Practice 8: KEY MANAGEMENT USING PUBLIC ENCRYPTION (Cont.)

8.1 OVERVIEW

8.1.1 Introduction

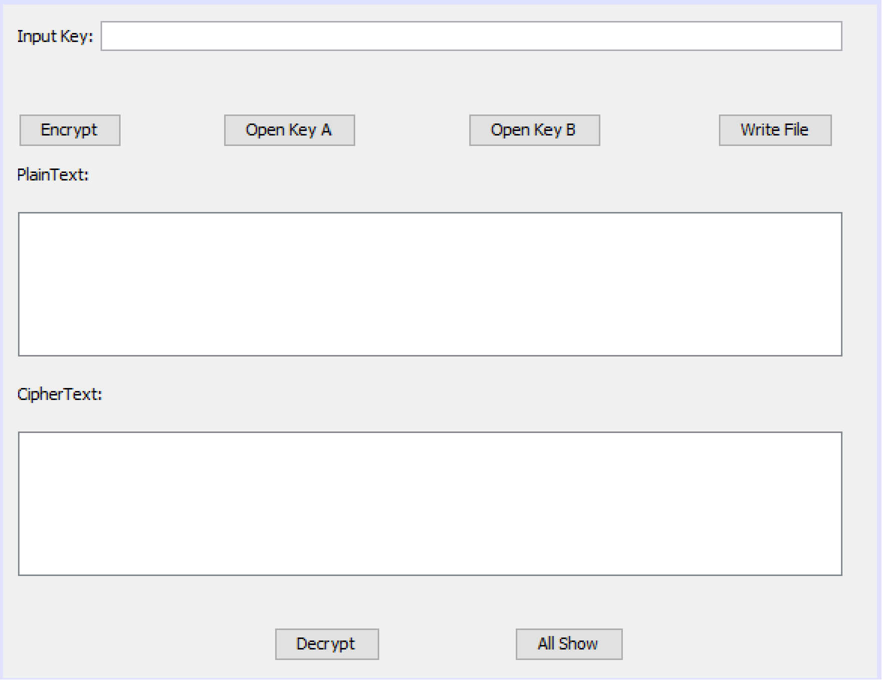
- Lab 8: Key Management using Public Encryption (Cont)
- Practice time: class: 3 study hours, self-study: 3 study hours.
- Requirements: Students using Netbeans Software

8.1.2 Objective

- This course provides students with knowledge of cryptographic algorithms and how they are used in today's world.
- The content emphasizes the principles, topics, approaches, and problem solving related to the underlying technologies and architectures of the field.

