

# 电子科技大学 2023-2024 学年第 2 学期

## 《Python 语言程序设计》课程设计

---

姓名：经彭宇

学号：202104092007

专业：机器人工程

---

课程设计的题目：记事本工具的实现

### 第一部分 程序功能简介

这个程序是一个简单的记事本应用，基于 Python 的 Tkinter 库开发。它具有以下主要功能：

1. 文件操作：新建文件、打开文件、保存文件、另存为、重命名、删除文件。
2. 编辑功能：撤销、重做、剪切、复制、粘贴、查找、全选。
3. 格式设置：设置字体、加密文本、解密文本。

程序主要由菜单栏、快捷栏和文本编辑区域组成。菜单栏包括文件、编辑和格式三个主要菜单，用户可以通过菜单栏或快捷键执行相应的功能操作。

这是程序运行的截屏图，可以看到菜单栏中包含了文件、编辑和格式三个主要菜单，用户可以通过点击菜单项或使用快捷键来执行相应的功能操作。文本编辑区域显示了用户可以编辑的文本内容。

### 第二部分 程序的运行使用方法和步骤说明

1. 打开程序后，可以在菜单栏中选择文件操作、编辑功能和格式设置等。
2. 文件操作：
  - 新建文件：点击菜单栏中的“文件”->“新建”，或者使用快捷键“Ctrl+N”。
  - 打开文件：点击菜单栏中的“文件”->“打开”，或者使用快捷键“Ctrl+O”。
  - 保存文件：点击菜单栏中的“文件”->“保存”，或者使用快捷键“Ctrl+S”。
  - 另存为：点击菜单栏中的“文件”->“另存为”，或者使用快捷键“Ctrl+Shift+S”。
  - 重命名：点击菜单栏中的“文件”->“重命名”，或者使用快捷键“Ctrl+R”。
  - 删除文件：点击菜单栏中的“文件”->“删除”，或者使用快捷键“Ctrl+D”。
3. 编辑功能：
  - 撤销：点击菜单栏中的“编辑”->“撤销”，或者使用快捷键“Ctrl+Z”。
  - 重做：点击菜单栏中的“编辑”->“重做”，或者使用快捷键“Ctrl+Y”。
  - 剪切：点击菜单栏中的“编辑”->“剪切”，或者使用快捷键“Ctrl+X”。
  - 复制：点击菜单栏中的“编辑”->“复制”，或者使用快捷键“Ctrl+C”。

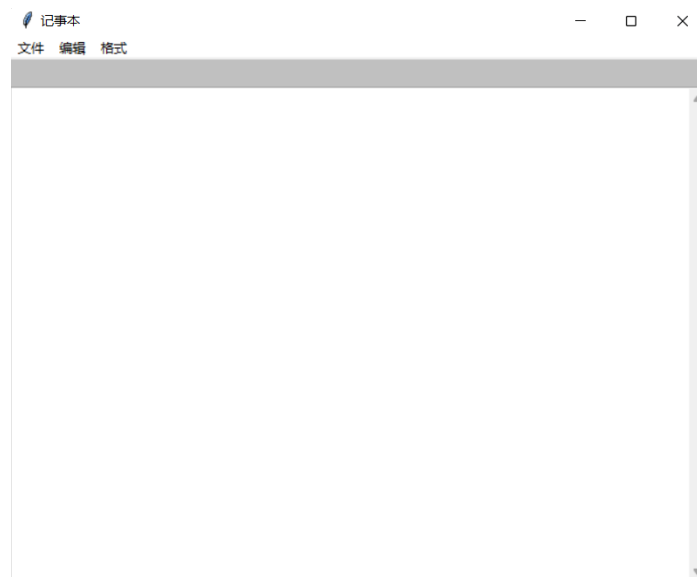
- 粘贴：点击菜单栏中的“编辑”->“粘贴”，或者使用快捷键“Ctrl+V”。
- 查找：点击菜单栏中的“编辑”->“查找”，或者使用快捷键“Ctrl+F”。
- 全选：点击菜单栏中的“编辑”->“全选”，或者使用快捷键“Ctrl+A”。

