

《Java 程序设计》

教师评价：

☐算法/实验过程正确； ☐源程序/实验内容提交 ☐程序结构/实验步骤合理；

☐实验结果正确； ☐语法、语义正确； ☐报告规范；

其他：

评价教师签名：

一、实验目的

运用面向对象程序设计思想，基于 Java 文件管理和 I/O 框架，实现基于图形界面的 GUI 文件管理器。

二、实验项目内容

- 1、实现文件夹创建、删除、进入。
- 2、实现当前文件夹下的内容罗列。
- 3、实现文件拷贝和文件夹拷贝（文件夹拷贝指深度拷贝，包括所有子目录和文件）。
- 4、实现指定文件的加密和解密。
- 5、实现指定文件和文件夹的压缩。
- 6、实现压缩文件的解压。
- 7、文件管理器具有图形界面。

三、实验过程或算法（参照 MVC 模式进行类的设计，实现的算法流程）

- 1、确定模式
- 2、确定算法
- 3、理清思路
- 4、搭建模型
- 5、代码实现
- 6、程序调试

基于 MVC 模式设计，确定了文件管理器的基本框架：

报告创建时间：

即先创建一个框架，确定需要把 UI 面板设计成几个部分才能让 UI 界面对用户友好，让用户在使用本文件管理器的时候能得心应手。从以下方面开始设计：

- 1、确定顶层容器，用来容纳所有内容
- 2、首先要确定界面分布，确定分为 4 个部分即状态栏、工具栏、操作栏以及主面板，为便于用户直接进行文件管理，又将主面板分为了导航栏和主显示操作栏。
- 3、目的：要实现对计算机内存的访问，文件或文件夹的创建、删除、赋值、加密、压缩、解密以及解压，那么为了把这些功能添加进去，初步确定将这些功能封装在按钮中，通过点击实现。
- 4、模型搭建，最后发现可能是受 windows 操作界面的影响，形成的模型框架满是 windows 的影子，那么干脆就按照 windows 的模式创建一个文件管理器。

框架设计：

- 1、代码设计时初步想法是设计一个主类来创建容器，搭建界面，实现按钮等的封装以及添加按钮出发操作，按钮需要哪些根据需求来添加。
- 2、还要设计一个功能类，来丰富文件管理器的功能
- 3、设计一个文件展示类，来完成文件的展示满足查阅需求

代码实现过程以及算法分析：

面板：

通过继承 JFrame 实现主面板的搭建，通过 setSize 等方法设计面板大小和最初显示位置。

因为已经确定要把管理器分布为四大部分，所以定义四个 JPanel 嵌板用于内容填充，而这四个嵌板要想按照需求进行分布，那么暂时可考虑运用自定义布局方式，便于自己控制嵌板的大小和现实位置。

初步实现功能，所以通过 setResizable()方法将窗口大小固定即用户不可通过最大化按钮最大化窗口，后续可通过获取用户计算机屏幕分辨率大小的方法完成最大化扩充。

因为主面板要分为导航栏和显示面板，所以我在第四大面板之上又添加两个 JPanel 嵌板，因为主面板内容可能会超过界面大小，那么就智能呢个通过加入滚动条来让文件内容可以滚动显示，时的文件内容显示完全，添加滚动条即借助于 JScrollPane 类以一个底层嵌板为基础添加，而需要滚动的内容就是现实在这个底层嵌板上，

就此，面板设计完成。

控件填充：

首先我考虑到对文件的操作，需要添加一些初步的按钮控件在工具栏，方便自己了解到需要完成那些任务，故在第二第三嵌板添加了文件、帮助、编辑、新建文件、新建文件夹等按钮，方便后续进行功能填充。

进入到文件管理器的主体--文件。文件作为文件管理器的主要部分，完成文件的罗列和添加操作是极其重要的，文件既要完成可视化罗列又要可以打开进入下一层文件，那么就要设置点击事件---使用按钮是一个毋庸置疑的选择。