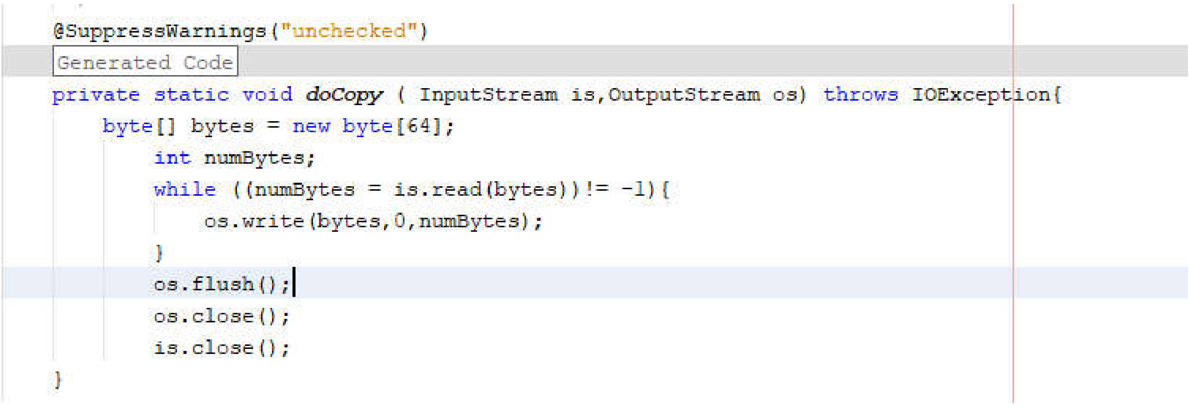
8.2 CONTENTS

❖ **Design Form:**

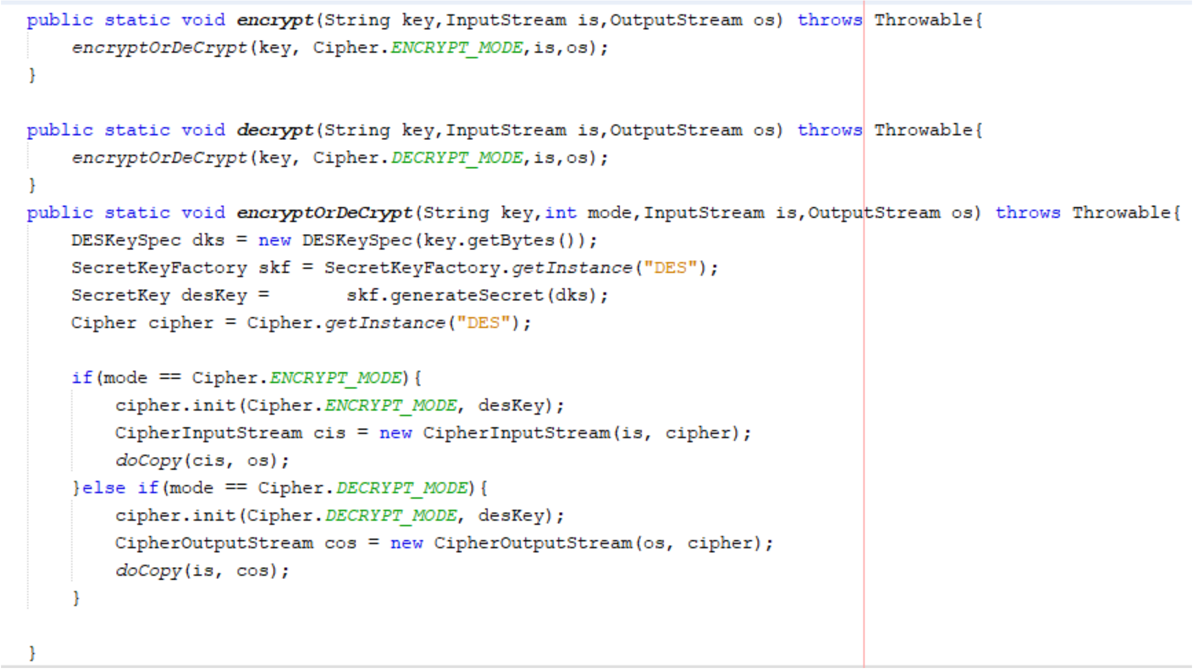


❖ Step 2: Write an event handler function:

2.1 Variable environment:

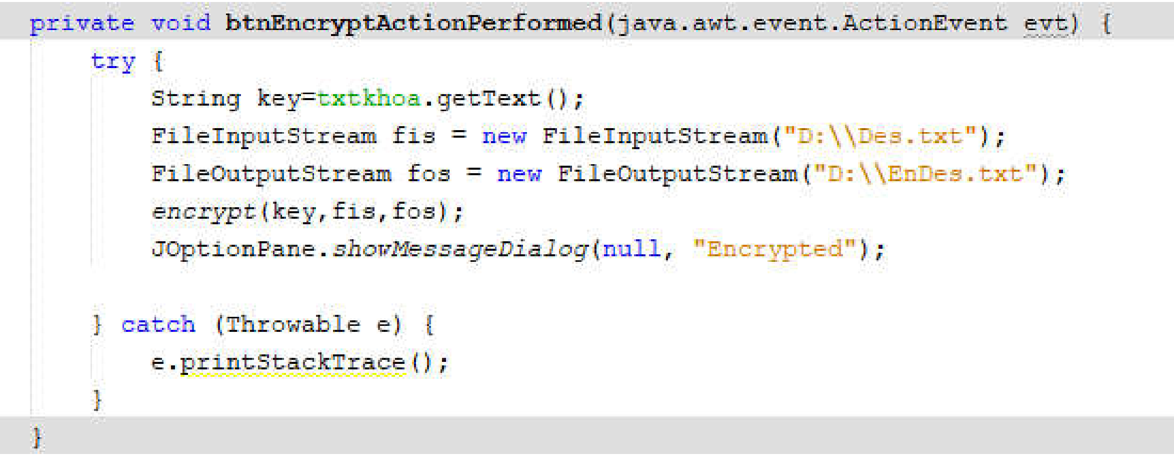


```
@SuppressWarnings("unchecked")
Generated Code
private static void doCopy ( InputStream is,OutputStream os) throws IOException{
    byte[] bytes = new byte[64];
        int numBytes;
        while ((numBytes = is.read(bytes))!= -1){
            os.write(bytes,0,numBytes);
        }
        os.flush();
        os.close();
        is.close();
}
```



