

Vigenere

```

1 //import.....
2 import java.io.BufferedReader;
3 import java.io.BufferedWriter;
4 import java.io.FileReader;
5 import java.io.FileWriter;
6 import java.io.IOException;
7 import java.util.logging.Level;
8 import java.util.logging.Logger;
9 import javax.swing.JOptionPane;
181 //Encrypt.....
182 private void QENActionPerformed(java.awt.event.ActionEvent evt) {
183     int tableRowSize = 26;
184     int tableColumnSize = 26;
185     int vignereTable[][] = new int[26][26];
186
187     for (int rows = 0 ; rows < tableRowSize ; rows++) {
188         int tableColumnSide;
189
190         for (int columns = 0; columns < tableColumnSize; columns++) {
191             vignereTable[rows][columns] = (rows + columns) % 26;
192         }
193     }
194     System.out.println("enter to plaintxt");
195     String plainText = Qtxtvanban.getText(); //Qtxtvanban: là văn bản plaintext
196
197     plainText = plainText.toUpperCase();
198
199     System.out.print("enter the key");
200
201     String key = Qtxtkhoea.getText(); //Qtxtkhoea: là văn bản khóa
202     key = key.toUpperCase();
203     String cipherText = "";
204     int keyIndex = 0;
205     for (int ptextIndex = 0; ptextIndex < plainText.length(); ptextIndex++)
206     {
207
208
209         char pChar = plainText.charAt(ptextIndex);
210         int asciiVal = (int) pChar;
211         if (pChar == ' '){
212             cipherText += pChar; continue;
213         }
214         if (asciiVal < 65 || asciiVal > 90 ) { cipherText += pChar; continue;
215     }
216     int basicPlainTextValue = ((int) pChar) - 65;
217     char kChar = key.charAt(keyIndex);
218     int basicKeyValue = ((int) kChar) - 65;
219     int tableEntry = vignereTable[basicPlainTextValue][basicKeyValue];

```

```

220
221     char cChar=(char) (tableEntry +65 );
222     cipherText += cChar;
223     keyIndex++;
224     if (keyIndex ==key.length())
225         keyIndex = 0;
226     System.out.println("ciphertext is "+cipherText);
227     Qtxtmahoa.setText(cipherText.toUpperCase()); //Qtxtmahoa: là văn bản ciphertext
228 }
229 }
230
231
232
233 @SuppressWarnings("empty-statement")
234 //Dencrypt.....
235 private void QDEActionPerformed(java.awt.event.ActionEvent evt) {
236     // TODO add your handling code here:
237     int tableRowSize = 26;
238     int tableColumnSize = 26;
239     int vignereTable[][] = new int[26][26];
240     String cipherText=Qtxtmahoa.getText(); //Qtxtmahoa: là văn bản ciphertext
241     String plainText="";
242     for (int rows =0; rows < tableRowSize; rows++){
243         for(int columns = 0; columns <tableColumnSize; columns++){
244             vignereTable[rows][columns] = (rows + columns) %26;
245         }
246     }
247     System.out.println("Enter the cipher text");
248     cipherText = cipherText.toUpperCase();
249     System.out.print("Enter the key: ");
250     String key = Qtxtkhoea.getText(); //Qtxtkhoea: là văn bản khóa
251     key = key.toUpperCase();
252     int keyIndex = 0;
253     for (int ctextIndex = 0; ctextIndex < cipherText.length(); ctextIndex++){
254         char cChar = cipherText.charAt(ctextIndex);
255         int asciiVal = (int) cChar;
256         if(cChar == ' '){
257             plainText +=cChar;
258             continue;
259         }
260         if (asciiVal < 65 || asciiVal > 90){
261             plainText +=cChar;
262             continue;
263         }
264         int basiccipherTextValue = ((int) cChar) -65;
265         char kChar = key.charAt(keyIndex);
266         int basicKeyValue = ((int) kChar) - 65;
267         for (int pIndex =0; pIndex < tableColumnSize; pIndex++){
268             if(vignereTable[basicKeyValue][pIndex]== basiccipherTextValue){
269                 char potChar = (char) (vignereTable[basiccipherTextValue][pIndex]+ 65);
270                 char potpChar = (char) (pIndex + 65);
271                 plainText += potpChar;
272             }
273         }

```

```

274     }
275     keyIndex++;
276     if (keyIndex == key.length())
277         keyIndex = 0;
278 }
279 System.out.println("plaintext is" +plainText);
280 Qtxtvanban.setText(cipherText.toString().toUpperCase());//Qtxtvanban: là văn bản plaintext
281 Qtxtmahoa.setText((plainText.toString().toUpperCase()));//Qtxtmahoa: là văn bản ciphertext
282
283 }
284
285 private void QtxtkhoaActionPerformed(java.awt.event.ActionEvent evt) {
286     // TODO add your handling code here:
287 }
288 //Mo File.....
289 private void QMFActionPerformed(java.awt.event.ActionEvent evt) {
290     // TODO add your handling code here:
291     try {
292         String txtDuongdan = "D:\\NQLQuang\\text\\dulieu.txt";
293         BufferedReader br = null;
294         br = new BufferedReader(new FileReader(txtDuongdan));
295         StringBuffer sb = new StringBuffer();
296         JOptionPane.showMessageDialog(null,"Mở File thành công");
297         char[] ca=new char[5];
298         while (br.ready()){
299             int len=br.read(ca);
300             sb.append(ca,0,len);
301         }
302         br.close();
303         String chuoi = sb.toString();
304         Qtxtvanban.setText(chuoi);//Qtxtvanban: là văn bản plaintext
305     } catch (IOException ex) {
306         Logger.getLogger(Quang.class.getName()).log(Level.SEVERE,null,ex);
307     }
308 }
309 //Ghi File.....
310 private void QGFActionPerformed(java.awt.event.ActionEvent evt) {
311     // TODO add your handling code here:
312     try {
313         String txtLuu = Qtxtvanban.getText();//Qtxtvanban: là văn bản plaintext
314         BufferedWriter bw = null;
315         String txtDuongdan = "D:\\NQLQuang\\text\\dulieu.txt";
316         bw = new BufferedWriter(new FileWriter(txtDuongdan));
317         bw.write(txtLuu);
318         bw.close();
319         JOptionPane.showMessageDialog(null,"Ghi File thanh cong");
320     } catch (IOException ex) {
321         Logger.getLogger(Quang.class.getName()).log(Level.SEVERE,null, ex);
322     }
323 }

```