



TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHỆ

KHOA CÔNG NGHỆ THÔNG TIN

Môn: Bảo Mật Thông Tin

Bài thực hành số 1



Bài 1: Viết chương trình mã hóa và giải mã văn bản với thuật toán mã hóa Ceasar.

Chương trình có thể thực hiện các chức năng sau:

Cho phép nhập văn bản vào hệ thống.

Cho phép nhập khóa bảo vệ văn bản.

Cho phép ghi File và mở File.

Hướng dẫn mã dịch chuyển Caesar:

-Ta lần lượt đánh chỉ số cho các chữ cái bắt đầu từ 0.

- Gọi k là 1 số nguyên từ 0 ->25 được gọi là khóa.

-Hàm mã hóa: $E(p,k)=(p+k)\bmod 26$ với p là chỉ số của ký tự cần mã hóa.

-Hàm giải mã: $D(c,k)=|c-k|\bmod 26$ với c là chỉ số của ký tự cần giải mã.

Encryption:

1. Map the plaintext characters to numbers : $a = 0, \dots, z = 25$

2. Encrypt the message (sequence of numbers m) using

$$c = a*m + b \bmod 26$$

where a and b are the encryption keys.

3. Map the numbers back to characters to obtain the cipher

Decryption:

1. Map the cipher characters to numbers : $a = 0, \dots, z = 25$

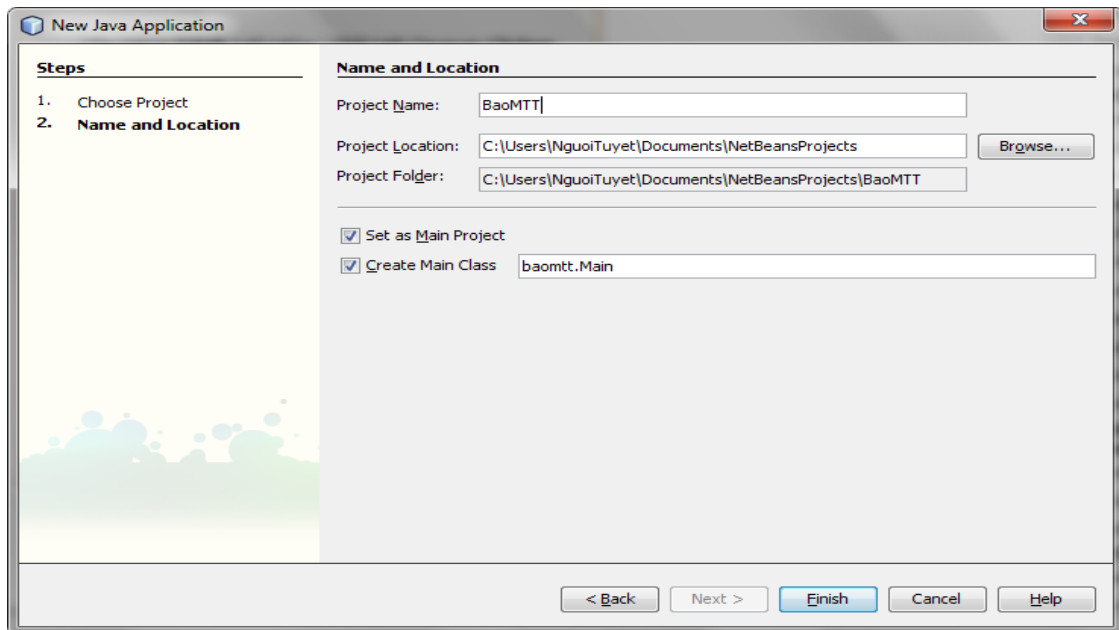
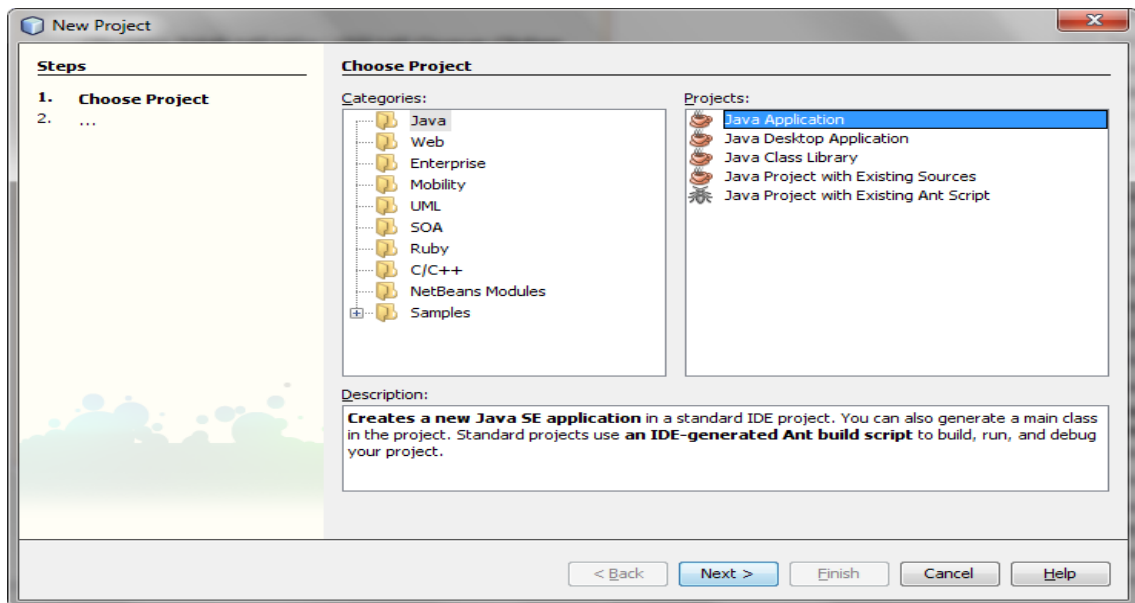
2. Decrypt the number sequence c using