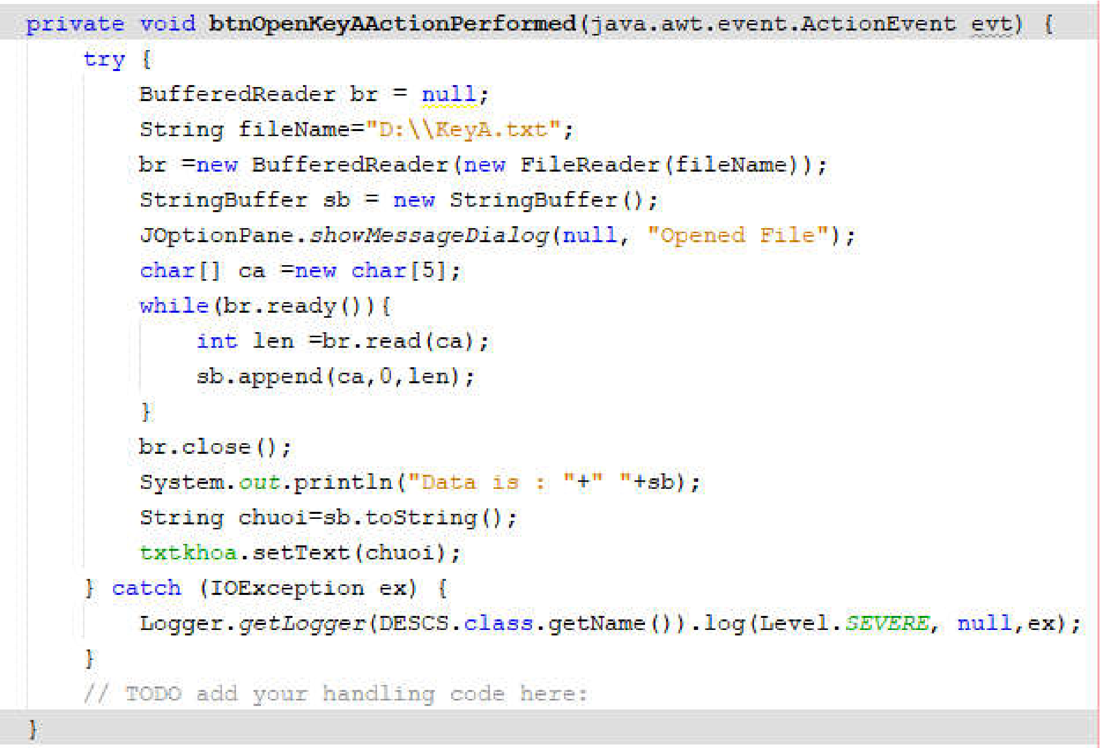
```
public static void encrypt(String key,InputStream is,OutputStream os) throws Throwable{
    encryptOrDeCrypt(key, Cipher.ENCRYPT_MODE,is,os);
}

public static void decrypt(String key,InputStream is,OutputStream os) throws Throwable{
    encryptOrDeCrypt(key, Cipher.DECRYPT_MODE,is,os);
}
public static void encryptOrDeCrypt(String key,int mode,InputStream is,OutputStream os) throws Throwable{
    DESKeySpec dks = new DESKeySpec(key.getBytes());
    SecretKeyFactory skf = SecretKeyFactory.getInstance("DES");
    SecretKey desKey =      skf.generateSecret(dks);
    Cipher cipher = Cipher.getInstance("DES");

    if(mode == Cipher.ENCRYPT_MODE){
        cipher.init(Cipher.ENCRYPT_MODE, desKey);
        CipherInputStream cis = new CipherInputStream(is, cipher);
        doCopy(cis, os);
    }else if(mode == Cipher.DECRYPT_MODE){
        cipher.init(Cipher.DECRYPT_MODE, desKey);
        CipherOutputStream cos = new CipherOutputStream(os, cipher);
        doCopy(is, cos);
    }

}
```