在获取文件之前，我要知道计算机磁盘的分布情况，便于后续通过盘符分辨不同文件的所属地，完成文件归档和分类。所以我要访问计算机内存获取磁盘信息，Java 语言可通过 `File.listRoots()` 方法获取主磁盘信息，通过罗列并借助于 `JButton` 类将每个磁盘的名称添加到按钮上作为文本显示，通过循环自动递增按钮位置，调用针对按钮的 `setBounds()` 方法设置了按钮的位置和大小，循环设置按钮时通过位置坐标的累加使得按钮紧邻而不重合，精确的实现了主盘符在面板上的显示，而主盘符需要添加的功能是进入对应磁盘分区，所以这些盘符信息一定显示在首页，第四面板分为导航栏和主面板，导航栏要完成导航功能即在客户需要的时候快熟进入对应磁盘，所以主盘分区要永远显示在导航栏。同时，主显示面板也要可以显示主盘符分区信息，便于用户选择和操作，因此通过对应的嵌板调用 `ass()` 方法，将盘符信息通过 `ListRootd()` 方法存储在一个 `File` 数组中，通过 `for` 循环将文件对象罗列并添加到两个面板中，将这一操作封装为主类的 `showHomepage()` 方法，因为后期还要从其他目录返回主页进行操作，所以封装为函数是为了后期操作。

完成了主页的展示，就要进行进入对应分区，展示文件内容功能的封装。如何进入对应分区查看文件？因为文件就在对应磁盘分区之下，而盘符已经被我以按钮形式封装与面板之上。所以我设计通过点击盘符按钮进入对应磁盘，即为对应按钮添加点击事件，而如果使用鼠标监听器，那么我不需要用的方法也要进行实现，增加了代码量，所以我使用鼠标事件适配器，只实现需要的方法，所以引入了 `java.awt.event.MouseAdapter` 等包来实现点击事件，通过实现事件之下的 `actionPerformed` 方法来添加响应事件。

那么要如何进入对应的磁盘空间就是新的问题，其实就可以直接将盘符通过 `getPath()` 方法取得路径，通过路径创建文件实例，再通过 `listfiles()` 方法取得对应分区之下的文件数组，就可以取得对应分区的文件信息了，然后通过 `getName()` 方法取得对应文件的名称，通过创建按钮并把文件名作为按钮名就可以实现文件的可视化创建和操作了。同理，通过递归实现，就可以实现实现对每一层文件的访问和查看。

添加返回上一层和进入下一层按钮事件：

因为有时候可能会进入某个文件夹想要返回上一层，所以添加一个上一层按钮是必要的，那么如何返回上一层，我就定义一个全局字符串变量 `text`，用它来记忆当前路径，要想返回上一层，那么就只要直接将路径字符串按照文件路径分隔符分割然后把最后一部分去掉就是上一层路径，所以通过 `string.split(File.separator)` 方法实现字符串分割，通过文件创建实例化，再进入到对应目录之下即可。同理进入下一层是现在使用返回上一层的基础上使

用的，要想进入下一层，就必须执行过返回上一层事件。点击进入下一层实现下一层的进入。

文件目录更换时在面板上的展示：

要想更新展示内容，就要通过 `panel.updateUI()` 方法和 `panel.removeAll()` 方法来实现，更新面板内容和删除所有内容，`updateUI()` 方法就是将你新添加的空间展示出来，否则当前显示的还是就面板的内容。而后再通过文件列举方式进入文件夹，将文件目录内容展示出来即可。

添加新建文件或新建文件夹：

在第三面板设置了新建文件和新建文件夹的按钮，当点击时就是在当前面板（路径）下新建内容，就要为按钮添加响应事件，如何获取当前面板路径？前面提到的 `text` 字符串全局变量就记录了当前路径，所以就可以直接在当前路径下创建文件或文件夹了，需要用户输入新建文件或文件夹的名字，如果是文件则必须要输入文件的后缀名，否则得不到相应的文件。创建结果通过对话标签告知用户。

添加文件处理操作：

操作有文件删除、复制、压缩、加密、解压、解密。设置为右键单击文件或文件夹实现菜单栏的罗列，并且为菜单栏的按钮添加了相应事件处理操作，通过点击按钮可以实现相应的功能。

操作方法封装在另一个类 `fileOperation` 里面，通过继承主类 `manager`，所以可以使用主类定义的方法和变量，并可以对主面板作出相应的更改。

文件压缩对文件是直接压缩，对文件夹是递归压缩，最后得到一个压缩包保存在当前目录之下。文件加密只支持对文件的加密，不支持对文件夹的加密，所以如果为文件夹加密会提示相应的错误信息。

所有的文件处理操作都会有相应的提示对话告知用户操作结果。

显示当前路径：

为了便于用户了解当前在那一路径，我将全局变量 `text` 保存的路径通过一个在第一面板的 `JTextArea` 文本框显示出来，便于用户直接查看和了解当前处于的路径。