$$m = a^{-1}*c + a^{-1}*b \bmod 26$$

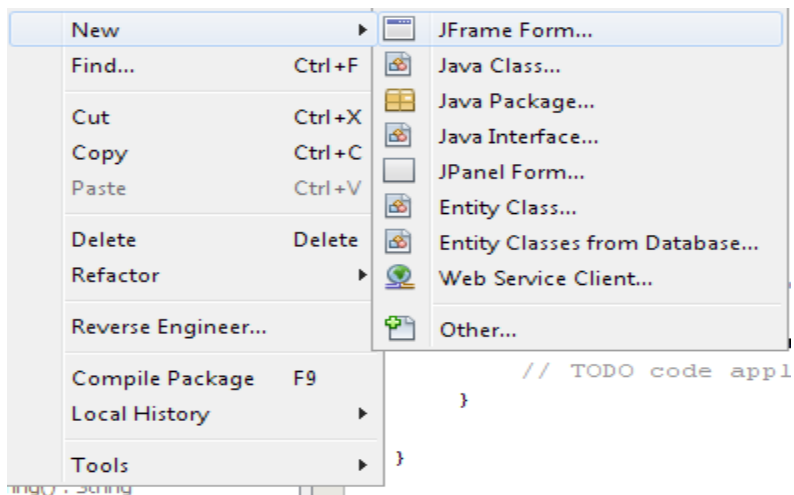
where a^{-1} is the **multiplicative inverse** of a mod 26

3. Map the numbers back to characters

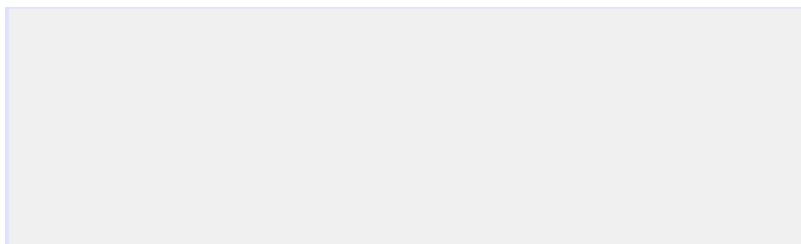
Bước 1: Tạo project mới: File → New Project



Bước 2: Tạo mới JFrame Form thiết kế:



Giao diện thiết kế Frame:



Bước 3: Thiết kế Form

The image shows a screenshot of a Java Swing window titled "Chương Trình Mã Hóa/ Giải Mã Ceasar Cipher". The window has a light gray background and a thin orange border. It contains the following elements:

- Title Bar:** "Chương Trình Mã Hóa/ Giải Mã Ceasar Cipher"
- PlainText:** A text area with a vertical scrollbar on the right.
- Khóa :** A text field.
- Buttons:**
 - "v Encrypt v": A button with a green 'v' on the left and a green 'v' on the right.
 - "Ghi File": A button.
- Cipher Text :** A text area with a vertical scrollbar on the right.
- Buttons:**
 - "^ Decrypt ^": A button with a green '^' on the left and a green '^' on the right.
 - "Mở File": A button.

Bước 4: Viết hàm xử lý sự kiện

a. Hàm xử lý sự kiện Encrypt

```
private void btnmahoa(java.awt.event.ActionEvent evt) {  
    int k=Integer.valueOf(this.txtkhoa.getText());  
    String br=this.txtvanban.getText();  
    this.txtmahoa.setText(mahoa(br,k));  
}
```

b. Hàm xử lý sự kiện Ghi File

```
private void bntGhiFileActionPerformed(java.awt.event.ActionEvent evt) {  
    try {  
        // TODO add your handling code here:  
        BufferedWriter bw = null;  
        //Nơi lưu dữ liệu  
        String fileName = "D:\\\\Dulieu.txt";  
        //luu van ban  
        String s = txtmahoa.getText();  
        // Ghi dữ Liệu S vào tạo tin fileName  
        bw = new BufferedWriter(new FileWriter(fileName));  
        bw.write(s);  
        bw.close();  
        JOptionPane.showMessageDialog(null, " Đã Ghi File Thành Công !!!");  
    } catch (IOException ex) {  
        Logger.getLogger(Cesar.class.getName()).log(Level.SEVERE, null, ex);  
    }  
}
```

c. Hàm xử lý sự kiện Dencrypt

```
private void btngiaima(java.awt.event.ActionEvent evt) {  
    int k=Integer.valueOf(this.txtkhoa.getText());  
    String br=this.txtmahoa.getText();  
    this.txtvanban.setText(mahoa(br,-k));  
}  
  
char mahoakt(char c,int k)  
{  
    if(!Character.isLetter(c)) return c;  
    return (char) (((Character.toUpperCase(c) - 'A') + k) % 26 + 26) % 26 + 'A';  
}  
  
private String mahoa(String br, int k) {  
    String kq="";  
    int n=br.length();  
    for(int i=0;i<n;i++)  
        kq+=mahoakt(br.charAt(i),k);  
    return kq;  
}
```

d. Hàm xử lý sự kiện Mở File

```
private void bntMoFileActionPerformed(java.awt.event.ActionEvent evt) {  
    try {  
        // TODO add your handling code here:  
        // Lớp BufferedReader kế thừa từ lớp Reader thuộc nhóm Input  
        //Nhóm này là 1 trong 2 loại chính của Các luồng Ký Tự  
        // Có chức năng đọc dữ liệu dạng ký tự  
        BufferedReader br = null;  
        String fileName = "D:\\Dulieu.txt";  
        br = new BufferedReader(new FileReader(fileName));  
        //nhận dữ liệu  
        StringBuffer sb = new StringBuffer();  
        JOptionPane.showMessageDialog(null, " Đã mở File Thành Công !!!");  
        //Đọc mỗi lần tối đa 5 ký tự  
        char[] ca = new char[5];  
        while (br.ready()) {  
            int len = br.read(ca);  
            sb.append(ca, 0, len);  
        }  
        br.close();  
  
        //xuất chuỗi  
        System.out.println("Du Lieu la : " + " " + sb);  
        String chuoi = sb.toString();  
        // Hiển thị lên Form  
        txtvanban.setText(chuoi);  
    } catch (IOException ex) {  
        Logger.getLogger(Cesar.class.getName()).log(Level.SEVERE, null, ex);  
    }  
}
```

Bài 2: Viết chương trình mã hóa và giải mã văn bản với thuật toán mã hóa Vigenere.