2.2 Button Encrypt:



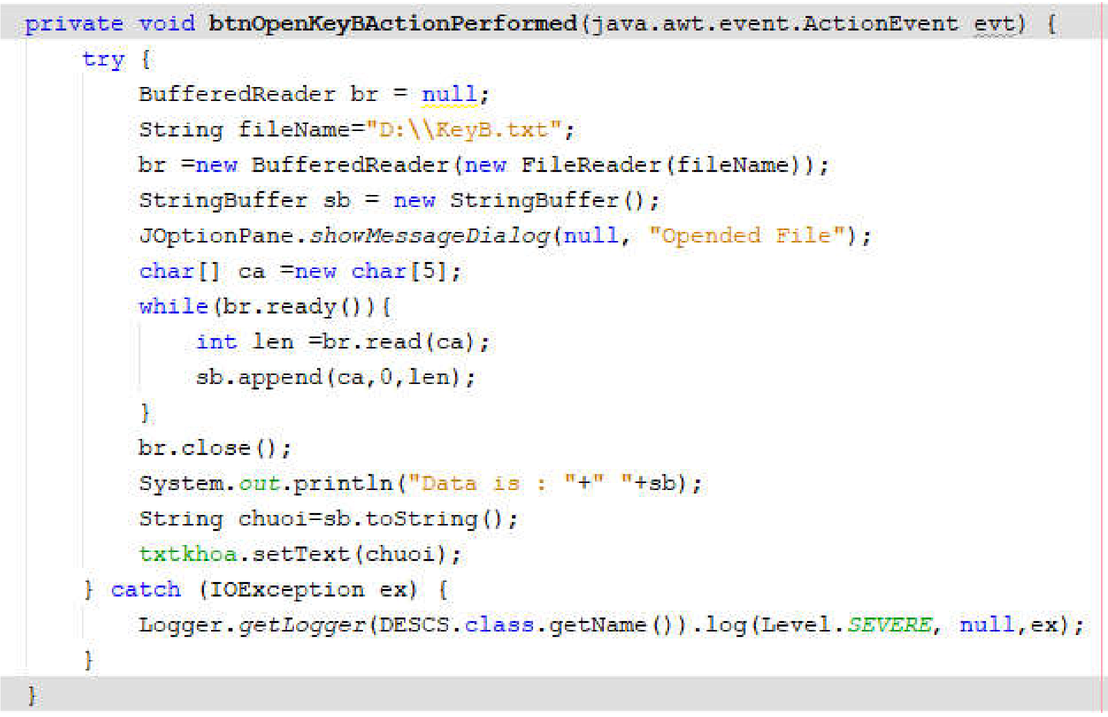
```
private void btnEncryptActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        String key=txtkhoa.getText();
        FileInputStream fis = new FileInputStream("D:\\Des.txt");
        FileOutputStream fos = new FileOutputStream("D:\\EnDes.txt");
        encrypt(key,fis,fos);
        JOptionPane.showMessageDialog(null, "Encrypted");

    } catch (Throwable e) {
        e.printStackTrace();
    }
}
```

