

文件加解密软件

编号 design-se-2025

0.1 开发要求

综合应用 Java 面向对象、GUI、多线程等技术，集成本实验提供的加解密代码，实现一个图形化的文件加解密软件。具体要求如下：

- 1. 使用 Swing GUI 工具包实现图形界面开发，基本界面元素及功能要求如图 1 所示。

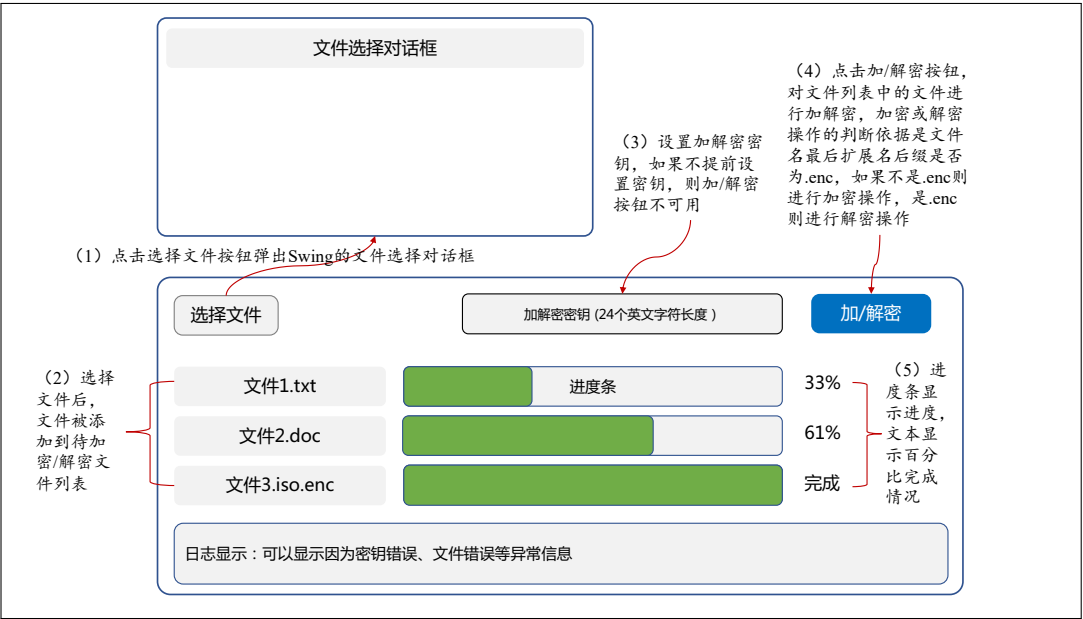


图 1: 界面及功能要求

- 2. 界面交互流程参考图 1，说明如下：

- (a) 点击选择文件按钮弹出 Swing 的文件选择对话框。
- (b) 选择文件后，文件被添加到待加密/解密文件列表。

- (c) 设置加解密密钥，如果不提前设置密钥，则加/解密按钮不可用。**注意：要求加解密密钥能够保存到配置文件中，无需每次都设置。**
 - (d) 点击加/解密按钮，对文件列表中的文件进行加解密，加密或解密操作的判断依据是文件名最后扩展名后缀是否为.enc，如果不是.enc 则进行加密操作，是.enc 则进行解密操作。
 - (e) 进度条显示进度，文本显示百分比完成情况。
 - (f) 日志显示：可以显示因为密钥错误、文件错误等异常信息。
3. 加解密算法采用 DES 对称密钥算法，以下提供算法工具类和文件加密测试类供参考，代码如下：

DESUtil.java

提供加密和解密两个静态方法，key 是加解密密钥。

```
1  import javax.crypto.Cipher;
2  import javax.crypto.SecretKey;
3  import javax.crypto.spec.SecretKeySpec;
4
5  public class DESUtil {
6      private static String ALGORITHM = "DESede";
7
8      /**
9       * 加密
10      *
11      * @param key
12      * @param src
13      * @return
14      */
15     public static byte[] enCode(byte[] key, byte[] src) {
16
17         byte[] value = null;
18         SecretKey deskey = new SecretKeySpec(key, ALGORITHM);
19         try {
20             Cipher cipher = Cipher.getInstance(ALGORITHM);
21             cipher.init(Cipher.ENCRYPT_MODE, deskey);
22             value = cipher.doFinal(src);
23         } catch (Exception e) {
24             e.printStackTrace();
25         }
26
27         return value;
28     }
29
30     /**
31      * 解密
32      *
33      * @param key
34      * @param src
```

```
35     * @return
36     */
37     public static byte[] deCode(byte[] key, byte[] src) {
38         byte[] value = null;
39         SecretKey deskey = new SecretKeySpec(key, ALGORITHM);
40
41         try {
42             Cipher cipher = Cipher.getInstance(ALGORITHM);
43             cipher.init(Cipher.DECRYPT_MODE, deskey);
44             value = cipher.doFinal(src);
45         } catch (Exception e) {
46             e.printStackTrace();
47         }
48
49         return value;
50     }
51 }
```