#### 4. 格式设置：

- 设置字体：点击菜单栏中的“格式”->“字体”。
- 加密文本：点击菜单栏中的“格式”->“加密”。
- 解密文本：点击菜单栏中的“格式”->“解密”。

### 第三部分 程序运行效果的截屏图

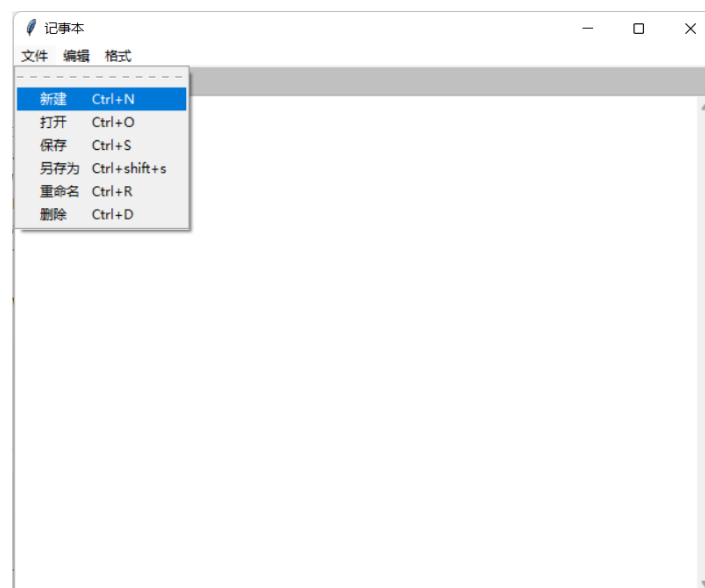
#### 1.源程序运行结果

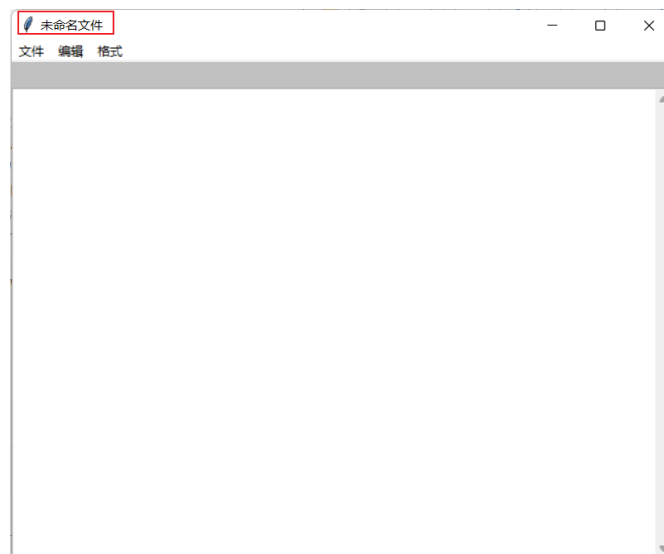


#### 2.记事本功能

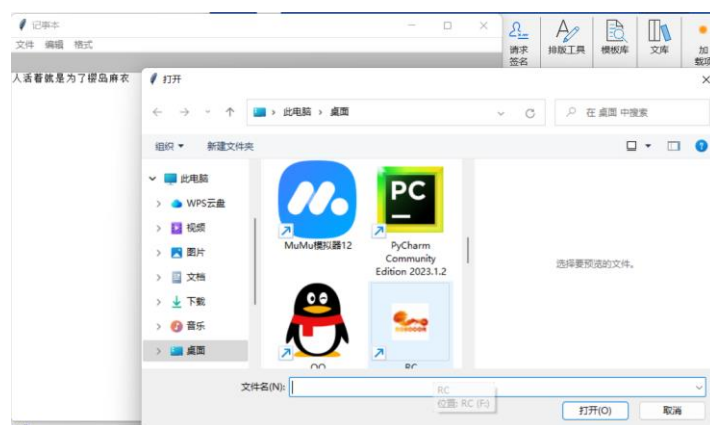
##### (1) 文件

##### ①新建

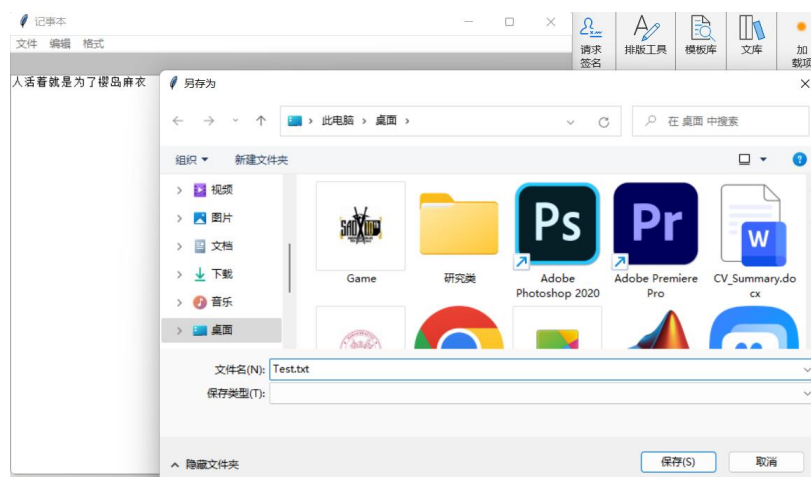




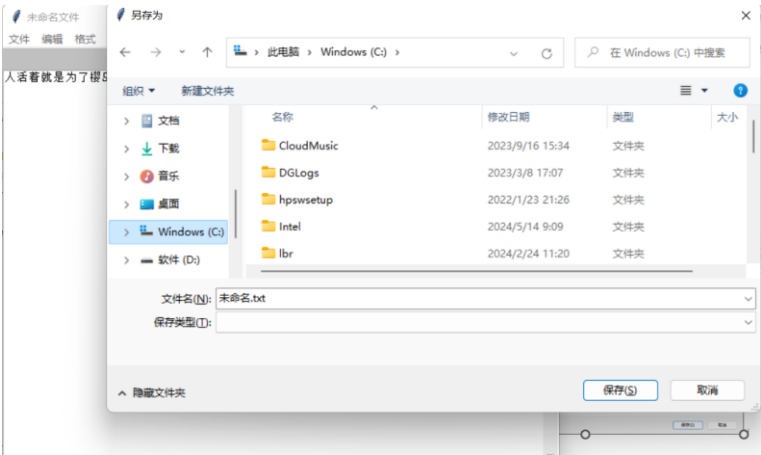
②打开



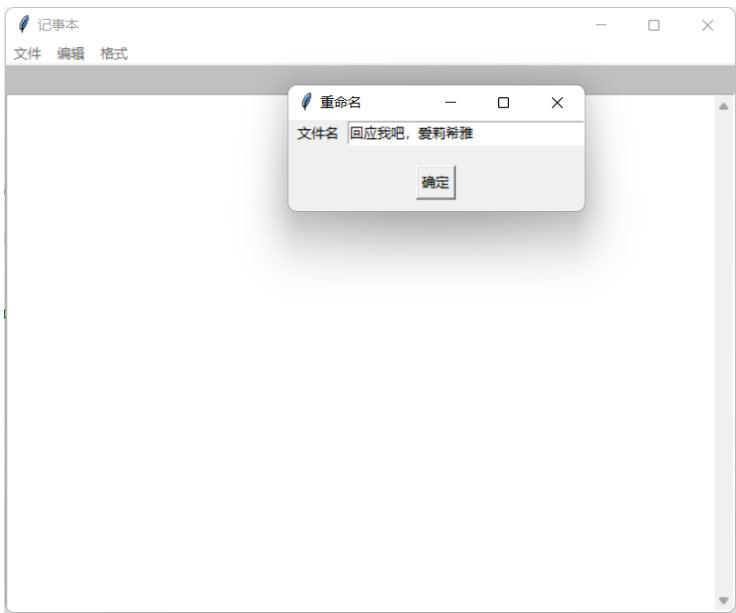
③保存



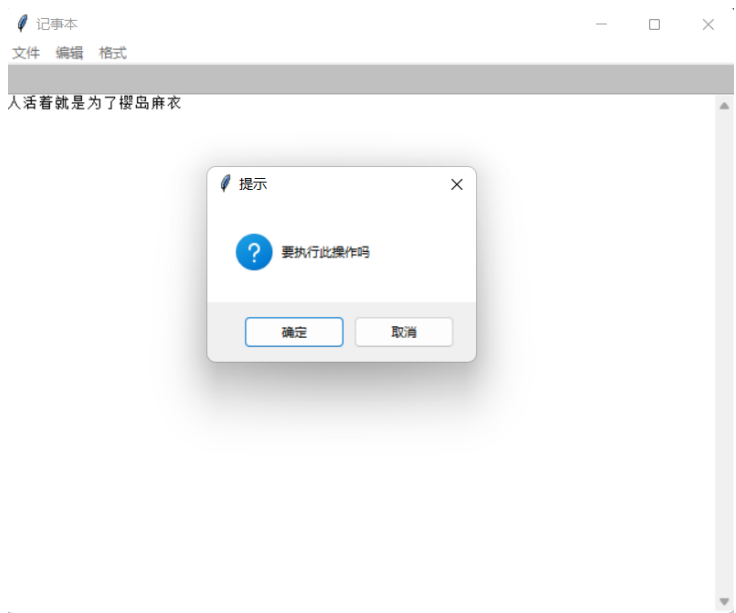
④另存为



⑤重命名



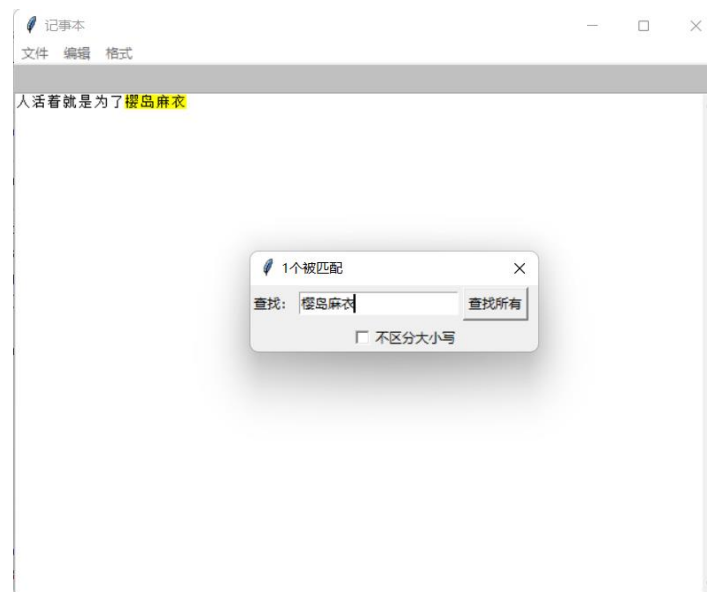
⑥删除



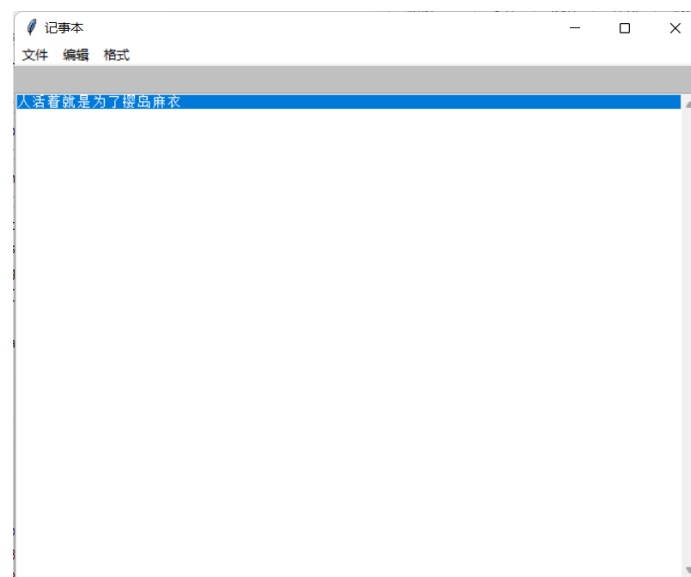
## (2) 编辑

①撤销、重做、剪切、复制、粘贴等操作实际上手即可

②查找

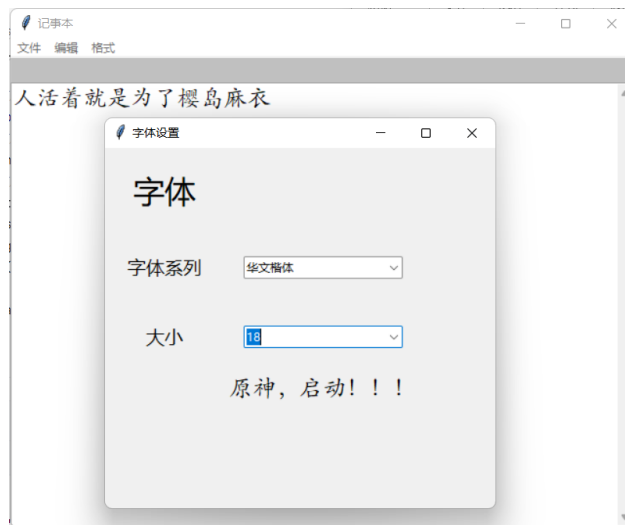


③全选

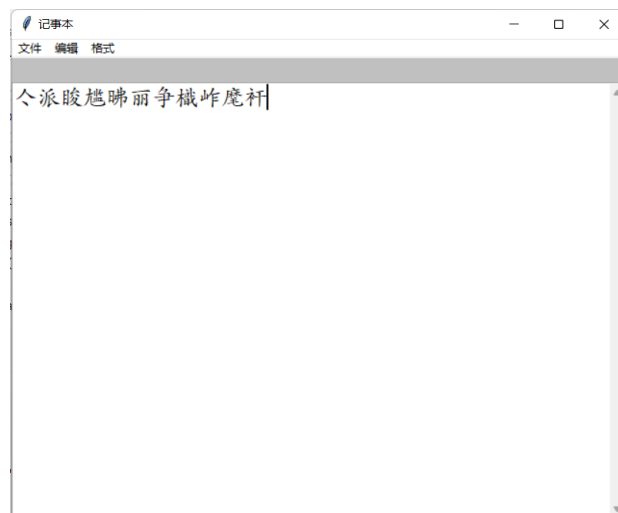


### (3) 格式

#### ①更改字体



#### ②加密



#### ③解密

点击解密便可显示原来的文字

3.其他功能

(1) 右键操作



(2) 快捷键操作

功能	快捷键
新建文件	Ctrl+N
打开文件	Ctrl+O
保存文件	Ctrl+S
另存为	Ctrl+Shift+S
重命名	Ctrl+R
删除文件	Ctrl+D
撤销	Ctrl+Z
重做	Ctrl+Y
剪切	Ctrl+X
复制	Ctrl+C
粘贴	Ctrl+V
查找	Ctrl+F
全选	Ctrl+A

第四部分 程序源代码

这里是程序的源代码，包含了各种功能的实现和相关的注释说明。例如：新建文件、打开文件、保存文件等操作的实现方法以及相应的注释说明。另外，也包括了对于窗口的布局、菜单栏的创建、快捷键的绑定等部分的代码实现和注释。 请参考下方给出的代码段。

源代码：

```
from tkinter import *
from tkinter.filedialog import *
from tkinter.messagebox import *
from tkinter import scrolledtext
import os
```

```

from tkinter.ttk import Combobox

# 初始化全局变量
filename = ''
font = ('微软雅黑', 10)
encryped = False
decoded = False

# 新建文件
def new_file(*args):
    global top, filename, textPad
    top.title("未命名文件")
    filename = None
    textPad.delete(1.0, END)

# 打开文件
def open_file(*args):
    global filename
    filename = askopenfilename(defaulttextextension=".txt")
    if filename == "":
        filename = None
    else:
        top.title("" + os.path.basename(filename))
        textPad.delete(1.0, END)
        f = open(filename, 'r', encoding="utf-8")
        textPad.insert(1.0, f.read())
        f.close()

# 单击打开文件
def click_open(event):
    global filename
    top.title("" + os.path.basename(filename))
    textPad.delete(1.0, END)
    f = open(filename, 'r', encoding="utf-8")
    textPad.insert(1.0, f.read())
    f.close()

# 保存文件
def save(*args):
    global filename
    try:
        f = open(filename, 'w', encoding="utf-8")
        msg = textPad.get(1.0, 'end')
        f.write(msg)

```



```

        f.close()
    except:
        save_as()

# 另存为文件
def save_as(*args):
    global filename
    f = asksaveasfilename(initialfile="未命名.txt",
defaultextension=".txt")
    filename = f
    fh = open(f, 'w', encoding="utf-8")
    msg = textPad.get(1.0, END)
    fh.write(msg)
    fh.close()
    top.title("'" + os.path.basename(f))

# 重命名文件
def rename(newname):
    global filename
    name = os.path.basename(os.path.splitext(filename)[0])
    oldpath = filename
    newpath = os.path.dirname(oldpath) + '/' + newname + '.txt'
    os.rename(oldpath, newpath)
    filename = newpath
    refresh()

# 打开重命名对话框
def rename_file(*args):
    global filename
    t = Toplevel()
    t.geometry("260x80+200+250")
    t.title('重命名')
    frame = Frame(t)
    frame.pack(fill=X)
    lable = Label(frame, text="文件名")
    lable.pack(side=LEFT, padx=5)
    var = StringVar()
    e1 = Entry(frame, textvariable=var)
    e1.pack(expand=YES, fill=X, side=RIGHT)
    botton = Button(t, text="确定", command=lambda: rename(var.get()))
    botton.pack(side=BOTTOM, pady=10)

# 删除文件
def delete(*args):