2.3 Button Open Key A:



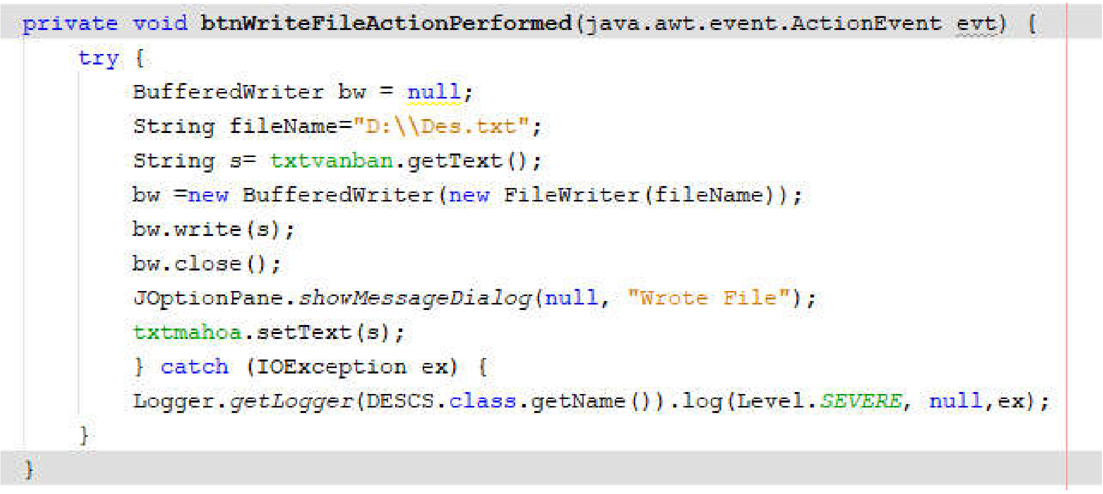
```
private void btnOpenKeyAActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        BufferedReader br = null;
        String fileName="D:\\KeyA.txt";
        br =new BufferedReader(new FileReader(fileName));
        StringBuffer sb = new StringBuffer();
        JOptionPane.showMessageDialog(null, "Opened File");
        char[] ca =new char[5];
        while(br.ready()){
            int len =br.read(ca);
            sb.append(ca,0,len);
        }
        br.close();
        System.out.println("Data is : "+" "+sb);
        String chuoi=sb.toString();
        txtkhoa.setText(chuoi);
    } catch (IOException ex) {
        Logger.getLogger(DESCS.class.getName()).log(Level.SEVERE, null,ex);
    }
    // TODO add your handling code here:
}
```