DESUtilTest.java

密钥 Key 在测试类中使用的是字符串形式，24 个字符。

```
1  import java.io.FileInputStream;
2  import java.io.FileOutputStream;
3
4  public class ClassFileEncoder {
5
6      public static void main(String[] args) {
7          String inFile = "/Users/kevin/README.doc";
8          FileInputStream fis = null;
9
10         try {
11             fis = new FileInputStream(inFile);
12             int len = fis.available();
13             byte[] data = new byte[len];
14             fis.read(data);
15             fis.close();
16
17             data = DESUtil.enCode("1234567890qwertyuiopasdf".getBytes(), data);
18
19             String outFile = "/Users/kevin/README.doc.enc";
20             FileOutputStream fos = new FileOutputStream(outFile);
21             fos.write(data);
22             fos.close();
23         } catch (Exception e) {
24             e.printStackTrace();
25         }
26     }
27
28 }
```

4. Swing 文件选择对话框参考代码如下：

```
1  import java.awt.event.ActionEvent;
2  import java.awt.event.ActionListener;
3  import java.io.File;
4
5  import javax.swing.JButton;
6  import javax.swing.JFileChooser;
7  import javax.swing.JFrame;
8  import javax.swing.JLabel;
9
10 public class FileChooser extends JFrame implements ActionListener {
11     JButton open = null;
12
13     public static void main(String[] args) {
14         new FileChooser();
15     }
16
17     public FileChooser() {
18         open = new JButton("选择文件");
19         this.add(open);
20         this.setBounds(400, 200, 100, 100);
21         this.setVisible(true);
22         this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
23         open.addActionListener(this);
24     }
25
26     @Override
27     public void actionPerformed(ActionEvent e) {
28         // TODO Auto-generated method stub
29         JFileChooser jfc = new JFileChooser();
30         jfc.setFileSelectionMode(JFileChooser.FILES_AND_DIRECTORIES);
31         jfc.showDialog(new JLabel(), "选择");
32         File file = jfc.getSelectedFile();
33         if (file.isDirectory()) {
34             System.out.println("文件绝对路径: " + file.getAbsolutePath());
35         } else if (file.isFile()) {
36             System.out.println("文件名: " + file.getAbsolutePath());
37         }
38         System.out.println(jfc.getSelectedFile().getName());
39     }
40 }
```

5. 关于进度条，可以使用 Swing 的进度条组件，也可以使用 panel 网格自行构造进度条组件。

6. 关于加密文件存储路径，可以将加密后的文件（以原文件名加.enc 后缀的命名）放在与原始文件一致的路径下。**注意：对加密后的文件解密产生的新文件，建议不要与原始文件名一**

致，以免出现任何异常导致原始文件被破坏。可以采用时间戳-原始文件名的形式存储解密文件。

7. 要求采用多线程技术，能够支持多个文件的同时加解密。加解密的过程同步更新进度条，进度百分比。
8. 关于日志输出，建议将软件中出现的异常处理代码产生的异常信息显示在日志组件中。日志显示可以采用 Swing 的 JTextField 组件。

0.2 评估说明

1. 要求每人独立完成，项目名称统一为 **file-sec-tool**，并录制简单的操作演示视频。
2. 开发完成情况会被评估，包括代码质量和由 jar 包反映出来的功能完成度。
 - 代码质量：包括代码结构规范性、命名规范性、注释规范性、能够体现一定的封装等。
 - 功能完成度：是否包含了要求的所有功能内容，界面是否美观，设计是否体现了一定的独立思考能力。
3. 本软件开发项目占期末成绩的 20%，4 实验课时。