```

```

global filename, top
choice = askokcancel('提示', '要执行此操作吗')
if choice:
    if os.path.exists(filename):
        os.remove(filename)
        textPad.delete(1.0, END)
        top.title("记事本")
        filename = ''

# 剪切
def cut():
    global textPad
    textPad.event_generate("<<Cut>>")

# 复制
def copy():
    global textPad
    textPad.event_generate("<<Copy>>")

# 粘贴
def paste():
    global textPad
    textPad.event_generate("<<Paste>>")

# 撤销
def undo():
    global textPad
    textPad.event_generate("<<Undo>>")

# 重做
def redo():
    global textPad
    textPad.event_generate("<<Redo>>")

# 全选
def select_all():
    global textPad
    textPad.tag_add("sel", "1.0", "end")

# 查找
def find(*args):
    global textPad
    t = Toplevel(top)
    t.title("查找")

```

```

t.geometry("260x60+200+250")
t.transient(top)
Label(t, text="查找: ").grid(row=0, column=0, sticky="e")
v = StringVar()
e = Entry(t, width=20, textvariable=v)
e.grid(row=0, column=1, padx=2, pady=2, sticky="we")
e.focus_set()
c = IntVar()
Checkbutton(t, text="不区分大小写", variable=c).grid(row=1,
column=1, sticky='e')
Button(t, text="查找所有", command=lambda: search(v.get(), c.get(),
textPad, t, e)).grid \
    (row=0, column=2, sticky="e" + "w", padx=2, pady=2)

def close_search():
    textPad.tag_remove("match", "1.0", END)
    t.destroy()

t.protocol("WM_DELETE_WINDOW", close_search)

# 弹出右键菜单
def mypopup(event):
    global editmenu
    editmenu.tk_popup(event.x_root, event.y_root)

# 查找功能
def search(needle, cssnstv, textPad, t, e):
    textPad.tag_remove("match", "1.0", END)
    count = 0
    if needle:
        start = 1.0 # 1.0 代表第一行第一个字符, 小数点前表示行数, 小数点后表示
光标所在字符数位数
        while True:
            pos = textPad.search(needle, start, nocase=cssnstv,
stopindex=END)
            if not pos:
                break
            strlist = pos.split('.')
            left = strlist[0]
            right = str(int(strlist[1]) + len(needle))
            lastpos = left + '.' + right
            textPad.tag_add("match", pos, lastpos)
            count += 1
            start = lastpos

```

```

        textPad.tag_config('match', background="yellow")
    e.focus_set()
    t.title(str(count) + "个被匹配")

# 刷新窗口标题
def refresh():
    global top, filename
    if filename:
        top.title(os.path.basename(filename))
    else:
        top.title("记事本")

# 设置字体
def fontset():
    global font
    w2 = Tk()
    w2.title("字体设置")
    w2.geometry("400x400")
    label_title = Label(w2, text='字体', font=('微软雅黑', 24))
    label_title.grid(row=0, column=0, pady=20)
    label1 = Label(w2, text='字体系列', font=('微软雅黑', 14))
    label1.grid(row=1, column=0, pady=20, padx=20)
    label2 = Label(w2, text='大小', font=('微软雅黑', 14))
    label2.grid(row=2, column=0, pady=20, padx=20)
    ziti = ['Arial', 'Calibri', '宋体', '微软雅黑', '华文隶书', '华文行楷', '华文楷体', '楷体']
    cb1 = Combobox(w2, values=ziti)
    cb1.current(0)
    cb1.grid(row=1, column=1, padx=20)
    daxiao = [x for x in range(8, 23, 2)]
    cb2 = Combobox(w2, values=daxiao)
    cb2.current(0)
    cb2.grid(row=2, column=1, padx=20)
    label_show = Label(w2, text='原神, 启动!!!', font=font)
    label_show.grid(row=3, column=1)

def change_ziti(event):
    global font, ziti_chosen, daxiao_chosen
    ziti_chosen = cb1.get()
    daxiao_chosen = cb2.get()
    font = (ziti_chosen, daxiao_chosen)
    print(font)
    label_show.config(font=font)
    textPad.config(font=font)

