**课程设计报告**

**题 目 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**课程设计**

**—— 信息处理与编码**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**7**  \_7\_\_1\_

**1**\_\_

**2022**

**\_\_\_\_\_年 \_\_\_月 \_\_\_日**

目 录

[1 需求分析 5](#_Toc107482959)

[2 设计思路分析 5](#_Toc107482962)

[2.1 信道编码总体设计 5](#_Toc107482963)

[2.2 信道编码各个部分设计 5](#_Toc107482964)

[3 系统实现 1](#_Toc107482966)0

[3.1 开发平台与工具 1](#_Toc107482967)0

[3.2 程序设计 1](#_Toc107482968)0

3.2.1 程序设计的总体思路............................................................................................10

[3.2.2香农编码.............................................................................................................](#_Toc107482968)....11

[3.2.3费诺编码..............................................................................................................](#_Toc107482968)...13

[3.2.4 LZW编码.................................................................................................](#_Toc107482968)...............15

[3.2.5](#_Toc107482968) 算术编码................................................................................................................20

[3.2.6 霍夫曼](#_Toc107482968)编码实现图像压缩解压............................................................................23

[3.2.7 游程编码](#_Toc107482968)实现图像压缩........................................................................................24

[4 系统测试 2](#_Toc107482971)6

[5 总结](#_Toc107482972) 34

[5.1 本次程序设计的优点： 3](#_Toc107482973)4

[5.2 本次程序设计的缺点：](#_Toc107482974) 34

# 需求分析

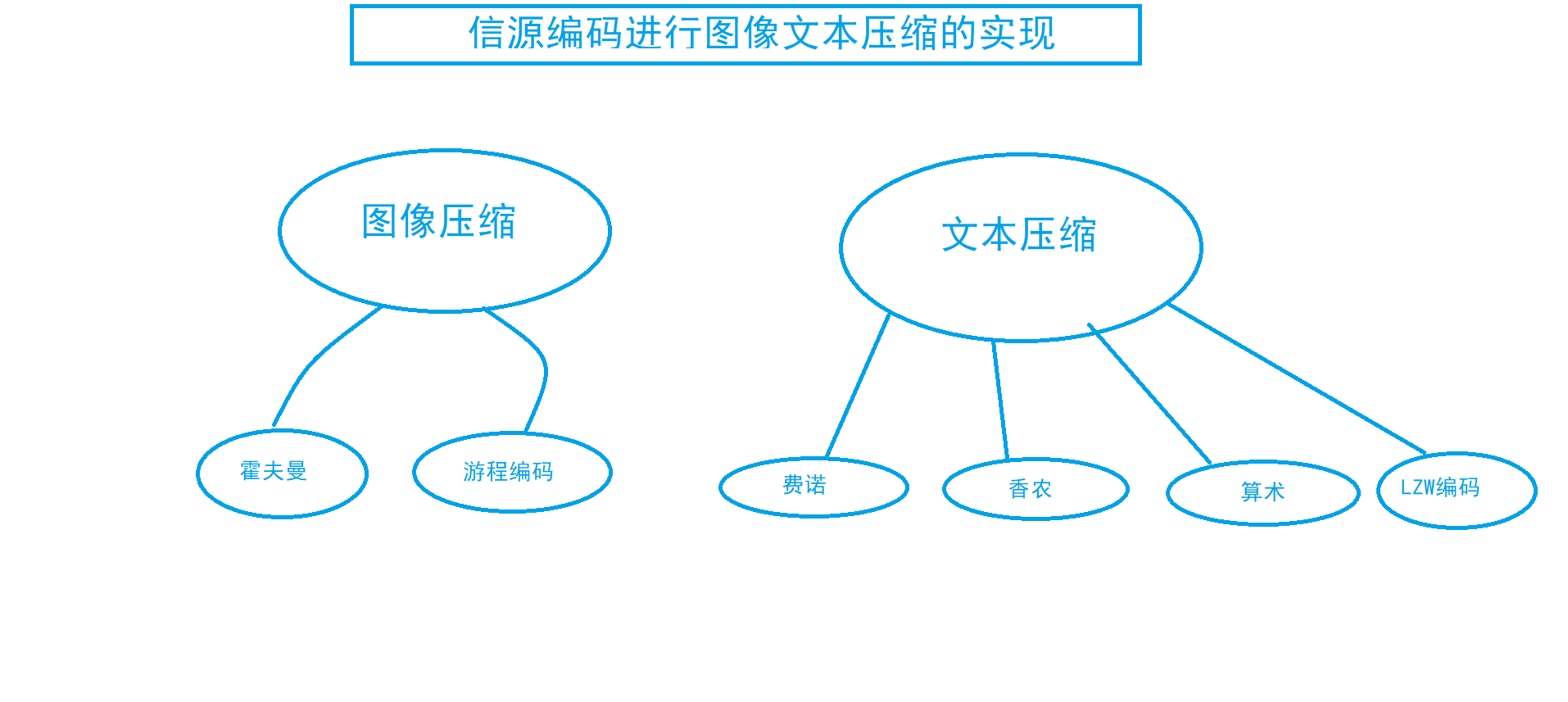
本次程序设计需要实现信源编码对图像，文本进行压缩，具体细节如下：

1. 实现GUI界面，能够从界面读取图像，文本，概率分布等等
2. 霍夫曼编码实现对图像的压缩和解压
3. 游程编码实现对图像的压缩
4. 费诺编码实现对文本压缩
5. 香农编码实现对文本压缩
6. 算术编码实现对文本的压缩
7. LZW编码实现对文本的压缩

# 设计思路分析

首先根据要求，设计出能具体读取图像，文本，概率分布的界面，然后根据其他队友的代码，创造输入输出“接口”，即将文本，图像等信息输入到具体代码里，将编码之后的结果反馈给界面并展示出来。