2.4 Button Open Key B:



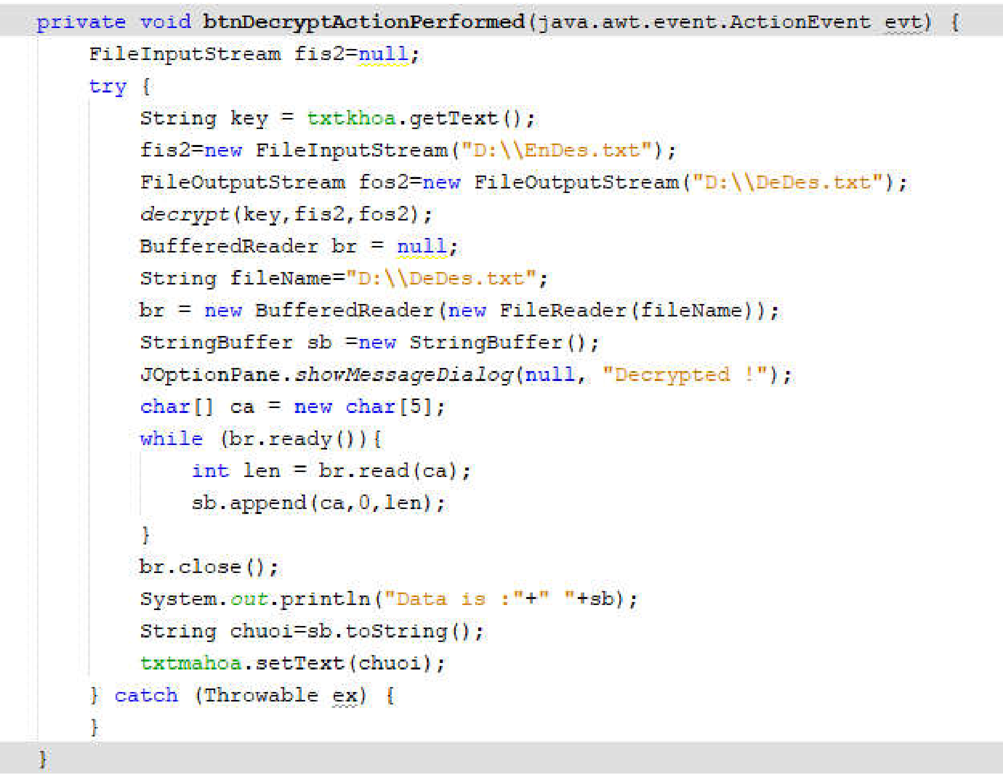
```
private void btnOpenKeyBActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        BufferedReader br = null;
        String fileName="D:\\KeyB.txt";
        br =new BufferedReader(new FileReader(fileName));
        StringBuffer sb = new StringBuffer();
        JOptionPane.showMessageDialog(null, "Opended File");
        char[] ca =new char[5];
        while(br.ready()){
            int len =br.read(ca);
            sb.append(ca,0,len);
        }
        br.close();
        System.out.println("Data is : "+" "+sb);
        String chuoi=sb.toString();
        txtkhoa.setText(chuoi);
    } catch (IOException ex) {
        Logger.getLogger(DESCS.class.getName()).log(Level.SEVERE, null,ex);
    }
}
```

2.5 Button Write Fille:



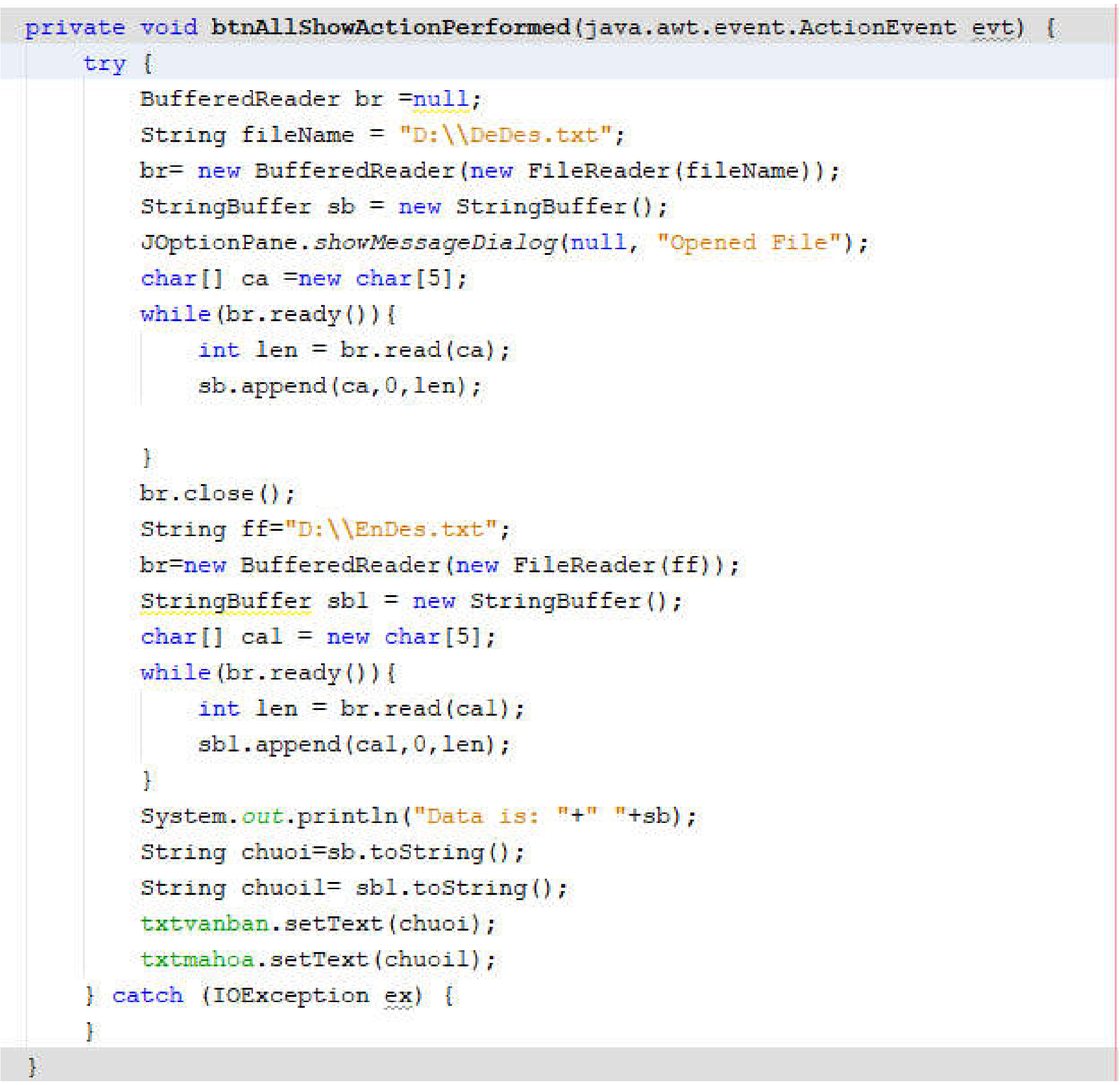
```
private void btnWriteFileActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        BufferedWriter bw = null;
        String fileName="D:\\Des.txt";
        String s= txtvanban.getText();
        bw =new BufferedWriter(new FileWriter(fileName));
        bw.write(s);
        bw.close();
        JOptionPane.showMessageDialog(null, "Wrote File");
        txtmahoa.setText(s);
        } catch (IOException ex) {
        Logger.getLogger(DESCS.class.getName()).log(Level.SEVERE, null,ex);
    }
}
```

