



大连理工大学
DALIAN UNIVERSITY OF TECHNOLOGY

大 作 业

课程名称: 开源软件基础

专业班级: 软 1610

学生姓名: 李辉

学 号: 201692324

题 目: 基于 python 的压缩工具

修读方式: 正常

授课教师: 戴轶群

评分标准 总分 (100)

序号	模块	评分项	分值	得分
1	大作业报告 (40 分)	需求分析：需求描述准确简练，意思表达清楚，无明显错误；需要画出 UML 需求用例图及用例分析说明。	10 分	
2		系统设计：需要画出参与整个系统的各个模块，解释各模块功能；用 UML 画出核心模块时序图、类图，及其各个模块之间的关系。模型中的各图表达准确，无明显错误，图与图之间的关系明确，表达完整。	10 分	
3		系统测试：对主要功能模块需要有测试用例，最后要有测试结果。	5 分	
4		系统实现：核心模块界面、关键代码，并注释。	10 分	
5		心得体会：每个人都要写。	5 分	
6	软件作品 (40 分)	界面设计：要求界面美观，操作简易，体现个性。	10 分	
7		数据库设计：数据库内表的建立合理，具有最基本的约束，设置表的主键、外键，需要阐明这样设置的原因。	5 分	
8		系统实现：完成项目基本功能，以该项目在市场上同类项目中一般应用功能作比较。	10 分	
9		创新性：完成项目基本模块之外的特殊功能。	15 分	
10	平时表现	平时考勤。	10 分	
11	(20 分)	平时作业。	10 分	
总分			备注	
教师签字：				
年 月 日				

目录

1 引言.....	1
1.1 课题背景.....	1
1.2 所用技术.....	1
2 需求分析.....	2
2.1 需求描述.....	2
3 系统分析.....	2
4 系统设计.....	3
4.1 结构分析.....	3
4.2 模块功能图.....	4
5 系统实现.....	5
5.1 步骤及截图.....	5
5.2 代码实现.....	7
6 结束语.....	10

基于 Python 的压缩工具

1 引言

1.1 课题背景

压缩工具是我们日常工作、生活、学习中必不可少的工具。一个好的压缩工具应该具有压缩率高、压缩速度快、数据无损，可加密等特点。使用压缩工具可以极大的减少存储一些文件所需要的空间，从而实现更加方便、快速的存储。

自 Phil Katz（菲尔·卡茨）在 1989 年发明了 zip 压缩算法之后，PKZIP 一度成为压缩工具的不二之选，他将人们带入了一个更加广阔的数字世界。在这之后，各种更加高效，压缩率更高的算法不断出现，文件可以变得更小、更小。这是令人兴奋的。

学习开源软件基础以及 Python 之后，我希望可以实现一个带有界面的压缩工具，可以实现多种压缩格式的压缩，提供带有密码的压缩，从而实现对一些数据的保密处理，同时可以进行压缩率的设置，实现对压缩时间和压缩率的自主选择，对处理需求不同的文件能够进行多样化、个性化的处理。

1.2 所用技术

Python 是一种开源的、解析性的，面向对象的语言编程。它语法优雅，可读性强，让程序员注重功能实现，而非代码本身实现细节。现广泛运用于 Web 开发、运维自动化、测试自动化及数据挖掘等多个行业和领域。Python 语言有很大的优势：比 Java、C++ 简单更易于使用；比 PHP 易懂易学并且用途更广；比 Perl 更简洁的语法、更简单的设计，更具可读性、更易于维护，有助于减少 Bug。但它的性能不如 Java、C、C++ 这类编译性语言强大。

同时，我想借助对该项目的开发，对 Python 课上学习的知识进行实践，更进一步的学习 Python 语言的其他知识。由于是一个学习型软件，暂不考虑性能的情况下，本系统开发选用 Python 语言编写。

在界面的实现上，使用 Python 的 tkinter 库进行开发。

2 需求分析

2.1 需求描述

本软件实现的主体功能为压缩及解压缩。在此过程中，还希望拥有以下功能来丰富压缩的可选项，提高压缩功能及压缩体验。

1) 选择不同路径下的多个文件进行合并压缩。对于选择出来的文件要求进行文件名、文件大小、文件路径的展示。对文件列表中选中的文件可以进行删除操作。

2) 可以自己选择压缩后的存储路径。同时，在不进行设置的情况下，能够根据操作系统的不同显示不同的默认的路径。

3) 选择不同的文件保存格式。包括 `zip`、`7z`、`tar` 等格式。

4) 选择压缩等级。根据不同文件的需求进行压缩等级的选择，比如对不关注压缩率而关注文件保密的文件可以选择零压缩；对于既关注文件保密又关注文件压缩解压缩时间的可能频繁操作的文件可以选择快速压缩；对于一般文件选择正常压缩；对于关注压缩率的文件可以忍受压缩时间长的文件选择最大压缩。

5) 设置压缩密码。对于一些比较隐私、机密性高的文件，希望可以设置压缩密码以实现在解压时，需要提供正确密码才能成功解压。对于压缩密码要能进行非法字符检验，密码与确认密码一致性检验。

6) 使用密码（或无密码）对选择的压缩文件进行定向的解压。

3 系统分析

本次设计用 PyCharm 开发工具，Python 语言，tkinter 及 pyminizip 等工具进行压缩软件系统的开发，主要功能有可选路径、可选保存格式及密码的压缩。用户选择新建压缩文件，并通过界面化的操作对压缩文件进行一系列的设置。选择打开文件，选择要解压的文件，对所选文件进行解压。

4 系统设计

4.1 结构分析

(1) 根据系统分析，画出系统功能模块结构图：

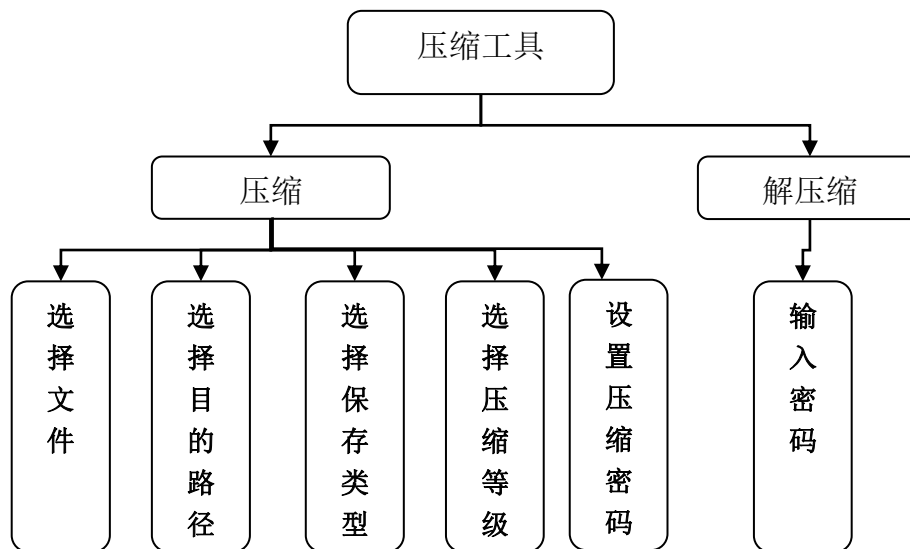


图 4.1 系统功能模块结构图

(2) 画出系统模块功能流程图：

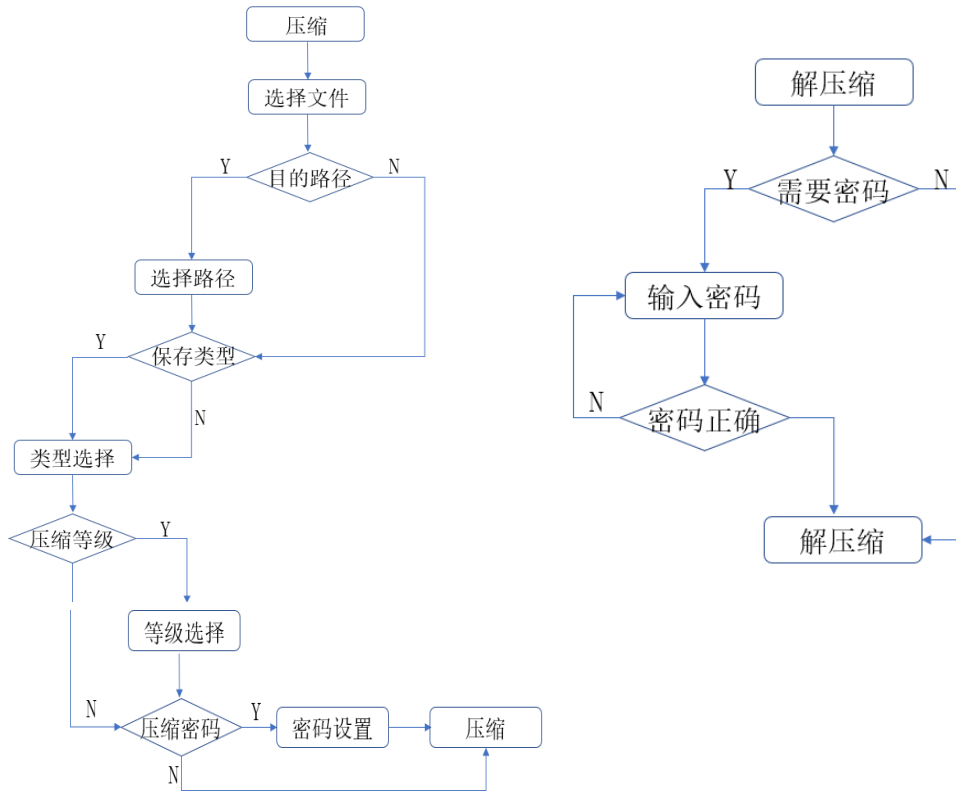


图 4.2 系统功能模块流程图

4.2 模块功能图

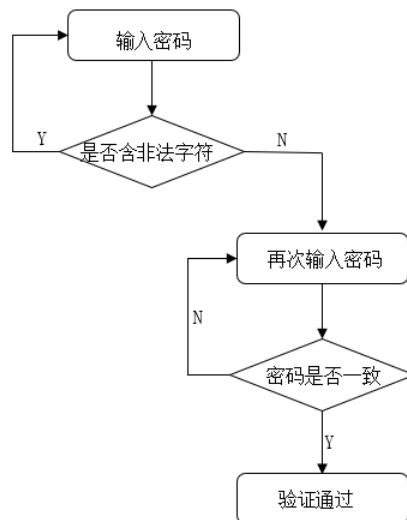


图4.3 密码模块

5 系统实现

5.1 步骤及截图

(1) 主界面

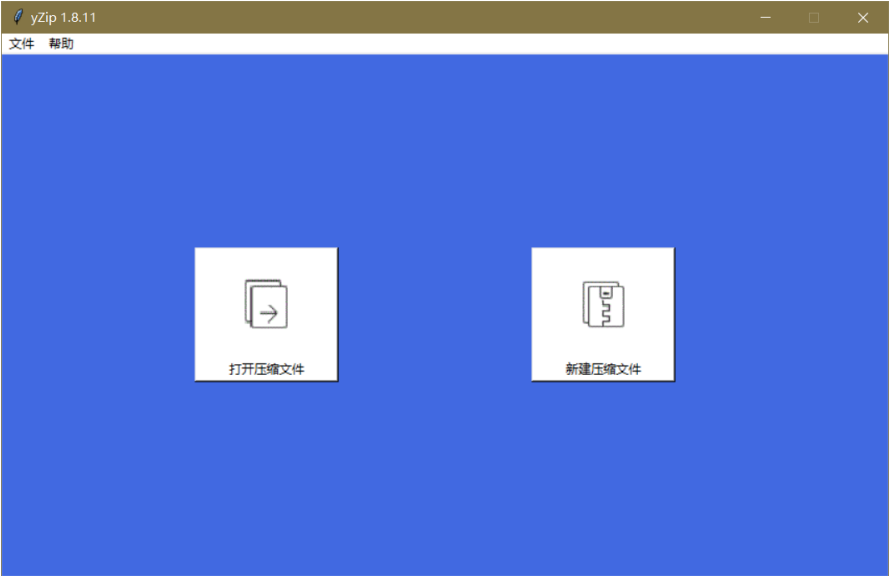


图 5.1 主界面

(2) 新建压缩文件界面

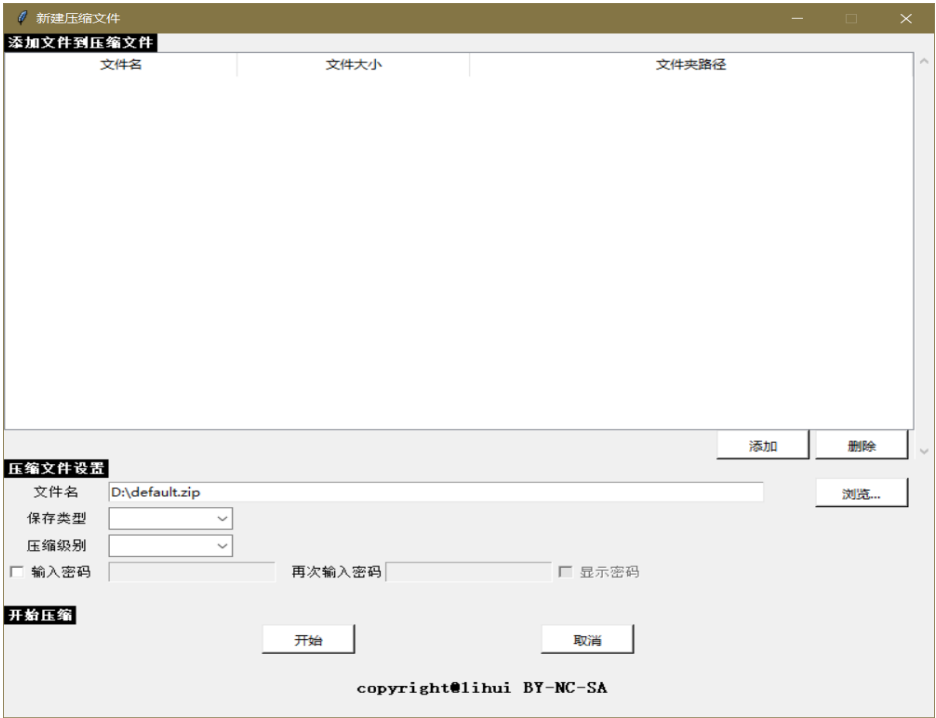


图 5.2 新建压缩文件

(3) 添加要压缩的文件

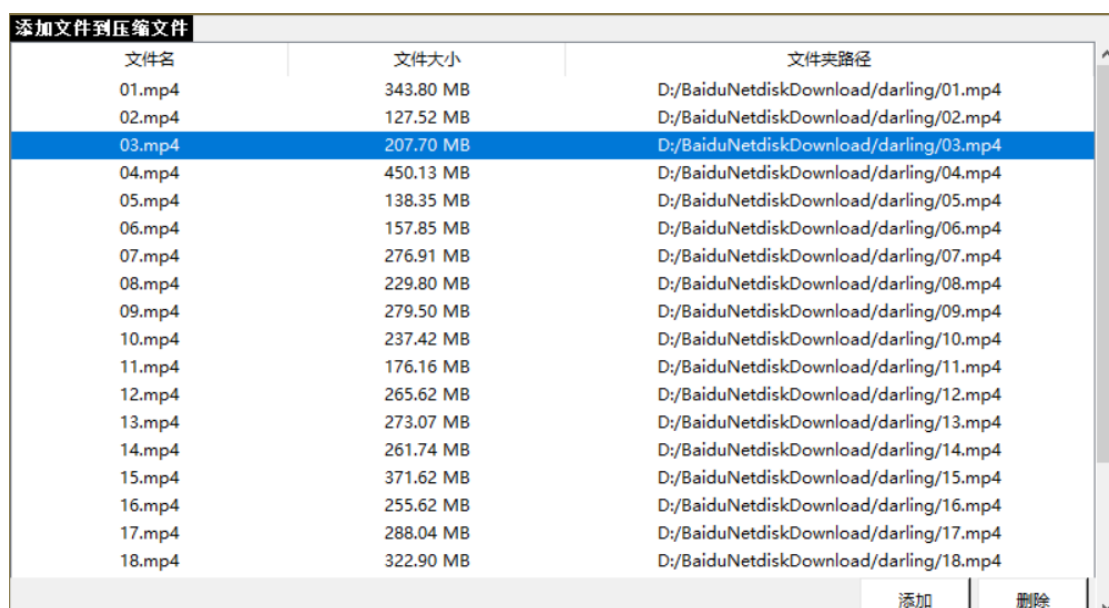


图 5.3 添加要压缩的文件

(4) 设置界面

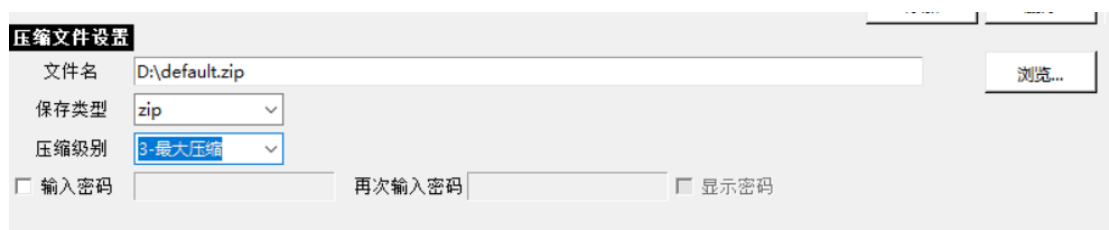


图 5.4 设置界面

(5) 密码非法字符检验



图 5.5 非法字符

(6) 密码不一致检验

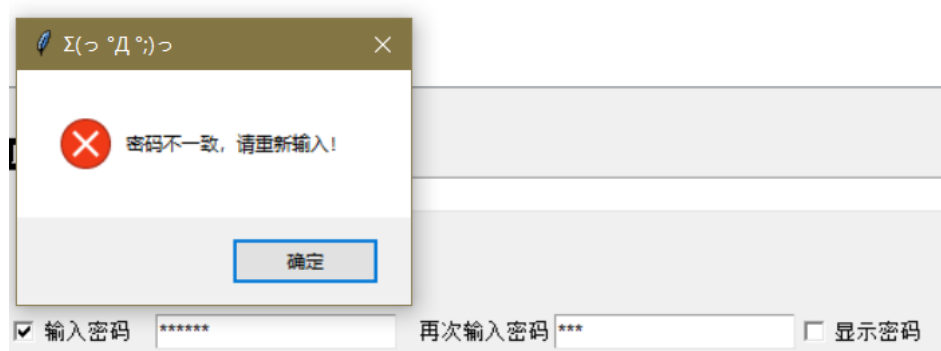


图 5.6 密码不一致检验

(7) 压缩过程提示

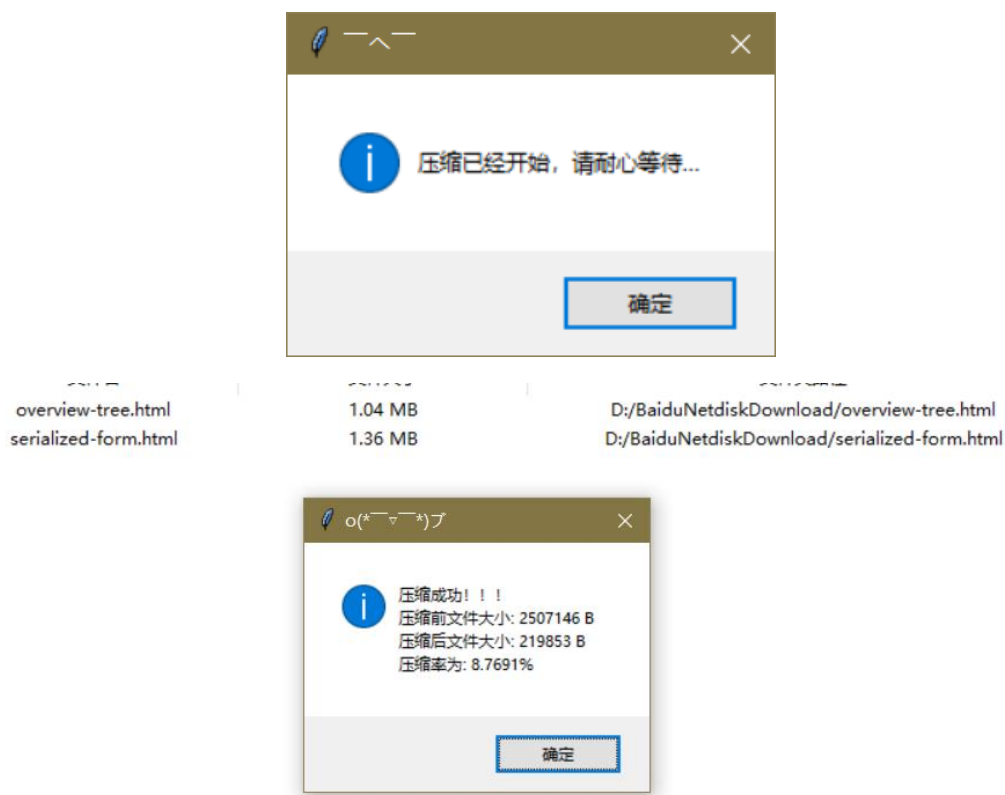


图 5.7 过程提示

5.2 代码实现

(1) 选择文件部分

标签一: 添加文件到压缩文件

```
Label(self.top, text="添加文件到压缩文件",  
      font=("YaHei", 10, "bold"), bg="black", fg="white").grid(row=0, column=0, sticky=W)
```

生成文件选择表格的容器

```
frame1 = Frame(self.top)
```

```
frame1.grid(row=1, column=0, sticky='w') # sticky='w' 指定了组件在单元格中靠左对齐
```

滚动条

```
scrollBar = Scrollbar(frame1)
```

```
scrollBar.pack(side=RIGHT, fill=Y)
```

文件选择表格

```

columns = ("文件名", "文件大小", "文件夹路径")
treeview = Treeview(frame1, height=18, show="headings",
                    columns=columns, yscrollcommand=scrollBar.set)
# 设置每列宽度和对齐方式
treeview.column("文件名", width=200, anchor='center')
treeview.column("文件大小", width=200, anchor='center')
treeview.column("文件夹路径", width=380, anchor='center')
# 设置每列表头标题文本
treeview.heading("文件名", text="文件名")
treeview.heading("文件大小", text="文件大小")
treeview.heading("文件夹路径", text="文件夹路径")
treeview.pack()
# Treeview 组件与垂直滚动条结合
scrollBar.config(command=treeview.yview)
# 添加与删除按钮
deleteButton = Button(frame1, text="删除",
                      bg='white', fg='black', width='10', activeforeground='black',
                      activebackground='LightSkyBlue',
                      cursor='hand2', anchor='s', compound="center",
                      command=lambda: DeleteFilesFromTreeview(treeview))
deleteButton.pack(side=RIGHT, padx=5)
addButton = Button(frame1, text="添加",
                  bg='white', fg='black', width='10',
                  activeforeground='black', activebackground='LightSkyBlue',
                  cursor='hand2', anchor='s', compound="center",
                  command=lambda: SelectFiles(treeview, self.master))
addButton.pack(side=RIGHT)
return treeview

```

(2) 使用正则表达式密码非法字符检验

密码输入验证, 密码要求为 ASCII 码 (十六进制 20-7E), 即 ASCII 中可显示字符部份

```

def validateKey(key):
    # 如果之前有非法字符, 则在再次验证时把输入密码有误的信息去掉
    keyLabel.config(text="")
    pattern = re.compile('[\u0020-\u007E]*')
    match = pattern.fullmatch(key)
    if match is None:
        return False
    return True

```

(3) 产生非法字符处理

首先, 将密码的输入框和再次输入框进行清空, 然后产生提示信息, 并将上次的错误信息清空。

密码格式不正确后的处理

```
def invalid():
    # 之前如果有不一致提示, 则在下次验证时把错误信息去掉
    keyLabel.config(text="")
    # 清空输入框
    entryKey.delete(0, END)
    entryAgainKey.delete(0, END)
    # 错误提示文本
    keyLabel.config(text="密码含有非法字符")
```

(4) 开始压缩函数

```
def ZipFileBegin(tree, settings):
    children = tree.get_children()
    srcFileList = list()
    for c in children:
        srcFileList.append(tree.item(c)["values"][2])
    dstFilePath = settings[0].get()
    key = settings[2].get()
    keyAgain = settings[3].get()
    if keyAgain != key:
        messagebox.showerror('Σ(っ ° ▽ °;)っ', '密码不一致, 请重新输入!')
        return
    zipLevelStr = settings[1].get()
    # 选择压缩等级, 默认为正常压缩 3
    if zipLevelStr == "0-不压缩":
        zipLevel = 0
    elif zipLevelStr == "1-快速压缩":
        zipLevel = 3
    elif zipLevelStr == "2-正常压缩":
        zipLevel = 6
    else:
        zipLevel = 9
    messagebox.showinfo("￣へ￣", "压缩已经开始, 请耐心等待...")
    pyminizip.compress_multiple(srcFileList, [], dstFilePath, key, zipLevel)
    srcFileSize = 0
    for file in srcFileList:
        srcFileSize += os.path.getsize(file)
    dstFileSize = os.path.getsize(dstFilePath)
    zipRate = "%(rate).4f" % {'rate': (dstFileSize / srcFileSize) * 100} + "%"
    messagebox.showinfo("o(*￣▽￣*)ブ",
        "压缩成功!!! \n 压缩前文件大小: %s B\n 压缩后文件大小: %s B\n 压缩率
为: %s" % (srcFileSize, dstFileSize, zipRate))
```

6 结束语

这次做大作业使我受益良多。一方面，这次大作业对于我来说还是有一定难度的，Python 从零开始，凭着自己不断地学习和各种百度谷歌，总算是完成了这次大作业的基本骨架内容，从零到一的过程是艰难的，但同时也是令人兴奋的。从这次的大作业开发中，我学到的更多的是 Python 的 GUI 开发，不断地探索窗口之间的关系，不断地查找控件的功能。通过这次实验的完成，我对 Python 有了更深入的理解，能够更加熟练的运用这门语言，更加理解到了编程的魅力。而另一方面，使我明白了认真做事的重要性。使我体会到完成一件事的喜悦。在开发的过程中，养成了随手注释，并且记录开发过程中的难点以及解决的方法，我将开发文档（README.md）附在了压缩包中。同时对软件版本进行控制。以后面对困难的时候，都能有直面困难并克服它的决心。本设计能够顺利的完成，也感谢老师的教导和同学们的互相帮助学习以及网络上的资源，使我能够很好的掌握和运用专业知识，并在设计中得以体现。也许这次的软件设计并不完美，功能也不是很复杂，但却真正是自己做的，学到了许多知识，这便是收获吧。在以后的学习生活中我一定会能有更好的发挥，克服自己的缺点，不断改进，不断成长！

开发环境：PyCharm 2018.2