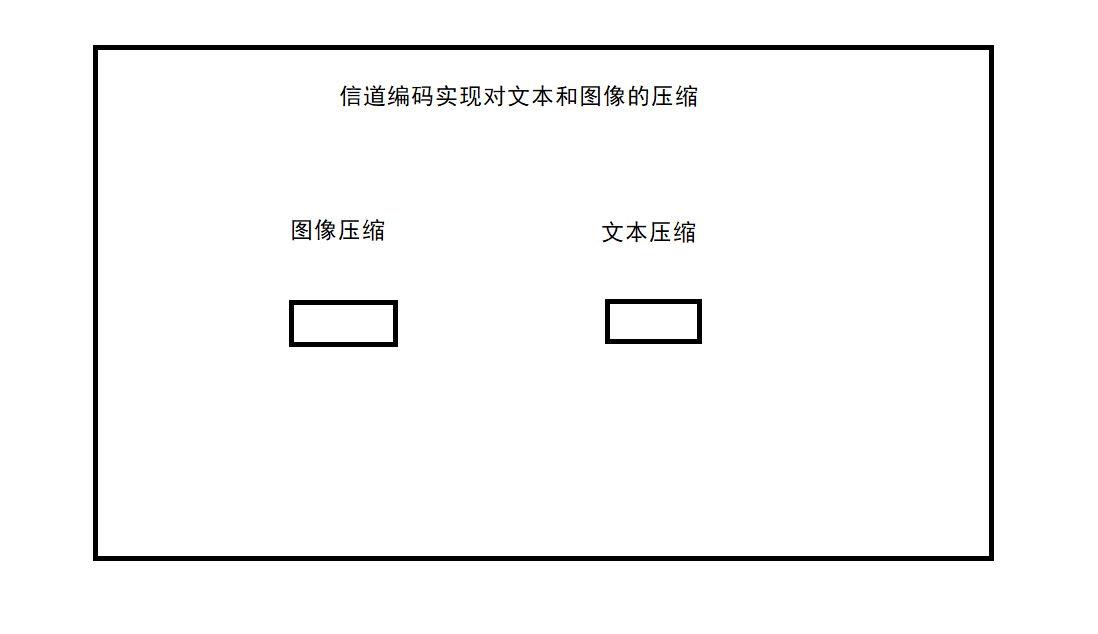
## 信道编码总体设计



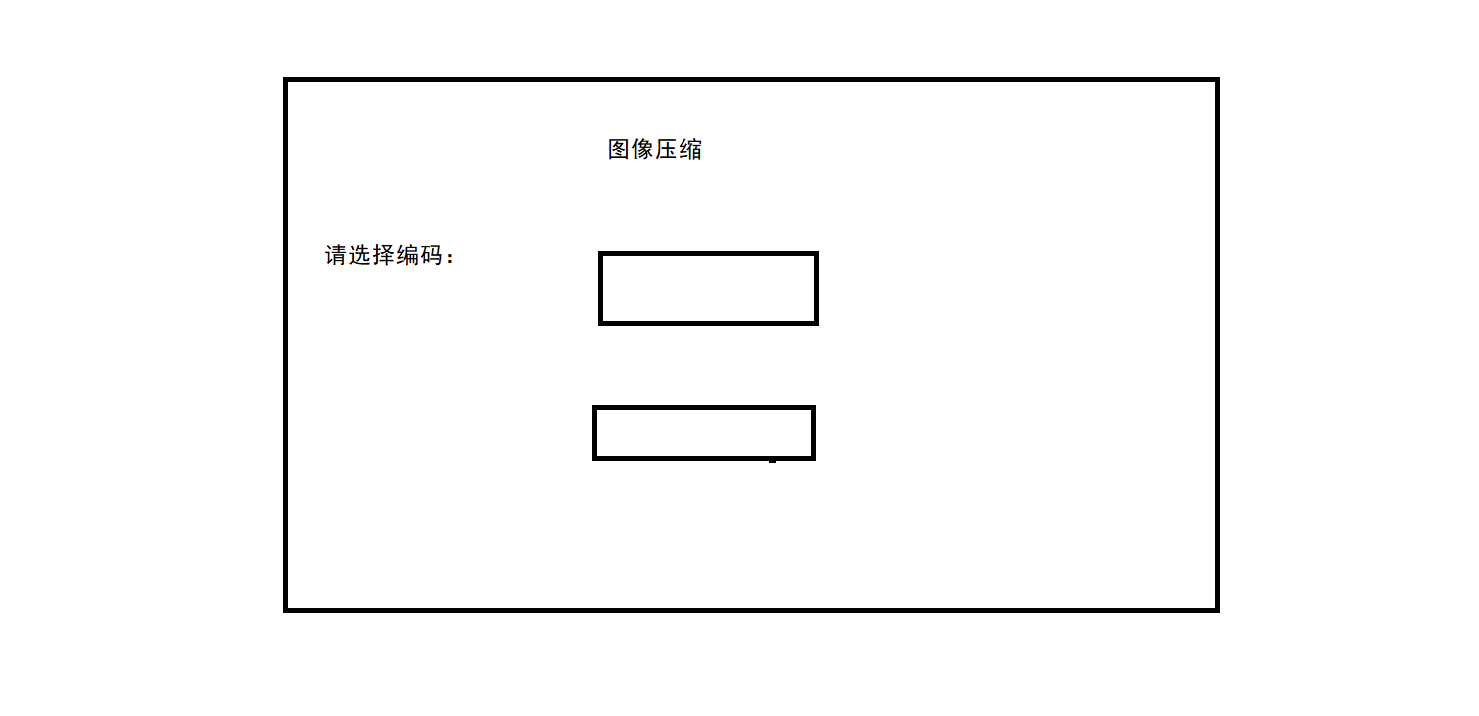
## 信道编码各个部分设计

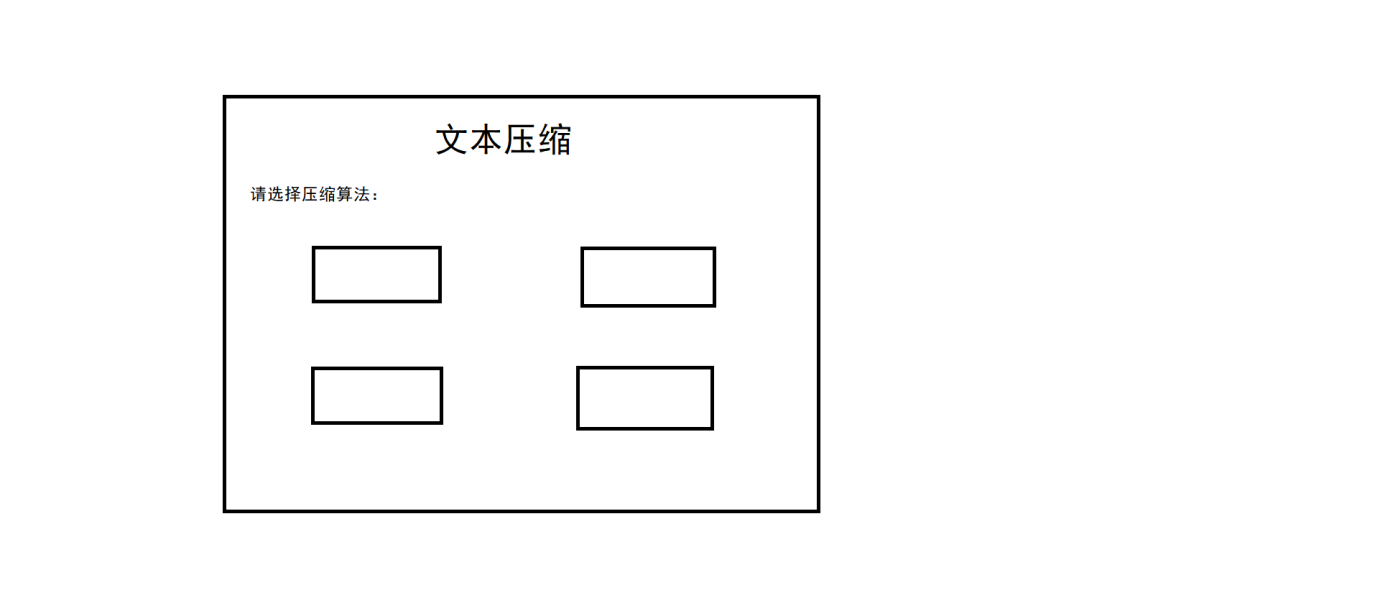
**1.首页设计：**

要包含两个部分，分别为：文本压缩和图像压缩

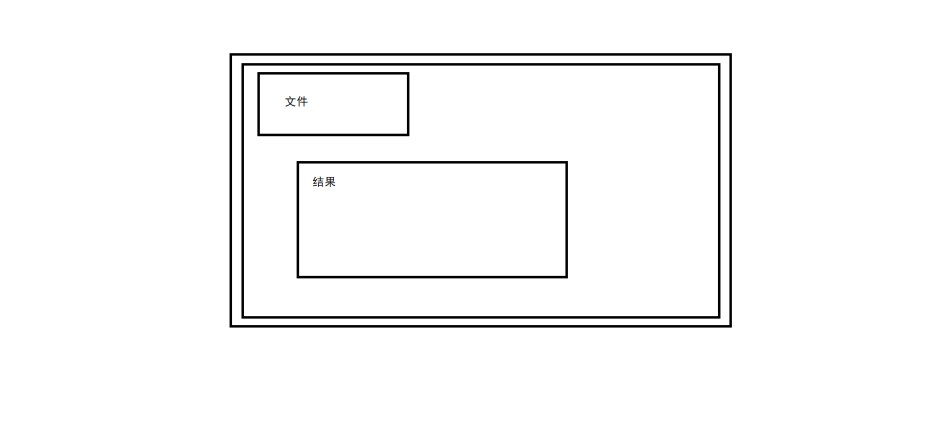


分别选择点击两个按钮之后：



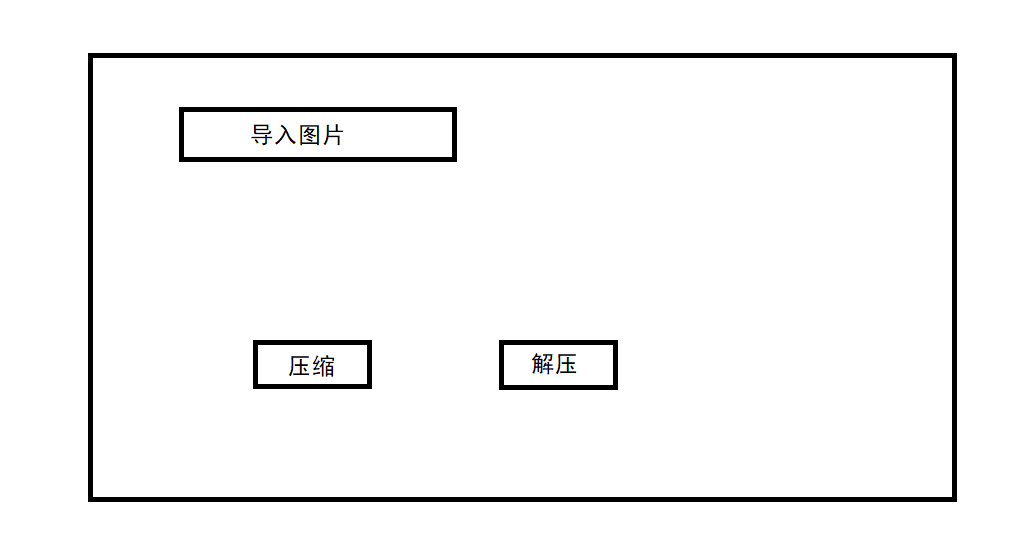


**2.游程编码设计：**将要编码的图像使用选择框，找到位置，然后提交按钮进行编码，再将结果展示在GUI界面上



**3.霍夫曼编码设计：**

同游程编码，选择要压缩或者解压的图像的路径，然后进行编码或者解码，最后将编码之后的压缩率等参数展现在GUI界面上。



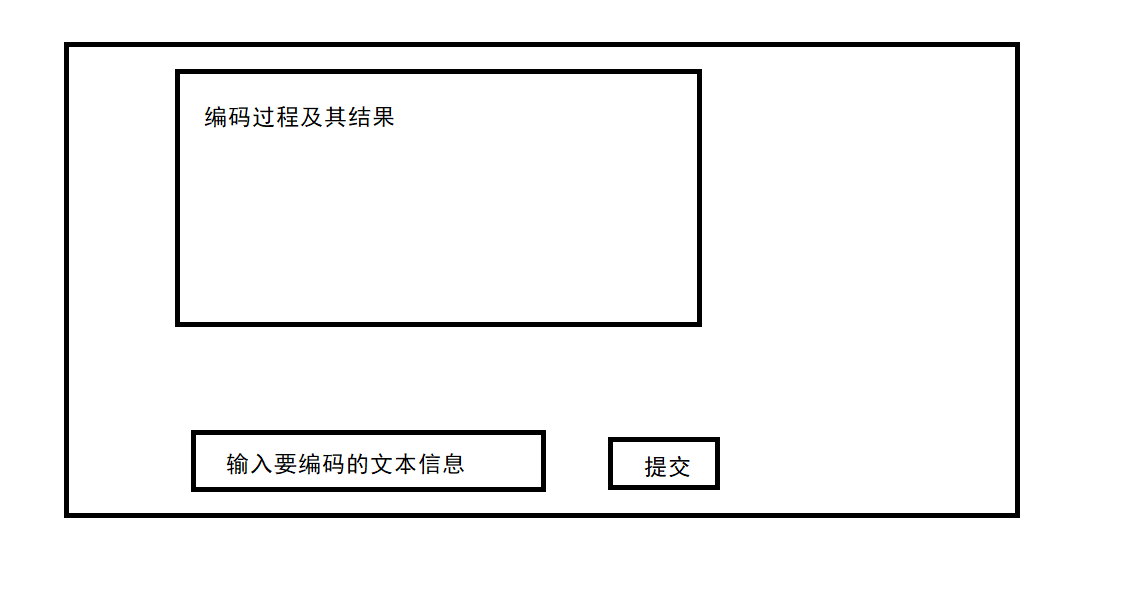
**4.算术编码设计：**

分别通过五个文本框和五个按钮提交输入要提交的文本信息，概率分布等等，然后进行编码，最后将编码之后的所有信息展示在GUI界面上。

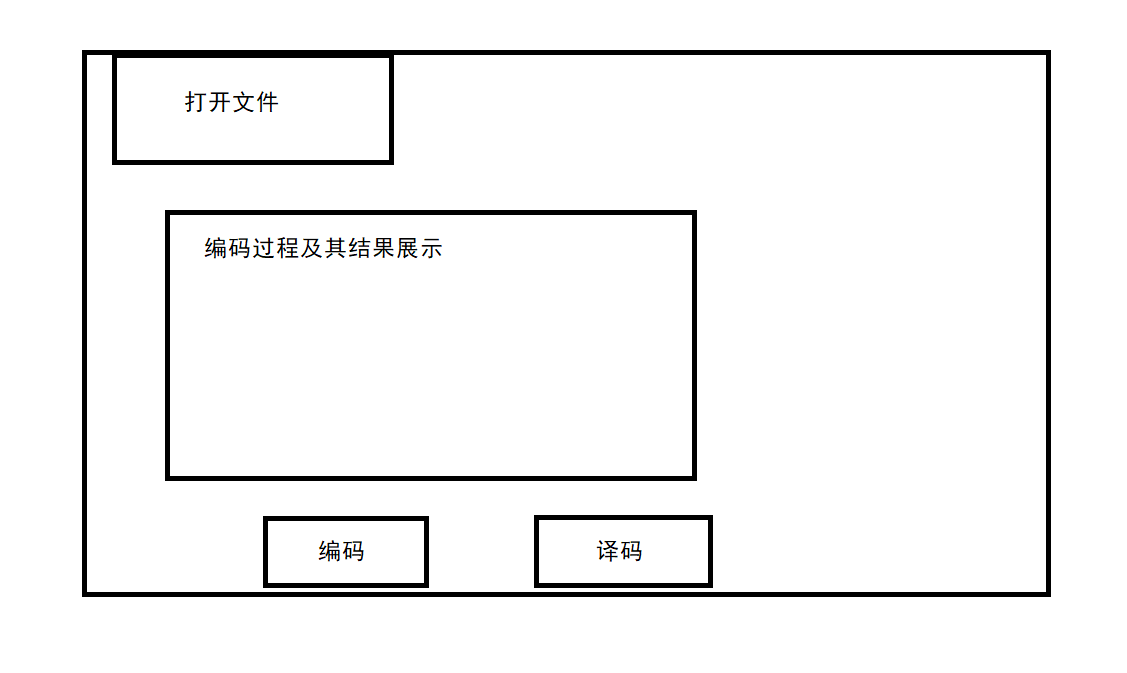


**5.费诺和香农编码设计：**

通过一个文本框输入编码的字符串，然后通过按钮提交，编码之后的结果展示在一个文本域中



**6.LZW编码设计：**  
设计一个下拉列表，有选项打开文件，然后找到要打开的文件的路径，打开之后，选择编码或者译码两个按钮，之后将结果保存在编码或者译码之后的文档中，然后打开查看。