2.6 Button Decrypt:



```
private void btnDecryptActionPerformed(java.awt.event.ActionEvent evt) {
    FileInputStream fis2=null;
    try {
        String key = txtkhoa.getText();
        fis2=new FileInputStream("D:\\EnDes.txt");
        FileOutputStream fos2=new FileOutputStream("D:\\DeDes.txt");
        decrypt(key,fis2,fos2);
        BufferedReader br = null;
        String fileName="D:\\DeDes.txt";
        br = new BufferedReader(new FileReader(fileName));
        StringBuffer sb =new StringBuffer();
        JOptionPane.showMessageDialog(null, "Decrypted !");
        char[] ca = new char[5];
        while (br.ready()){
            int len = br.read(ca);
            sb.append(ca,0,len);
        }
        br.close();
        System.out.println("Data is :"+" "+sb);
        String chuoi=sb.toString();
        txtmahoa.setText(chuoi);
    } catch (Throwable ex) {
    }
}
```

2.7 Button Decrypt:



```
private void btnAllShowActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        BufferedReader br =null;
        String fileName = "D:\\DeDes.txt";
        br= new BufferedReader(new FileReader(fileName));
        StringBuffer sb = new StringBuffer();
        JOptionPane.showMessageDialog(null, "Opened File");
        char[] ca =new char[5];
        while(br.ready()){
            int len = br.read(ca);
            sb.append(ca,0,len);

        }
        br.close();
        String ff="D:\\EnDes.txt";
        br=new BufferedReader(new FileReader(ff));
        StringBuffer sb1 = new StringBuffer();
        char[] ca1 = new char[5];
        while(br.ready()){
            int len = br.read(ca1);
            sb1.append(ca1,0,len);
        }
        System.out.println("Data is: "+" "+sb);
        String chuoi=sb.toString();
        String chuoi1= sb1.toString();
        txtvanban.setText(chuoi);
        txtmahoa.setText(chuoi1);
    } catch (IOException ex) {
    }
}
```

Result: