装载包tkinter，并赋以别名tk。

第2~6行定义类MainForm，继承了父类tk.Tk，MainForm作为主窗口控件（也称主窗体）。

第3~6行为类MainForm的构造方法。第4行“super().__init__()”调用父类的构造方法；第5行“self.title('复数计算器')”设置主窗口的标题为“复数计算器”。第6行“self.geometry('500x300+100+100')”设置主窗口的大小为500×300，左上角的坐标为(100, 100)，这里的“+100+100”可以省略，由系统确定窗口的位置。

第7行设置窗口大小不可调整。

第8行调用myinitgui方法设置窗口内的各个控件。

8.2 界面布局设计

（第二部分）：

```
10          def myinitgui(self):
11              self.label1=tk.Label(self,text='复
12              self.label1.place(x=30,y=15)
13              self.txt1 = tk.StringVar()
14              self.stxt1=tk.Entry(self,textvaria
15              self.stxt1.place(x=100,y=15,widt
16
17              self.label2 = tk.Label(self, text='
18              self.label2.place(x=30, y=50)
19              self.txt2 = tk.StringVar()
20              self.stxt2 = tk.Entry(self, textvar
21              self.stxt2.place(x=100, y=50, wi
```

定义myinitgui方法，用于布局窗口中的各个控件。这里将各个控件均设为self的成员。

第11~15行为创建静态文本框控件，显示“复数A：”；调用控件label1的place方法，将控件放置在窗口内部坐标(30,15)处，因为语窗口大小被锁定，所以这里使用了绝对坐标和绝对大小。接着定义tk模块的字符串变量对象，用于编辑框Entry或静态文本框Label中的显示内容。创建编辑框对象stxt1，并显示txt1中的内容，用于输入第1个复数。最后设置stxt1编辑框对象的显示位置、宽度和高度。

用相同的方法创建“复数B”相关控件。

8.2 界面布局设计

（第三部分）：

```
23          self.label3 = tk.Label(self, text='计算结果：',
24                                 background='#20B2AA',foreground='white')
25          self.label3.place(x=30, y=85)
26          self.txt3 = tk.StringVar(value='0')
27          self.stxt3 = tk.Entry(self, textvariable=self.txt3,readonlybackground='lightblue')
28          self.stxt3.configure(state='readonly')
29          self.stxt3.place(x=100, y=85, width=170, height=25)
30          self.txt3.set(str(0))
```

第23~24行定义静态文本框label3，显示内容为“计算结果：”，“background= '#20B2AA',”用于设定背景色，颜色的表示为“#RRGGBB”。“foreground = 'white'”用于设定前景色为白色，可以使用系统能识别的颜色英文单词，例如，blue、red等，可以缩写为“fg= 'white'”。第25行调用label3的place方法放置静态文本框。第26行定义tk模块的字符串变量对象txt3，用于保存编辑框（或静态文本框）中的内容。第27行定义编辑框stxt3，显示内容为txt3，只读的背景为浅蓝色。第28行设置stxt3编辑框为只读控件。第29行在窗口中放置stxt3控件。第30行“self.txt3.set(str(0))”设置显示内容为0。

8.2 界面布局设计

（第四部分）：

```
32          self.checkboxgroup=tk.LabelFrame(self,text='模式')
33          self.checkboxgroup.place(x=305,y=10,width=70,height=130)
34          self.checkboxvalue=[tk.IntVar(value=0),tk.IntVar(value=0)]
35          self.checkboxtitle=['取模','取共轭']
36          for i in [0,1]:
37              cb=tk.Checkbutton(self.checkboxgroup,text=self.checkboxtitle[i],
38                                variable=self.checkboxvalue[i])
39              cb.pack(anchor=tk.W)
```

第32行定义带标签的框架checkboxgroup，用作两个复选框的容器。第33行在窗口中放置框架checkboxgroup。第34行定义两个复选框的数据列表checkboxvalue，每个复选框的数据为tk模块的IntVar对象（称为整型变量对象）。第35行“定义列表checkboxtitle作为两个复选框显示的内容。第36~39行为一个for结构，循环两次设置两个复选框的标题和内容。第39行“cb.pack(anchor=tk.W)”调用cb的pack方法放置复选框，这里的参数“anchor=tk.W”表示左对齐，“W”是“West”的首字母。

8.2 界面布局设计

（第五部分）：

```
41          self.radiogroup=tk.LabelFrame(self,text='运算方式')
42          self.radiogroup.place(x=395,y=10,width=70,height=130)
43          self.radiovalue=tk.IntVar()
44          self.radiovalue.set(1)
45          for e,n in [('加',1),('减',2),('乘',3),('除',4)]:
46              rb=tk.Radiobutton(self.radiogroup,text=e,variable=self.radiovalue,value=n)
47              rb.pack(anchor=tk.CENTER)
```

第41行定义一个带标签“运算方式”的框架radiogroup，用作四个单选钮的容器。第42行放置框架radiogroup。四个单选钮为一组，每次只能有一个被选中，需要为这四个单选钮设定一个共享的取值对象，设定一个radiovalue对象。第44行设定radiovalue的值为1，由第45~47行可知，四个单选钮的值（value）依次为1、2、3、4，所以，这里radiovalue的值为1表示value值为1的单选钮被选中。第45~47行为一个for结构，用于设置四个单选钮的显示内容和（选中时的）返回值。第47行将每个单选钮居中放置。

8.2 界面布局设计

（第六部分）：

```
49          self.btn1=tk.Button(self,text='计算',command=self.btn1cal)
50          self.btn1.place(x=120,y=120,width=70,height=30)
```

第49行定义命令按钮btn1，显示“计算”，当按下该按钮时，调用函数btn1cal。这种按下按钮将调用的函数，称为“回调函数”，这是因为函数定义好后，在系统中“注册”了，所谓“注册”是指该函数与特定的事件相关联，例如，使一个函数与鼠标的左键被按下相关联，这样，当系统识别到鼠标左键被按下时，将调用相关的被“注册”好的函数，这个过程必须有操作系统的干预，因此，称为“回调”。这里的btn1cal函数也是“回调函数”，表面上没有调用入口，实质上是由操作系统认别到btn1被单击时，操作系统调用的。这里的“command=self.btn1cal”执行的是“注册”函数的过程。

第50行摆放命令按钮btn1。

8.2 界面布局设计

（第七部分）：

```
52          self.btxt=tk.Text(self)
53          self.btxt.place(x=10,y=165,width=478,height=120)
54          self.btxt.insert(tk.INSERT,'使用说明：\n')
55          self.btxt.insert(tk.INSERT,'1."运算方式"单选钮框中显示了四种运算方式，'
56                                     '即加、减、乘、除运算，
57                                     '进行两个复数的运算。\
58          self.btxt.insert(tk.INSERT,'2."模式"中有两个模式，
59                                     '如果两者都没有选中，则
60                                     '进行运算；如果选中"取
61                                     '取模后显示；如果选中"
62                                     '数取共轭，然后再参与运
63                                     '则先对两个复数分别取共
64                                     '运算，最后结果再取模后显示。\n')
65          self.btxt.insert(tk.INSERT,'3. 单击"计算"按钮完成两个复数的计算。')
66          self.btxt.configure(state='disabled')
```

第52行定义文本框btxt。

第53行放置文本框btxt。

第54~65行向文本框btxt内写入显示内容，其中的“INSERT”表示从当前光标位置插入文本。

第66行将btxt设为“不使能”（即不可编辑、不接收光标（或焦点））。

8.2 界面布局设计

（第八部分）：

```
68          def btn1cal(self):
69              pass
70
71      if __name__=='__main__':
72          mainform=MainForm()
73          mainform.mainloop()
```

第68~69行定义方法btn1cal，内容为空。第72行“mainform=MainForm()”创建MainForm类的对象mainform。第73行“mainform.mainloop()”调用mainloop方法，使图形用户界面程序进行等待事件（或消息）状态。

图形用户界面程序的最后一条语句一定是调用mainloop方法，将程序的控制权交给操作系统，由操作系统管理用户输入或鼠标按键，并将这个输入转化为事件（或消息），触发图形用户界面程序中的相应控件执行相关的“回调”函数（或方法），并输出执行结果。

8.3 “复数计算器”程序算法设计

在上一个实例的方法myinitgui中，将其中的各个控件均作为self的成员，这是一种标准的设计方法，但由各个控件创建好后本身不需要管理，所以，可以将各个控件设为方法中的局部“变量”，只需要将各个控件的数据相关的对象作为self的成员即可。例如，上一个实例的第14~15行：

```
14          self.stxt1=tk.Entry(self,textvariable=self.txt1)
15          self.stxt1.place(x=100,y=15,width=170,height=20)
```

可以写为：

```
14          stxt1=tk.Entry(self,textvariable=self.txt1)
15          stxt1.place(x=100,y=15,width=170,height=20)
```

进一步可以写为一行，即：

```
tk.Entry(self,textvariable=self.txt1). place(x=100,y=15,width=170,height=20)
```

8.3 “复数计算器”程序算法设计

按照上述方法重新改写了上一个实例。同时，在代码中添加了方法btn1cal的代码，完成了两个复数间的四则运算。

（修改部分）：

```
10          def myinitgui(self):
11              tk.Label(self,text='复数 A：').place(x=30,y=15)
12
13              self.txt1 = tk.StringVar()
14              tk.Entry(self,textvariable=self.txt1).place(x=100,y=15,width=170,height=20)
15
16              tk.Label(self, text='复数 B：').place(x=30, y=50)
17
18              self.txt2 = tk.StringVar()
19              tk.Entry(self, textvariable=self.txt2).place(x=100, y=50, width=170, height=20)
```

8.3 “复数计算器”程序算法设计

下面是修改btn1 cal()方法的定义。

```
66      def btn1cal(self):
67          try:
68              a=complex(self.txt
69              b=complex(self.txt
70          except Exception as e:
71              print(e)
72          else:
73              if self.checkboxval
74                  a=a.real-a.ima
75                  b=b.real-b.ima
76              match self.radioval
77                  case 1:
78                      c = a + b
79                  case 2:
80                      c=a-b
81                  case 3:
82                      c=a*b
83                  case 4:
84                      c=a/b
85                  case _:
86                      c=a+b
```

第66~90行为方法btn1cal，使用了try-except-else结构。try部分为第68~69行，第68行读取txt1编辑框中的数据并转化为复数，保存在a中，这里的“get”方法用于获取控件的内容。第69行读取txt2编辑框中的数据并转化为复数，保存在b中。try部分监督第68~69行的代码，如果遇到异常则执行第70~71行，第71行“print(e)”在命令行窗口输入异常提示信息。注意：在图形用户界面程序下，这个异常输出不显示；在使用PyCharm运行模式下，若有异常输出，可以PyCharm的命令行窗口中查看异常。

如果第68~69行的输入正常，则执行第73~90行。第73~75行为一个if结构，表示如果“取共轭”复选框选中（第73行返回1），则将a和b取共轭。第76~86行为一个match多分支结构，根据单选钮的状态，分别计算a和b的和、差、积或商。

8.3 “复数计算器”程序算法设计

```
88                      if self.checkboxvalue[0].get()==1:
89                          c=abs(c)
90                      self.txt3.set(str(c))
91      if __name__=='__main__':
92          mainform=MainForm()
93          mainform.mainloop()
```

第88~89行为一个if结构，表示如果“取模”复选框选中（第88行返回1），则计算c的模。

第90行“self.txt3.set(str(c))”将c的字符串形式赋给txt3。由第25行知，txt3为只读的编辑框stxt3的显示内容。当txt3的值改变后，图形用户界面刷新时，将txt3的新值显示在stxt3控件中。图形用户界面程序的刷新率不是固定的，由操作系统决定，一般，当某个控件的内容变化时，将启动一次显示刷新。

8.4 常用控件

通过前面几个实例的学习，用户可基本上掌握了图形用户界面设计的技巧。本节将用一定量的篇幅介绍一下Tkinter包中的常用控件，并进一步回顾一下曾出现在前面实例中的全部控件。

8.4 常用控件

表中的每个控件都具有众多的参数，在程序设计时，将鼠标移动到控件类名上方暂停一下，将自动弹出该类控件相关的参数。在显示方面，大部分控件都具有参数“bg=”、“fg=”、“text=”、“textvariable=”、“image=”、“relief=”、“anchor=”、“width=”、“height=”和“font=”等，依次表示设置背景色、前景色、显示的本文、显示的内容（可访问）、显示的图像、显示的样式（指3D效果）、位置、宽度、高度和字体样式等。在后续小节中具体介绍各个控件时，将进一步介绍上述参数的具体用法。

序号	控件类	含义
1	Button	命令按钮
2	Label	静态文本框，可显示本文和图片
3	Message	消息框，用于显示多行静态文本
4	messagebox	消息对话框
5	filedialog	文件对话框
6	colorchooser	颜色选择对话框
7	Text	文本控件，可显示多行文本和各种控件，是一个具有强大功能的编辑器
8	Entry	单行编辑框
9	Radiobutton	单选钮，常多个联合使用
10	Checkbutton	复选钮
11	Frame	框架控件，用作其他控件的容器
12	LabelFrame	带标签的框架控件
13	Listbox	列表框控件
14	Scrollbar	滚动条控件
15	Scale	进度条控件
16	Menu	菜单控件
17	Canvas	画布控件

8.4.1 命令按钮

命令按钮为Button类定义的对象，可接受用户的鼠标单击事件，并能调用其参数“command=”指定的回调函数。

```
1    import tkinter as tk
2    import tkinter.messagebox as msgbox
3    class MainForm(tk.Tk):
4        def __init__(self):
5            super().__init__()
6            self.title('控件用法演示')
7            self.geometry('300x100')
8            self.resizable(False,False)
9            self.myinitgui()
```