# 系统实现

## 开发平台与工具

本程序使用JAVA语言实现GUI界面以及各种算法，所有代码最后集中在WIN10上使用eclipse开发环境，JDK1.8，测试成功。

## 程序设计

## 3.2.1程序设计的总体思路

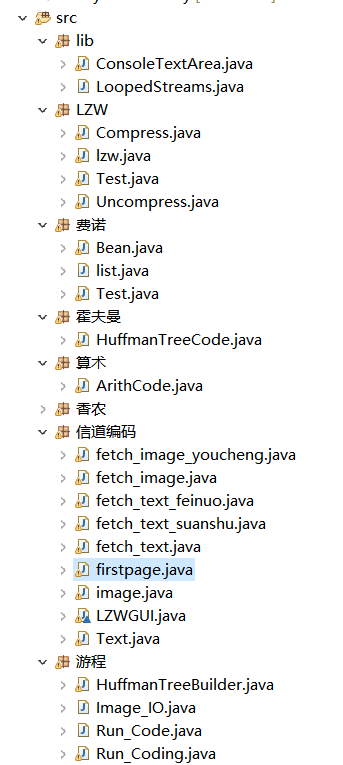
本程序的总体目录结构如下：

LZW，费诺，香农，霍夫曼，游程，算术 6个包分别存放队员的算法实现代码，

信道编码包 存放各个编码方式的具体GUI界面，

Lib包存放两个工具类，用于实现LZW编码时，获取控制台的结果

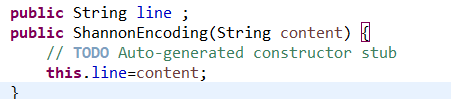
总体界面程序设计思想是：将文本或者图像信息传入相应的编码算法中，在将编码之后的输出结果抽取到GUI界面中来。



## 3.2.2香农编码

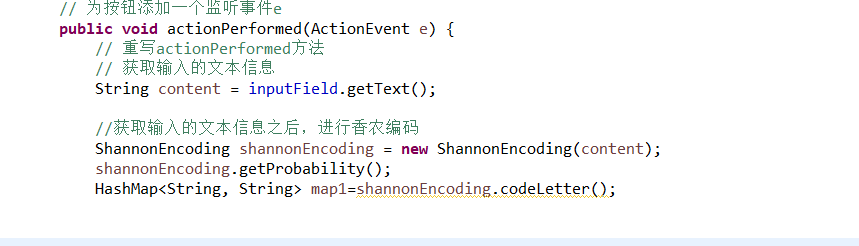
**输入从控制台转移到GUI：**

修改代码：ShannonEncoding.java类中的从控制台输入字符串修改为



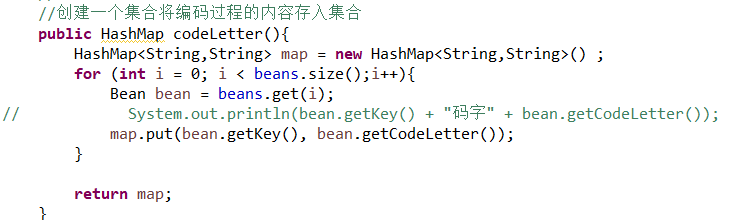
即定义一个变量，然后写一个当前类的构造方法，将该变量初始化，这里就可以作为从外界获取输入信息的“接口”，传入信息到类中。

首先建立香农编码界面：参见fetch\_text.java，添加一个文本框，一个按钮，一个文本域，为按钮添加事件，当在文本框上输入信息时，点击按钮提交，将文本框中输入的信息捕获然后创建ShannonEncoding.java类的对象，在构造方法中传入捕获的值，传入编码类中运算。



**输出从控制台转移到GUI：**

在ShannonEncoding类中添加一个方法，返回一个map集合，该集合中存储我们需要的编码过程的代码，然后传出到GUI界面上，接受信息之后，添加到按钮的点击事件中去，将这些信息添加到文本域中，即可显示。



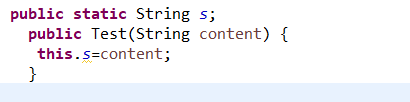


## 3.2.3费诺编码

**输入从控制台转移到GUI：**

**首先编写费诺编码的GUI界面，即**fetch\_text\_feinuo.java类。该类和香农的内容相似，一个文本框输入信息，一个文本域展示信息，一个按钮提交当前输入信息到编码类中。

修改费诺编码类，写一个变量s，然后写构造方法，作为输入的“接口”。

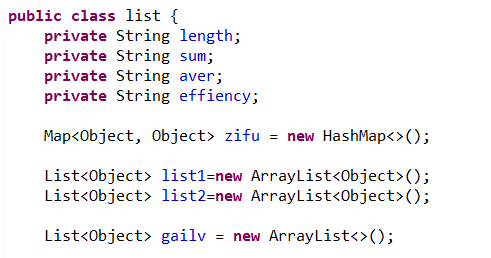


然后给按钮添加事件，将文本框中输入的信息捕获传入到编码类的构造方法中去。

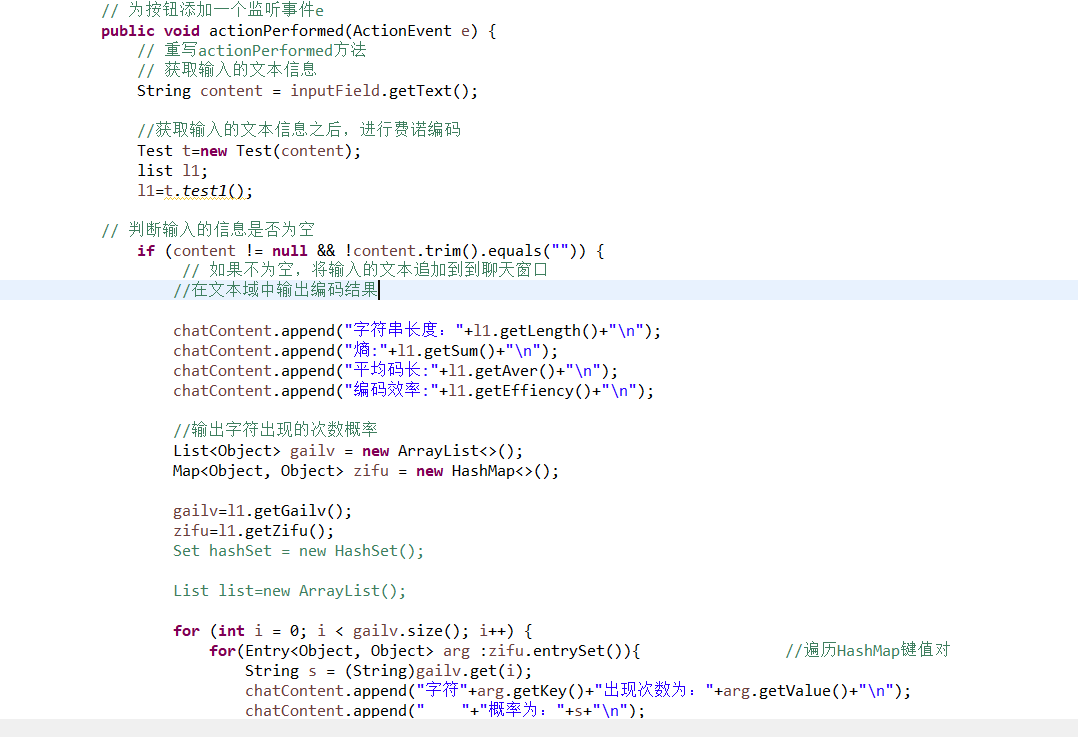


**输出从控制台转移到GUI：**

考虑到编码类中编码过程的输出结果比较多，建立一个list类，添加一些变量和集合用于存储编码类中产生的结果：



然后在编码类中修改，将所有输出的结果全存入定义的变量中，然后封装成list的一个对象，传出，在按钮的事件中添加一个list对象接受传来的值，然后分别使用各种循环方式将所有结果添加到当前界面的文本域展示信息区域。





## 3.2.4 LZW编码

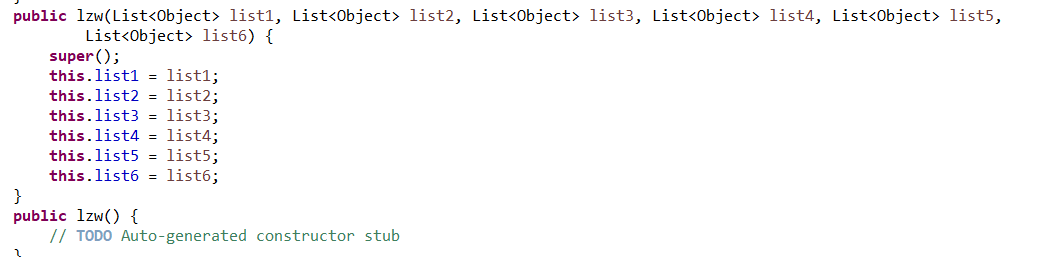
首先编写LZW的GUI界面，LZWGUI.java类，添加一个下拉列表，能打开，保存，退出文件，一个文本域显示打开的文本，两个按钮分别表示编码和译码。

**编码：**

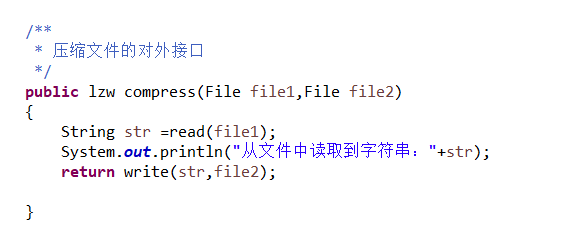
**将打开的文件路径从代码转移到GUI**，为按钮编码添加事件，将当前选择打开的文本信息的路径捕获，然后路径封装成一个字符串，然后创建一个编码类的对象，然后将这个路径传入其构造方法，即可完成传入文件，然后编码实现。

**将编码过程的显示从控制台转移到GUI界面，**首先由于输出结果比较多，创建一个lzw.java类，添加6个集合，用来存放数据，然后写有参和无参的构造方法，以及所有集合的get，set方法，





在编码类中分别将所有的输出结果传入到定义的6个结合中，然后使用一个方法，返回所有集合封装之后的对象，



在为按钮编码定义的事件中接收这个对象，然后将所有的集合分出来全部遍历出来，添加到文本域中展示：



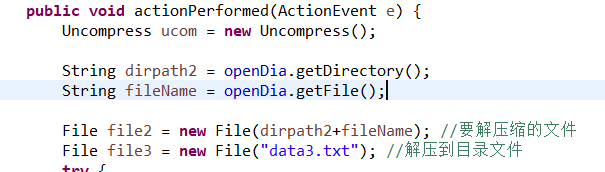




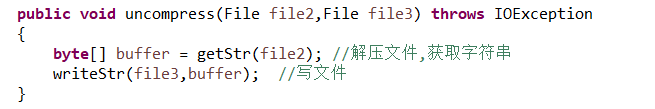
**译码：**

**将传入的路径从译码代码中转移到GUI界面：**

将打开的文件路径封装为一个字符串，然后通过译码类的方法传入译码类中，



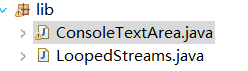
在译码类中接受路径，然后译码：



完成后传出结果。

**将译码过程展示在GUI：**

由于输出的内容过多，经过查阅资料之后，找到了一种方式可以将控制台输出的结果传出到一定的界面上。



使用上面两个类，在ConsoleTextArea.java类中修改：在test方法中写一个界面作为我们当前输出的结果展示，然后将译码过程的代码写入，就能将执行译码之后的控制台输出结果展示在当前弹出的界面上。



然后在界面的代码中，为译码按钮添加点击事件，创造ConsoleTextArea类的一个对象，然后调用当前对象的test方法，就能实现在我们的译码界面弹出弹窗，显示控制台输出的信息。



## 3.2.5 算术编码

首先写一个算术编码的界面，fetch\_test4.text，该界面包括一个文本域用来展示当前输入输出的各项信息，然后五个文本框和五个按钮，用来提交输入的信息。

**将输入的信息从控制台转移到GUI：**  
由于编码算法在实现的过程中耦合性太太太高，这部分花费了大量精力，在经过构造方法，修改全体代码等等之后，最后终于成功，解决办法是，将要输入的五个参数定义在类中，然后分别写他们的set方法（如下图），然后将编码过程封装在一个test1方法中：





这样之后，就可以在界面类中，写一个编码类的对象，然后在为每个按钮添加事件，分别获取五个文本框中输入的信息，然后将其封装为特定的类型，这个类型要和在编码类中定义的参数的类型相同，然后使用编码类对象的set方法，将从界面文本框输入的信息传入到编码类中，最后一个按钮的事件中要额外添加一点，就是调用上面编码类中的test1方法，然后就能实现编码。



**将输出的信息从控制台转移到GUI：**

由于要输出的信息太多，并且需要按照一定的排列方式，所以使用之前使用过的两个工具类，直接从控制台获取信息，输出到当前界面上，但是由于编码方法的实现已经安排在了按钮5的点击事件中，难以取出，尝试过很多方式之后仍然失败。

## 3.2.6 霍夫曼编码实现图像的压缩与解压

首先，写霍夫曼界面，fetch\_image.java，一个浏览按钮能获取当前选择的图片路径，一个解压，压缩按钮分别能实现图像的压缩和解压。

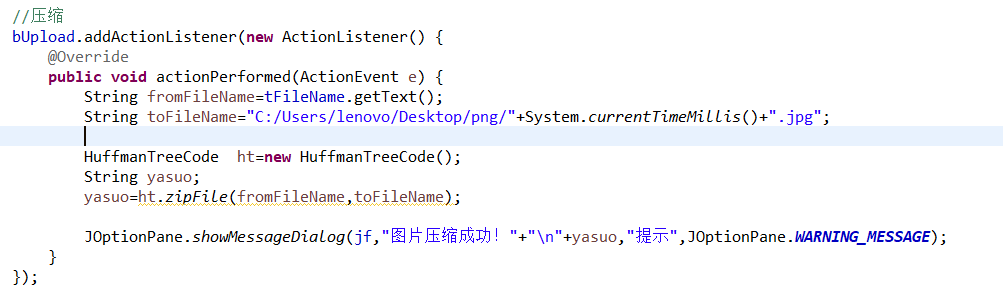
**压缩和解压过程相同：**

**输入图片路径信息从代码转移到GUI界面：**

由于霍夫曼算法实现的代码中提供了良好的入口，



直接在界面类中为压缩按钮添加点击事件，能将当前选择图片的路径封装为一个字符串，创建编码类的对象，然后将该路径传入编码类，即可完成压缩，压缩后的路径设置在了桌面的png文件夹中。



**压缩完成后信息输出在GUI界面上：**  
在压缩完成界面上写一个弹窗，然后显示压缩信息：



这个压缩信息从编码类中传出，同样的，编码类的两个方法，压缩和解压方法分别返回字符串，然后我们在界面类中为压缩解压按钮添加事件时，定义字符串将两个方法返回的信息接收，然后通过上面定义的弹窗输出。



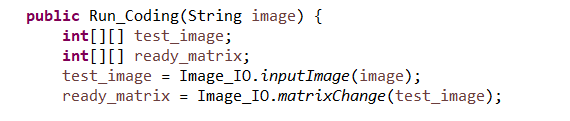


## 3.2.7 游程编码实现图像的压缩

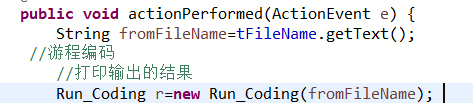
先写一个游程编码的界面类，fetch\_image\_youcheng.java 一个浏览按钮获取选择的图片路径，一个压缩按钮实现压缩功能，一个文本域展示压缩后各个像素点的编码结果。

**将输入图片的路径从代码转移到GUI：**

由于编码类的入口清晰，

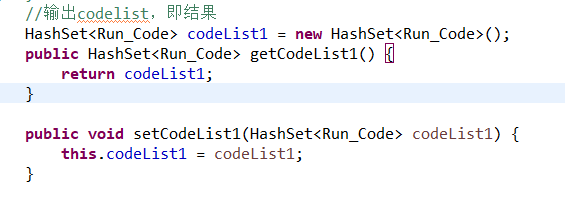


所以直接在界面类的压缩按钮事件中写入，将选择的图像路径封装为一个字符串，然后创建Run\_Coding类的一个对象，通过其构造方法传入图片路径，然后就能实现编码。

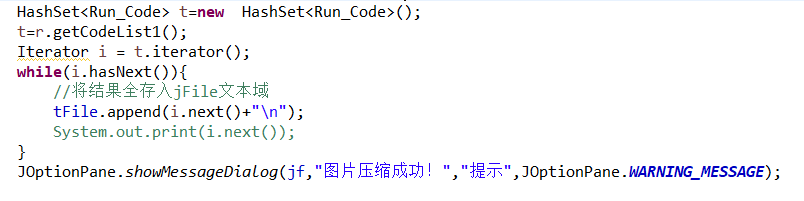


**将压缩后的结果信息从控制台转移到GUI界面：**

在编码类中写一个集合，存储输出结果，然后写其set，get方法



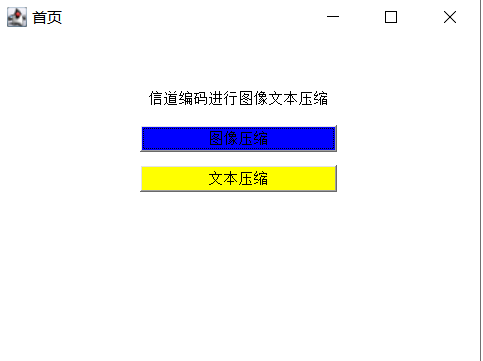
然后将结果通过get方法传入到编码按钮的单击事件中定义的一个集合中，然后将这个集合的所有值遍历，加换行，然后添加到文本域中展示出来即可。



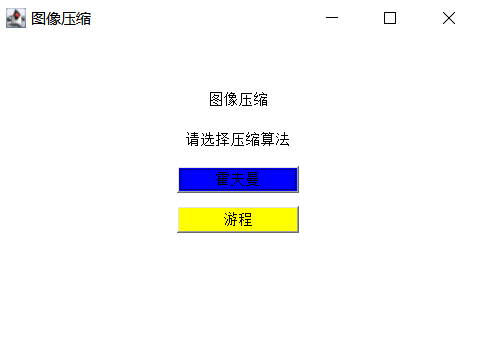
# 系统测试

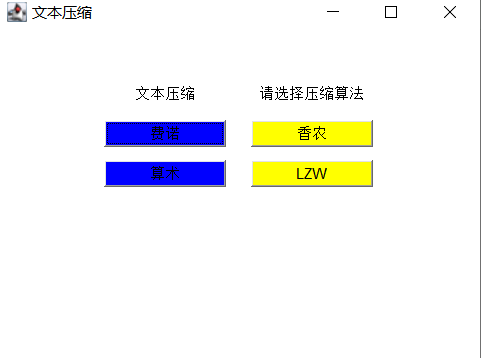
**1.主界面测试**

运行firstpage.java之后，出现首页界面如下：



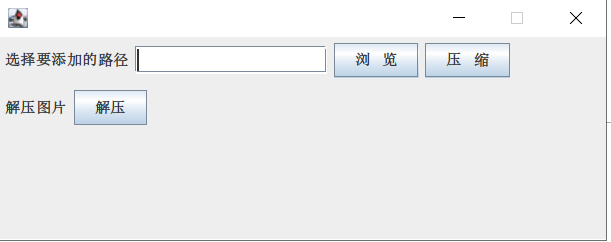
分别点击两个按钮之后：

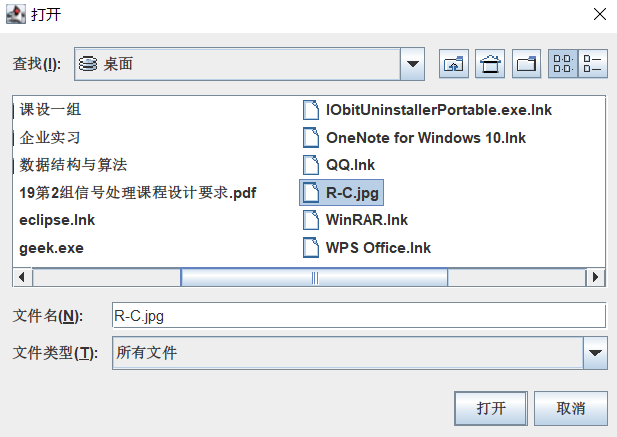




1. **霍夫曼实现图像压缩和解压过程**

首先点击浏览按钮，出现当前电脑的文件目录，找到当前要压缩的图像信息，确定之后，点击打开，即可选择当前图片路径，然后点击压缩按钮，就可以实现图像的压缩。压缩成功之后出现弹窗提示压缩成功，压缩偏差率等信息，然后文件保存在桌面的png文件夹里。





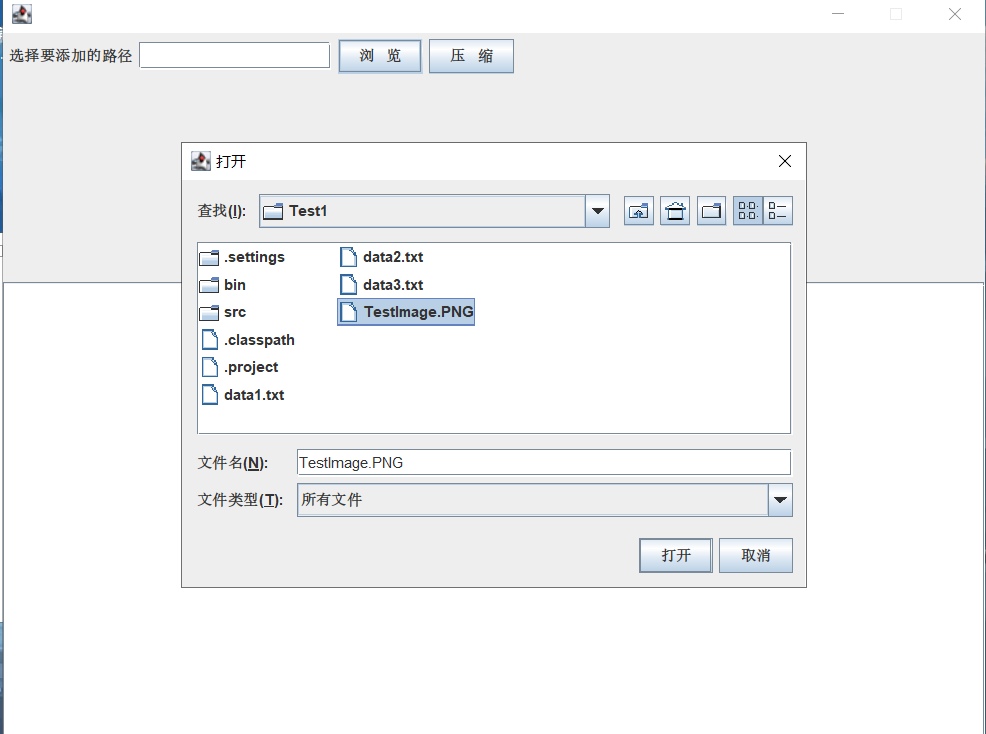
进行压缩之后：  
 

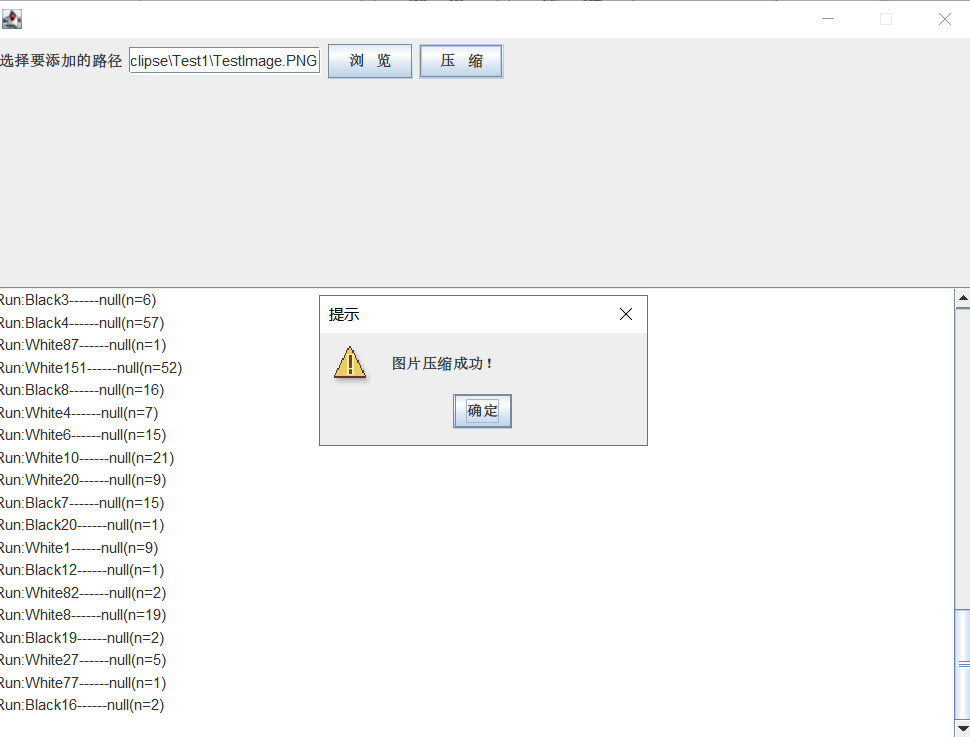
霍夫曼实现图片的解压过程：选择要解压的图片路径，然后点击解压按钮，等待小会，即可出现弹窗提示解压成功，解压后的文件以一定格式命名，然后保存在桌面上。



1. **游程编码实现图像的压缩：**

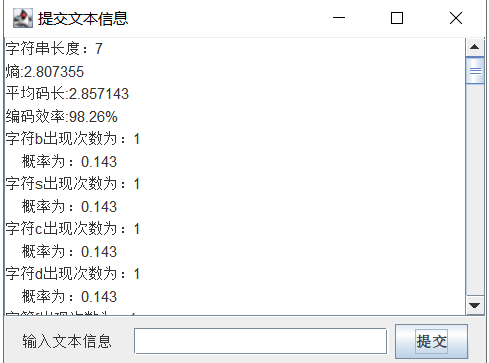
首先找到压缩的图像的路径，然后点击打开，在压缩，完成后弹窗显示压缩成功。

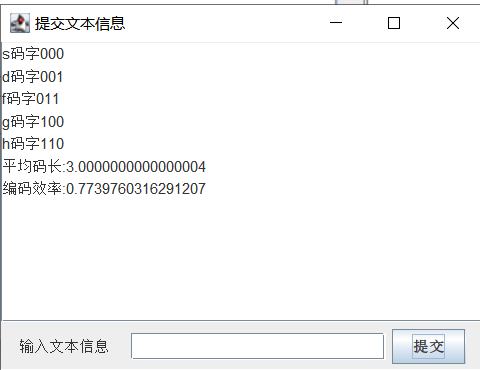




**4.费诺编码和香农编码实现对输入字符串的编码**

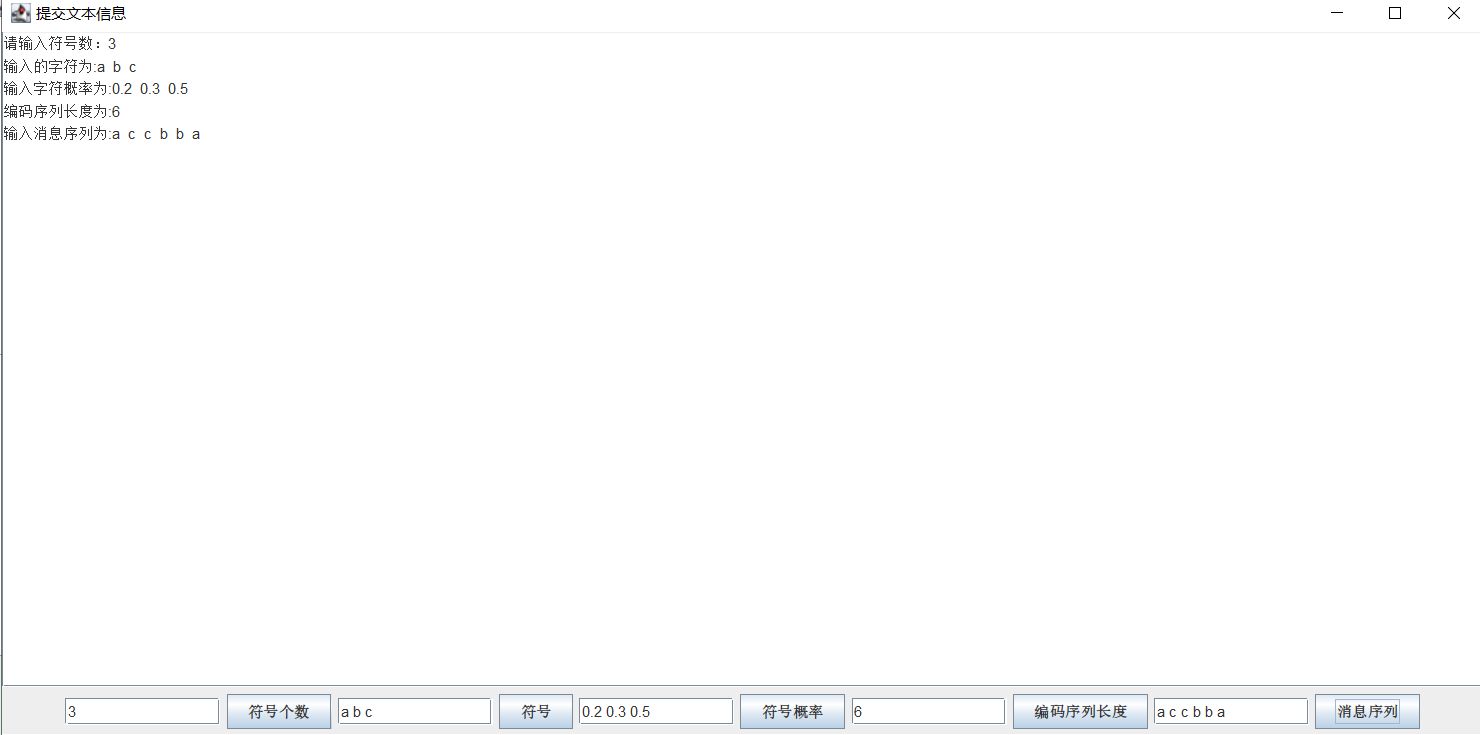
这两个编码的界面相同，同为打开当前界面，在文本框输入要编码的字符串，然后点击编码即可完成，结果展示在上边的文本域中。

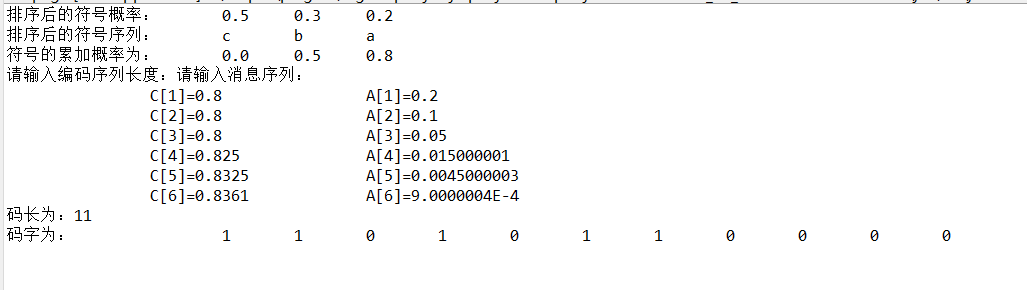




**5.算术编码的实现：**

打开算术编码的界面，然后在下方的五个文本框里分别输入编码的信息，点击相应按钮提交之后，可以看到上方的文本域中出现了输入的信息，依次提交完之后，最后一个按钮的功能包含对输入的信息进行编码，注意输入符号，符号概率，消息序列时要以空格隔开。

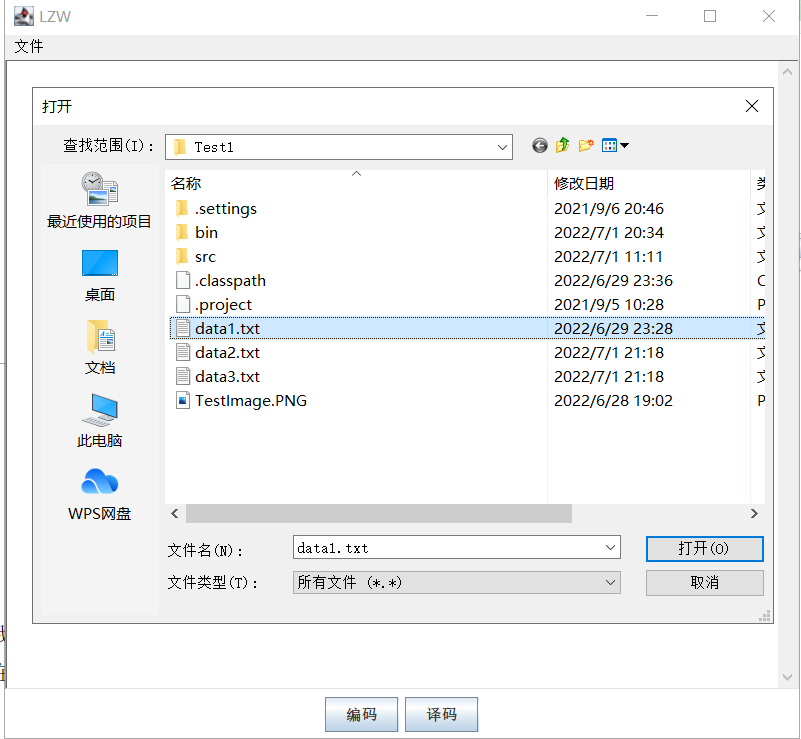


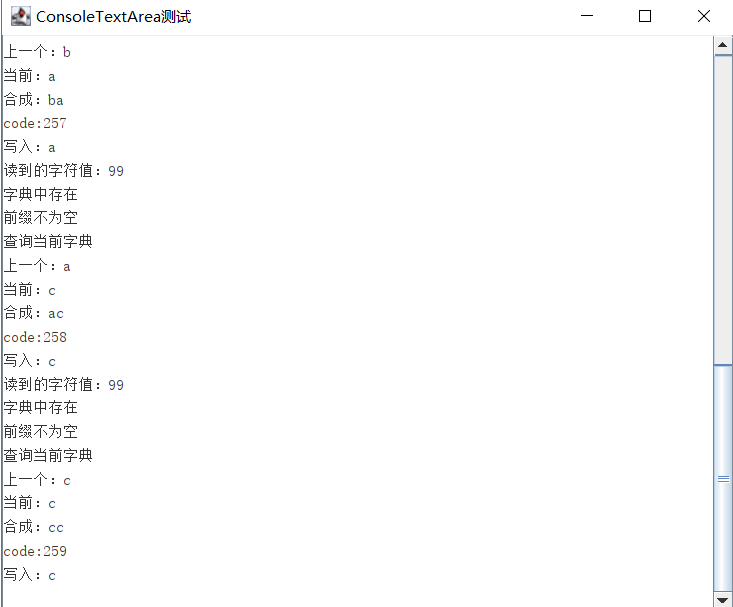


1. **LZW字典编码译码的实现**

打开LZW界面，点击上方的下拉列表，出现打开选项，点击进入文件系统，选择我们需要编码的文档，我们这里选择目录中的data1.txt文档，打开之后，结果就会展示在上方文本域中，然后点击下面的编码按钮，即可完成编码，编码过程展示在当前文本域中，然后结果保存在data2.txt文档中。

同样的，选择要译码的文档，如刚才编码的data2.txt文档，然后点击译码之后，弹出一个新界面展示编码过程。然后结果保存在data3.txt文档中，打开查看，发现与编码之前的data1.txt的文档相同，译码成功。

# 总结

## 本次程序设计的优点：

本次程序设计的优点有很多，使用java语言开发的程序，可以在各个操作系统上运行，java语言对于我们来说更加熟练且易于掌握，操作起来得心应手。

## 本次程序设计的缺点：

首先使用JAVA语言开发本次程序，GUI界面过于久远，并且在绘制曲线图等方面的能力很有限，以至于图像的精度无法体现，同时在实现算术编码的过程中，由于代码耦合性太高，导致无法将编码的过程全部展示在GUI界面上，只能在控制台输出结果，同时由于时间紧促，界面的优化工作做的不够完美。