并且添加了文本点击全选事件，便于用户复制路径。

文件管理器的优化：

按钮文本直接设置时文本默认居中且没有图标，所以通过 `ImageIcon` 类调用对应的给定图标作为按钮的图标，使得对用户更加友好，并通过 `setHorizontalAlignment()` 方法设置按钮文本对齐方式，设置文本左对齐，并对全局按钮做此操作。实现界面优化。

文件管理器教程：

因为管理器是我写的，所以要告诉用户如何使用我的文件管理器和告知一些必要的注意事项，所以通过设置一个对话框罗列了注意事项和使用教程，在用户启动文件管理器的时候显示出来供用户阅读，可以关闭或点击进入实现管理器的使用，还为”帮助“按钮添加了响应事件，就是用来展示此内容，方便用户查阅使用教程和注意事项。

文件打开查阅：

如果是文件夹那么就进入文件夹查看下层文件并罗列出来，如果是文件的话内部只有文件内容没有下层文件包含，所以文件我要提供打开操作，当点击文件时，就清空面板，设置一个文本域用于显示文件内容，且可以滚动阅读，并添加了自动换行功能。并将此模块封装在继承主类的 openShowfile 类里面。实现了文件的阅读，可以点击返回上一层按钮返回上一层并退出文集爱你阅读模块。

些许模块暂未写入，可后续添加。

总述：文件管理器操作已经完成，对每一部都有对应的文件处理和结果报告以及添加了必要的错误处理和接纳能力。

文件管理器设置完毕。

四、实验结果展示及分析和（或）调试过程分享

所遇问题以及解决办法：

问题：提示对话框显示的问题，要显示在屏幕中央且必须等待用户处理才能进入下一步操作：

解决办法：

使用 java.awt.toolkit 包下的 getscreensixe()方法或去屏幕大小并实现居中显示。

问题：文本标签在文本对话框显示的时候换行问题，普通\n不起作用

解决办法：越来越要使用<html><body><body></html>方式实现换行，这样设置是到达边界换行，如果自己换行则要加入
来实现换行。

问题：滚动条不会随着内容多少自动变换的问题

解决办法：计算内容数量，通过内容数量完成对滚动条和面板大小的设计，实现可以滚动查看。

问题：按钮的边框和填充色的显示问题

解决办法：通过 setBorderPainted()和 seContentAreaFilled()方法实现边框和填

充色的设计

问题：对于 panel 嵌板 setSize()方法不起作用

解决办法：必须要使用 setPreferredSize()方法才能改变 panel 嵌板的大小

问题：文件删除后使用 uodateUI()方法还是没完成内容的刷新

解决办法：因为删除文件后现在对内存的访问还在于刚刚的状态，所以刷新面板并没有实际对内存情况进行访问，所以要重新进入当前目录即完成内存刷新才能看到实际的文件目录情况即真实路径下的内容展示。

试验结果及内容展示：

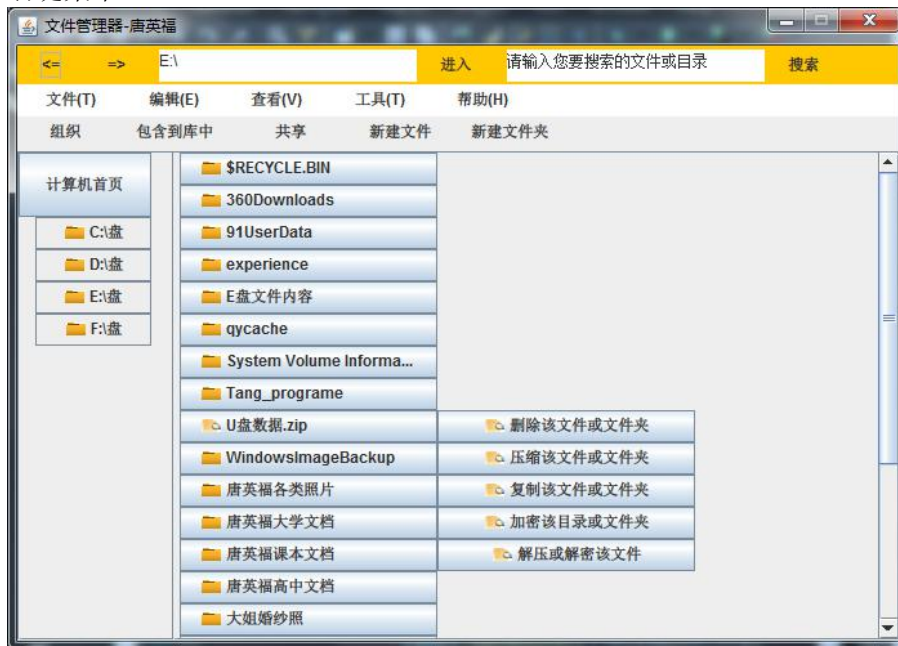
面板总览：



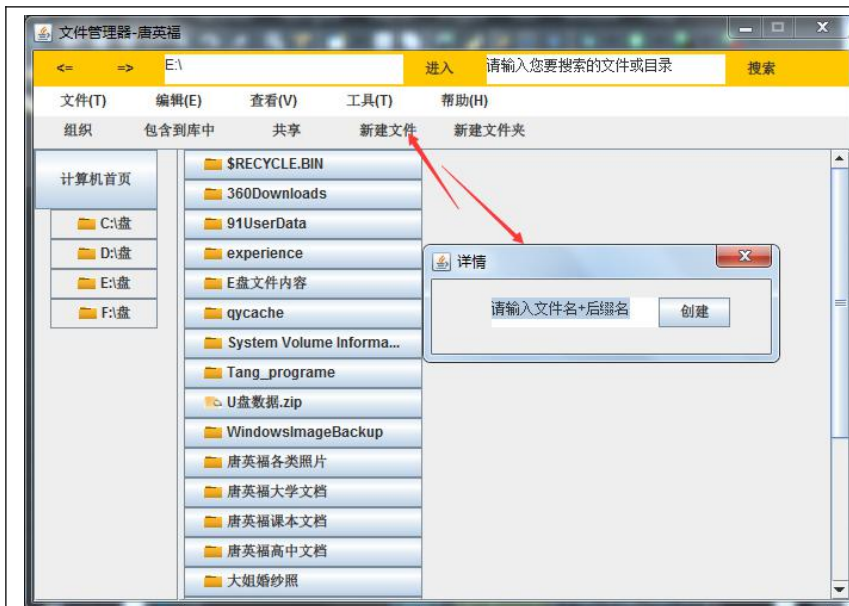
首页：



右键菜单:



新建文件:



文件操作实例-以文件压缩为例:



实现文件打开阅读:

文件管理器-唐英福

<=> E:\唐英福大学文档\大学作业文档\主题 进入 请输入您要搜索的文件或目录 搜索

文件(T) 编辑(E) 查看(V) 工具(T) 帮助(H)

组织 包含到库中 共享 新建文件 新建文件夹

计算机首页

C:\盘 D:\盘 E:\盘 F:\盘

Email or letters: which is the better way to contact?

I would much writing letters rather than emailing to others. While sending an email is definitely com mon in our daily lives, I'd love to write words in paper and post to others. Promoting emotional com munication, saving money and free time, focusing on the loser of Internet are the reasons why I love writing to others.

When people reading a letter, they will get the true emotions in it. 人们可以通过信中的笔迹联想到写信人写信时的感情和状态，感受到其中的深意，而邮件内容仅仅是传递消息的冰冷的打印字体。人们可以在书信中加入很多个性化的小标记来表达情感，这在邮件中是没有的。

写信比起发邮件更便宜和节约空闲时间。写信的人都会在一封信中将需要传递的消息全部写下以告知对方，寄信的费用通常是固定在0.5元左右，一旦信被寄出写信人在收到回复之前就不必再考虑这件事，而人们通过写邮件传递消息时需要根据字数支付相应的传输费用，往往人们不可能在一封邮件中把信息完全传递，需要回复对方对一些细节的询问，因此要花更多的时间去解释邮件信息，因此写邮件最终的传输费用大约1.0元，高于写信。

写信更会关注到那些不能使用网络的人们。我们都知道，信件交流只需要有人把信交到收信人手中而不需要其他媒介而且当今的人们大多数都能通过信件交流，与此不同的是邮件必须借助于网络来实现，在一些网络未覆盖的地区人们无法接收到邮件信息，而且一些老年人根本就不会使用网络设备接收信息。比起写邮件，我更喜欢写信件与人交流，尽管平时出于方便我会给人发邮件。写信这一传统交流方式永远不会过时而且会一直收到人们喜欢。

Email or letters: which is the better way to contact?

结束实验。

实验结果：满足实验要求。