第3~17行为类MainForm的构造方法。

第5行“super().__init__()”调用父类的构造方法。

第6行设置窗口的标题为“控件用法演示”。

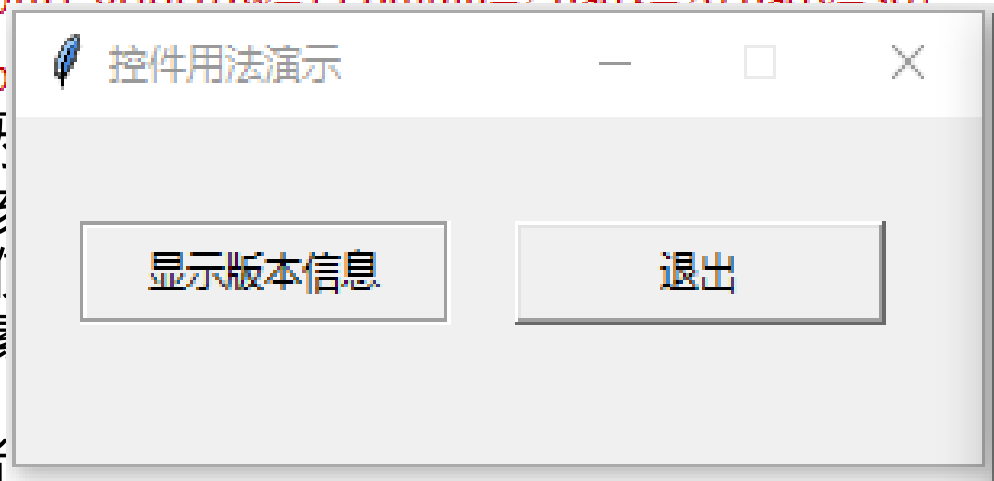
第7行设置窗口的大小为500×300（注意，语句中为小写的字母x），窗口初始位置为屏幕的(100,100)坐标处，表示窗口左上角位于屏幕的(100,100)坐标处。

第8行设置窗口大小不可调整。

第9行调用myinitgui方法设置窗口内的各个控件。

8.4.1 命令按钮

```
10          def myinitgui(self):
11              btn1=tk.Button(self,text='显示版本信息',command=self.btn1fun,
12                             relief='groove',width=15)
13              btn1.grid(row=1,column=2,padx=20,pady=30)
14              b                                                    h=15)
```



第10~15行为
示内容为“显示版
本信息”，单击该
边缘镶嵌形式平铺在
窗口中。第13行
左边界20个像素点，
距上边界30个像素
在的窗口（或控件）
，这里的“self”
第14行定义命
为quit，即退出程序；
显示样式“raised
。注意，grid方法
摆放控件采用了相
了同一行，将在同
一行显示。

第16~17行定义了方法btn1fun，只有一条语句。第17行调用showinfo方法弹出一个消息对话框，其标题为“版本信号”，显示内容为“版本1.0”。

8.4.2 静态文本框

静态文本框为类Label定义的对象，用于输入不可编辑的文本信息，可以输出图像信息。下面的实例就属于静态文本框的应用方法。

```
1       import tkinter as tk
2       import tkinter.messagebox as msgbox
3       class MainForm(tk.Tk):
4           def __init__(self):
5               super().__init__()
6               self.title('控件用法演示')
7               self.geometry('300x240')
8               self.resizable(False,False)
9               self.myinitgui()
10          def myinitgui(self):
11              self.capt=tk.StringVar(value='狗')
12              btn1 = tk.Button(self,textvariable=self.capt,comma
13              btn1.grid(row=1, column=2, padx=20, pady=30)
14              btn2 = tk.Button(self,text='退出',command=self.qu
15              btn2.grid(row=1, column=5)
16
17              self.pic1 = tk.PhotoImage(file='cat.png')
18              self.pic2 = tk.PhotoImage(file='dog.png')
19              self.lb1 = tk.Label(self,text='A cat.',image=self.pic1
20              self.lb1.grid(row=2, column=2)
21
22              self.info=tk.StringVar(value='这是猫咪！')
23              lb2=tk.Label(self,textvariable=self.info,fg='red')
24              lb2.grid(row=2,column=5)
```

第3~34行定义MainForm类，继承了类tk.Tk。

第10~24行为设计界面的方法myinitgui。第11行“self.capt=tk.StringVar(value='狗')”定义capt，初始值为“狗”，用作命令按钮btn1的显示内容。

第17行读入图像cat.png赋给pic1，注意，PhotoImage支持PNG、GIF、PGM和PPM四种格式的图像。

第18行“读入图像dog.png赋给pic2。

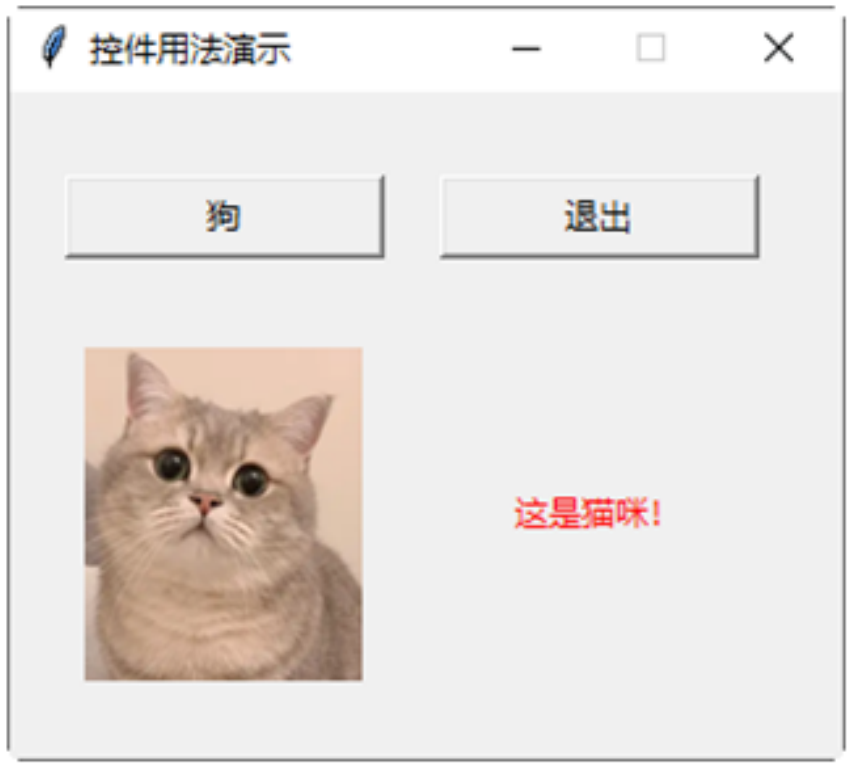
第19行“定义静态文本框lb1，显示图像pic1。

第22行定义info用作lb2的显示内容。

第23行定义静态文本框lb2，设定前景色为红色。

8.4.2 静态文本框

```
26      def btn1fun(self):
27          if self.capt.get()=='狗':
28
29
30
31
32
33
34
35  if
36
37      mainform.mainloop()
```



第26~34行为方法btn1fun。第
构，第27行如
]容为“狗”，
静态本文框中
btn1显示的内
lb2显示的内
否则（第31行）
静态本文框中
btn1显示的内
lb2显示的内

8.4.3 对话框

在tkinter.messagebox模块中提供了三种类型的消息对话框：

（1）信息提示对话框

信息提示对话框的形式为

showinfo(title='标题', message='提示信息')

这种形式的信息提示对话框，具有“标题”、“提示信息”和一个“OK”按钮，仅能返回OK信息（即messagebox.OK）。

（2）警告提示对话框

警告提示对话框的形式有两种，即

showwarning(title='标题', message='提示信息')

showerror(title='标题', message='提示信息')

这种形式的信息提示对话框，具有“标题”、“提示信息”和一个“OK”按钮，仅能返回OK信息。

8.4.3 对话框

（3）问题提示对话框

问题提示对话框具有五种形式，即

```
askquestion (title='标题', message='提示信息')    #可返回YES或NO
askokcancel (title='标题', message='提示信息')    #可返回True或False
askretrycancel (title='标题', message='提示信息')  #可返回True或False
askyesno (title='标题', message='提示信息')       #可返回True或False
askyesnocancel (title='标题', message='提示信息') #可返回True或False
```

在tkinter.filedialog模块中，提供了打开文件对话框和保存文件对话框，即askopenfilename和asksaveasfilename等方法。打开文件对话框返回选择的文件的完整路径字符串。

在tkinter.colorchooser模块提供了颜色选择对话框askcolor，返回的颜色值的形式形如“((245, 1, 10), '#f5010a')”。

8.4.3 对话框

下面的实例展示了这些对话框的用法（第一部分）：

```
1    import tkinter as tk
2    import tkinter.messagebox as msgbox
3    import tkinter.filedialog as filedlg
4    import tkinter.colorchooser as color
5    import os
6    class MainForm(tk.Tk):
7        def __init__(self):
8            super().__init__()
9            self.title('控件用法演示')
10           self.geometry('360x240')
11           self.resizable(False,False)
12           self.myinitgui()
13       def myinitgui(self):
14           self.capt=tk.StringVar(value='打开图像文件')
15           btn1 = tk.Button(self,textvariable=self.capt,comman
16           btn1.grid(row=1, column=2, padx=20, pady=30)
17           btn2 = tk.Button(self,text='退出',command=self.btn2
18           btn2.grid(row=1, column=3)
```

第2~4行装载messagebox模块，并赋以别名msgbox； 装载filedialog模块，并赋以别名filedlg,装载colorchooser模块，并赋以别名color。

第6~43行为自定义类MainForm，继承了类tk.Tk。

第13~26行为界面设计方法myinitgui。第14行定义capt对象，作为命令按钮btn1的显示内容。第15~16行定义命令按钮btn1，并使用grid方法将按钮放置到窗口。grid方法使用行、列位置摆放控件，padx和pady用于定义控件相对于窗口边缘的距离。btn1的单击事件响应方法为btn1fun。第17~18行定义命令按钮btn2，btn2的单击事件响应方法为btn2fun。

8.4.3 对话框

（第二部分）：

```
20          self.lb1 = tk.Label(self,text=
21          self.lb1.grid(row=2, colum
22
23          self.info=tk.StringVar(valu
24          self.lb2=tk.Label(self,text
25          self.lb2.grid(row=2,co
26          self.lb2.bind('<Button-1>',self.lb2fun
27      def btn1fun(self):
28          file=filedlg.askopenfilename(filetype
29                                       [('ima
30          if os.path.exists(file):
31              self.pic1 = tk.PhotoImage(file=f
32              self.lb1.config(image=self.pic1)
33              self.info.set(file)
```

一个事件用“< >”包括起来，“<Button-1>”表示鼠标左键单击事件，“<Button-3>”表示鼠标右键单击事件，“<Button-2>”表示鼠标中间键单击事件。

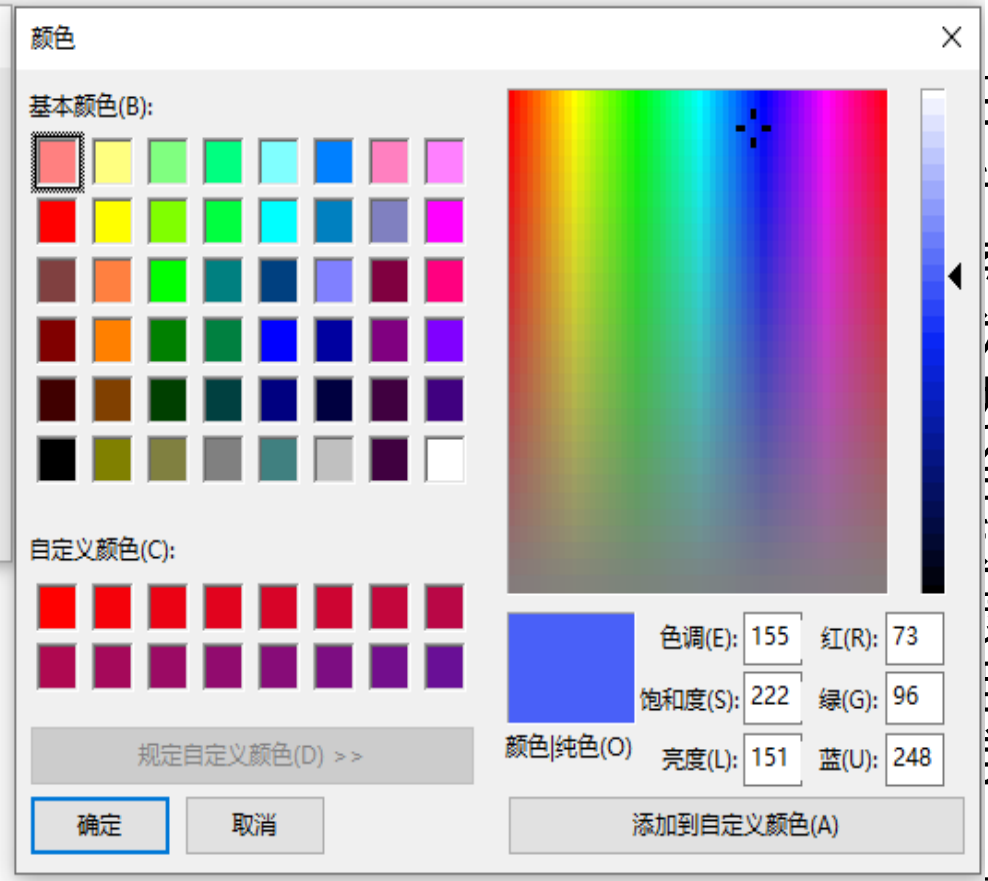
文本框lb1。
为静态文本框lb2的
静态文本框lb2。第
事件（即
法lb2fun相绑定，即当
本框lb2时，将触发lb2fun
函数。

第27~33行为btn1fun函数。第28~29行调用askopenfilename方法启动打开文件对话框。第30行如果file存在，则执行第31~33行。第31行将在打开文件对话框中选择的文件名file对应的图像文件装入到pic1中。第32行在静态文本框lb1中显示打开的图像。第33行在lb2中（info为lb2显示的内容）显示文件名。

8.4.3 对话框

（第三部分）：

```
34          def btn2fun(self):
35              a=msgbox.askquestion('退出软件','确定要退出软件吗？')
36              if
37
38              e
39
40          def lb
41              c
42              if
43
44      if __name_
45          mainf
46          mainf
```

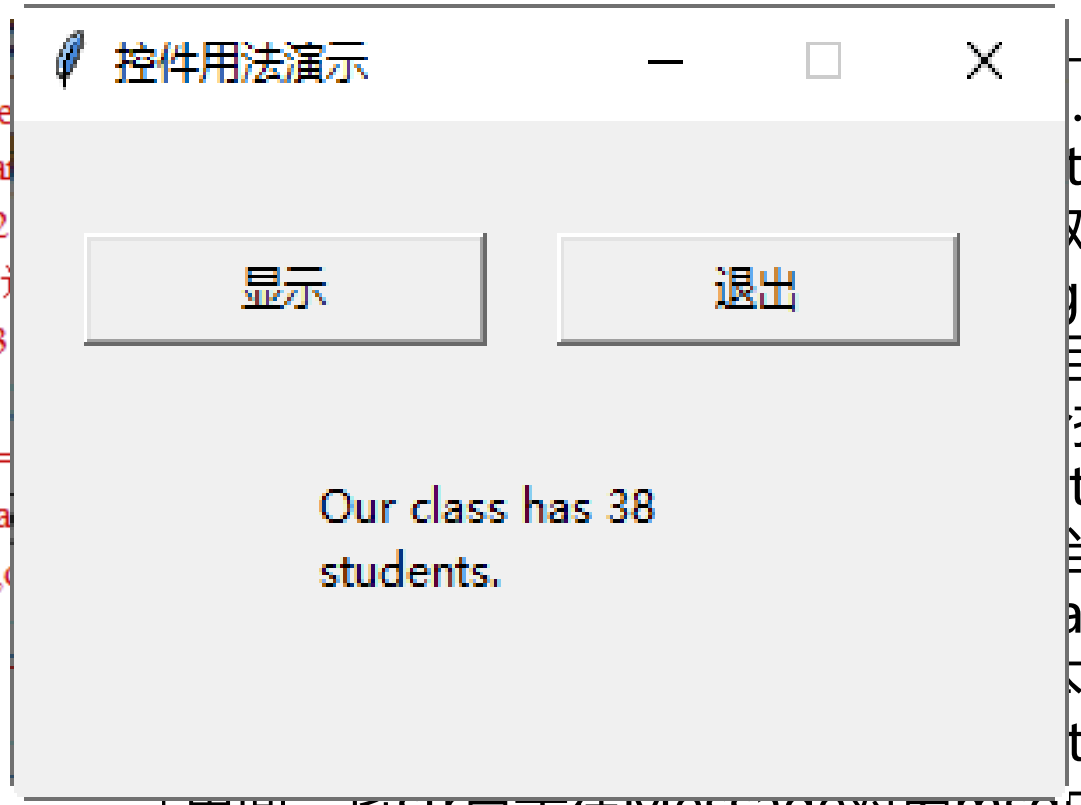


于“问题提
框返回
。
为静态文
，故需要
合法的标识
回的颜色
素为字符
，则第
字体颜色）

8.4.4 消息框

消息框也称静态文本控件，是Message类定义的对象，其与Label控件功能相似，但是消息框具有自动换行功能，可视为多行静态文本框。注意：Label静态文本框支持“\n”手动换行。

```
10        def myinitgui(self):
11            self.capt=tk.StringVar(value
12            btn1 = tk.Button(self,textva
13            btn1.grid(row=1, column=2
14            btn2 = tk.Button(self,text='
15            btn2.grid(row=1, column=3
16
17            self.str=tk.StringVar(value=
18            msg=tk.Message(self,textva
19            msg.grid(row=2,column=2,
20        def btn1fun(self):
21            file=open('zy.txt')
22            s=file.read()
23            file.close()
24            self.str.set(s)
```



.Tk。
tgui。
对象msg显示的内容，
ge对象msg。第19行
且占有3列的长度。
按钮btn1被按下时
txt')”打开文件
当前工程目录下有一
ass has 38
内容，赋给s；第23
tr，然后，系统刷新
界面，将str显示在Message对象msg中。

8.4.5 文本控件

文本控件是Text类的对象，借助于文本控件可以实现类于文档编辑软件（例如Word）类似的图文编辑处理，这个控件是实现文档编辑类软件的必备控件，可以作为容器放置其他控件和图像。

这里重点介绍一下该控件实现文字编辑的功能，设文本控件为txt，则

（1）在控件中可以手工编辑文字；

（2）借助于insert方法可以向txt中插入文字，例如：

txt.insert(tk.INSERT, 'Apple')

向txt文本控件中当前光标位置处插入字符串“Apple”。

或者：txt.insert('行号.列号', 'Apple')

表示在第“行号”和第“列号”处插入字符串“Apple”。如，txt.insert('1.3', 'Apple')表示在第1行第3列处插入字符串“Apple”。这里列号从0开始，行号从1开始。

8.4.5 文本控件

（3）借助于get方法从txt中提取文本，例如：

txt.get('行号1.列号1', '行号2.列号2')

表示从位置“行号1.列号1”至“行号2.列号2”（不含）间的文本被提取出来，返回字符串。

（4）借助于delete方法删除文本，例如：

delete('行号1.列号1', '行号2.列号2')

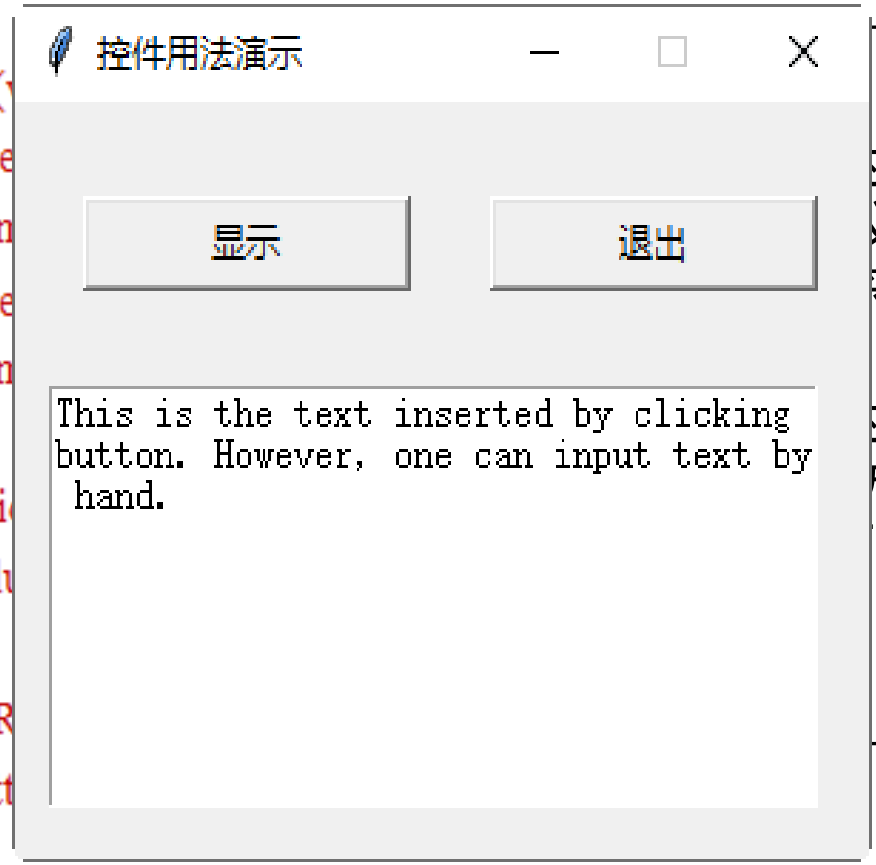
表示删除从位置“行号1.列号1”至“行号2.列号2”（不含）间的文本。

在上述操作中，可以使用tk.INSERT、tk.END表示文本的当前光标位置和文本的最后位置（的下一个位置）。

8.4.5 文本控件

下面将介绍文本控件的插入文本操作：

```
10      def myinitgui(self):
11          self.capt=tk.StringVar(
12          btn1 = tk.Button(self,te
13          btn1.grid(row=1, colun
14          btn2 = tk.Button(self,te
15          btn2.grid(row=1, colun
16
17          self.txt=tk.Text(self,wi
18          self.txt.grid(row=2,colu
19      def btn1fun(self):
20          self.txt.insert(tk.INSER
21                              ' butt
```



nForm。
用户界面设计方法myinitgui。
义文本控件txt，高度为38，高度
符。第18行调用grid方法在窗口

方法btn1fun，当单击控件
方法。第20~21行表示在当前光
his is the text inserted by
However, one can input text

8.4.6 编辑框

编辑框为Entry类定义的对象，编辑框类似于“控制台模式”下的input函数，可以输入各类数据。设编辑框对象为entry，其textvariable参数为val，则val.get方法将获得编辑框entry中的文本，而val.set方法将设置编辑框中显示的内容，set方法可以使用各种类型，例如，val.set('{'a': 5, 'b': 5, 'c': 6}')将输出字典到编辑框中。