```

```

cb1.bind('<<ComboboxSelected>>', change_ziti)
cb2.bind('<<ComboboxSelected>>', change_ziti)

# 加密文本
def encrypt():
    '凯撒加密'
    global encryped, decoded
    if not encryped:
        raw = textPad.get(0.0, END)
        string_list = raw.split('\n')
        string_list.remove('')
        print(string_list)
        encrypt_list = []
        # 一行一行进行加密
        for s in string_list:
            encrypt = ''
            for x in s:
                encrypt += chr(ord(x) + 3)
            encrypt_list.append(encrypt)
        print(encrypt_list)
        result = '\n'.join(encrypt_list)
        textPad.delete(0.0, END)
        textPad.insert(0.0, result)
        encryped = True
        decoded = False

# 解密文本
def decode():
    global encryped, decoded
    if not decoded:
        raw = textPad.get(0.0, END)
        string_list = raw.split('\n')
        string_list.remove('')
        print(string_list)
        encrypt_list = []
        # 一行一行进行加密
        for s in string_list:
            encrypt = ''
            for x in s:
                encrypt += chr(ord(x) - 3)
            encrypt_list.append(encrypt)
        print(encrypt_list)
        result = '\n'.join(encrypt_list)

```

```

        textPad.delete(0.0, END)
        textPad.insert(0.0, result)
        decoded = True
        encryped = False

# 创建主窗口
top = Tk()
top.title("记事本")
top.geometry("640x480+100+50")

# 创建菜单栏
menubar = Menu(top)

# 文件菜单
filemenu = Menu(top)
filemenu.add_command(label="新建", accelerator="Ctrl+N",
command=new_file)
filemenu.add_command(label="打开", accelerator="Ctrl+O",
command=open_file)
filemenu.add_command(label="保存", accelerator="Ctrl+S",
command=save)
filemenu.add_command(label="另存为", accelerator="Ctrl+shift+s",
command=save_as)
filemenu.add_command(label="重命名", accelerator="Ctrl+R",
command=rename_file)
filemenu.add_command(label="删除", accelerator="Ctrl+D",
command=delete)
menubar.add_cascade(label="文件", menu=filemenu)

# 编辑菜单
editmenu = Menu(top)
editmenu.add_command(label="撤销", accelerator="Ctrl+Z",
command=undo)
editmenu.add_command(label="重做", accelerator="Ctrl+Y",
command=redo)
editmenu.add_separator()
editmenu.add_command(label="剪切", accelerator="Ctrl+X", command=cut)
editmenu.add_command(label="复制", accelerator="Ctrl+C",
command=copy)
editmenu.add_command(label="粘贴", accelerator="Ctrl+V",
command=paste)
editmenu.add_separator()
editmenu.add_command(label="查找", accelerator="Ctrl+F",
command=find)

```

```

editmenu.add_command(label="全选", accelerator="Ctrl+A",
command=select_all)
menubar.add_cascade(label="编辑", menu=editmenu)

# 格式菜单
format_menu = Menu(top)
format_menu.add_command(label="字体", command=fontset)
format_menu.add_command(label="加密", command=encrypt)
format_menu.add_command(label="解密", command=decode)
menubar.add_cascade(label="格式", menu=format_menu)

top['menu'] = menubar

# 创建快捷栏和文本框
shortcutbar = Frame(top, height=25, bg='Silver')
shortcutbar.pack(expand=NO, fill=X)

textPad = Text(top, undo=True)
textPad.pack(expand=YES, fill=BOTH)
scroll = Scrollbar(textPad)
textPad.config(yscrollcommand=scroll.set)
scroll.config(command=textPad.yview)
scroll.pack(side=RIGHT, fill=Y)

# 绑定热键和右键菜单
textPad.bind("<Control-N>", new_file)
textPad.bind("<Control-n>", new_file)
textPad.bind("<Control-O>", open_file)
textPad.bind("<Control-o>", open_file)
textPad.bind("<Control-S>", save)
textPad.bind("<Control-s>", save)
textPad.bind("<Control-D>", delete)
textPad.bind("<Control-d>", delete)
textPad.bind("<Control-R>", rename_file)
textPad.bind("<Control-r>", rename_file)
textPad.bind("<Control-A>", select_all)
textPad.bind("<Control-a>", select_all)
textPad.bind("<Control-F>", find)
textPad.bind("<Control-f>", find)

textPad.bind("<Button-3>", mypopup)
top.mainloop()

```