Chương trình có thể thực hiện các chức năng sau:

Cho phép nhập văn bản vào hệ thống.

Cho phép nhập khóa bảo vệ văn bản.

Cho phép mở File và Ghi File.

Hướng Dẫn: Mật mã Vigenere còn gọi là mật mã nhiều bảng mã. Ưu điểm của mã này là việc sử dụng 26 bảng mã khác nhau. Do đó mà không bị phá trong một thời gian dài. Ngoài ra mã này còn hỗ trợ việc sử dụng từ khóa vô cùng tiện lợi.

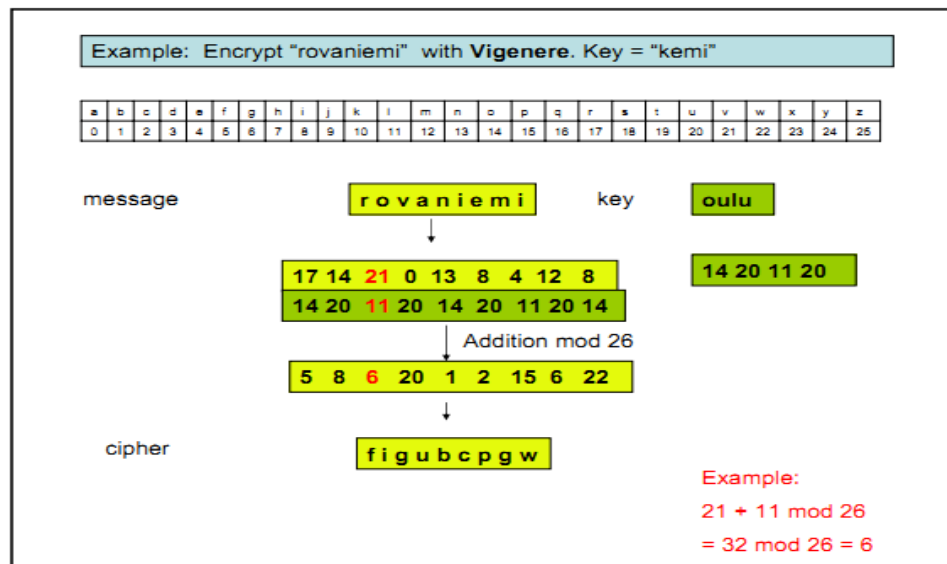
Thuật toán:

- Khoá K là một bộ gồm nhiều khoá con
 - $K = (k_1, k_2, \dots, k_m)$
- **Mã hoá:**
 - $e_K(x_1, x_2, \dots, x_m) = (x_1 + k_1, x_2 + k_2, \dots, x_m + k_m)$
- **Giải mã:**
 - $d_K(y_1, y_2, \dots, y_m) = (y_1 - k_1, y_2 - k_2, \dots, y_m - k_m)$
 - (cộng, trừ theo modulo 26)

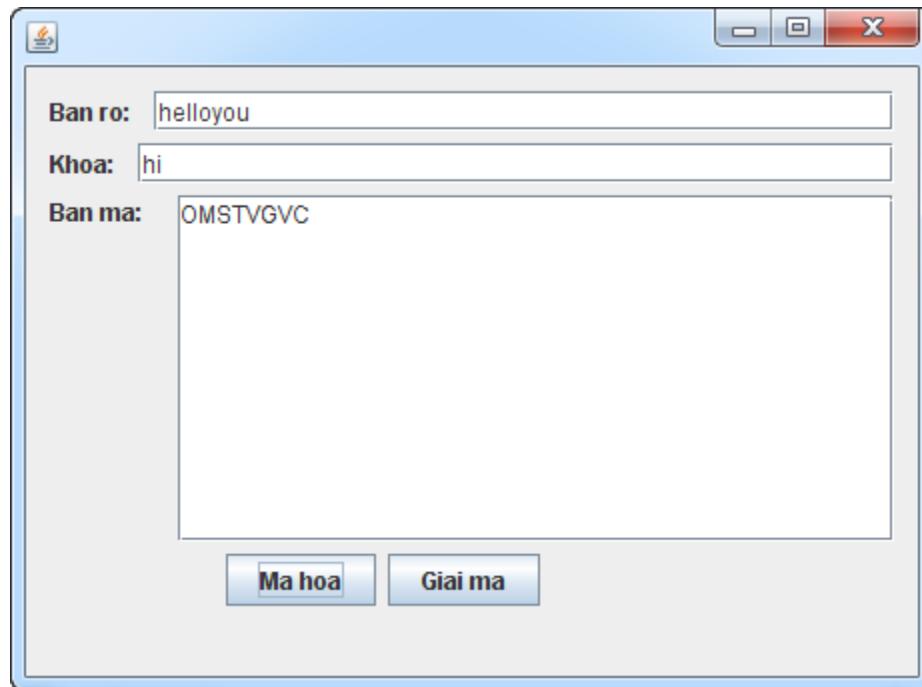
→ “MÃ KHỐI” (block cipher)

Ví dụ:

- $\{A, B, C, \dots, X, Y, Z\} = \mathbb{Z}_{26} = \{0, 1, \dots, 25\}$.
- $K = (2, 8, 15, 7, 4, 17)$ (“CIPHER”).
- $p = \text{“thiscryptosy”}$.
- $c = \text{“VPXZGIA\underline{X}IVWP\underline{P}”}$



Bước 1: Thiết Kế Form :



Hàm khởi tạo form:

```
public class Form2 extends javax.swing.JFrame {  
    int vig[] [];  
    /** Creates new form Form2 */  
    public Form2() {  
        initComponents();  
        vig=new int[26][26];  
        for(int i=0;i<26;i++)  
            for(int j=0;j<26;j++)  
                vig[i][j]=(i+j)%26;  
    }  
}
```

Bước 2: Viết hàm xử lý sự kiện

a. Hàm xử lý sự kiện Encrypt

```
private void btnMaHoa(java.awt.event.ActionEvent evt) {  
    String banro=this.txtbr.getText();  
    String k=this.txtk.getText();  
    String banma=mahoa(banro,k);  
    this.txtbm.setText(banma);  
}
```

```

private String mahoat(String banro, String key) {
    int n=banro.length();
    String banma="";
    int k=0;
    for(int i=0;i<n;i++)
        if(Character.isLetter(banro.charAt(i)))
        {
            banma+=mahoat(banro.charAt(i),key.charAt(k));
            k++;
            k=k%key.length();
        }else
            banma+=banro.charAt(i);
    return banma;
}

char mahoat(char x,char k)
{
    int xn=Character.toUpperCase(x)-'A';
    int kn=Character.toUpperCase(k)-'A';
    int yn=vig[kn][xn];
    return (char) (yn+'A');
}

```

b. Hàm xử lý sự kiện Dencrypt

```

private void btnGiaiMa(java.awt.event.ActionEvent evt) {
    String banma=this.txtbm.getText();
    String k=this.txtk.getText();
    String kt1="";
    int kn=k.length();
    for(int i=0;i<kn;i++)
        kt1+=(char) (((26-(Character.toUpperCase(k.charAt(i))- 'A')) % 26) + 'A');
    this.txtk.setText(kt1);
    String banro=mahoat(banma,kt1);
    this.txtbr.setText(banro);
}

```

Bài Tập Về Nhà: Yêu cầu viết phần mềm mã hóa và giải mã với 2 thuật toán trên bao gồm:

- Menu mã hóa: Thuật toán Ceasar, Thuật Toán Vigenere
- Menu giải mã: Thuật toán Ceasar, Thuật Toán Vigenere
- Các chức năng mã hóa và giải mã đều phải có mục mã hóa và giải mã theo File(File có thể là .txt, .dat,...)

Bài 3: Viết chương trình mã hóa và giải mã văn bản với thuật toán mã hóa Rail Fence.

Chương trình có thể thực hiện các chức năng sau:

Cho phép nhập văn bản vào hệ thống.

Cho phép nhập khóa bảo vệ văn bản.

Cho phép mở File và Ghi File.

Hướng Dẫn : Mã Rail Fence còn được gọi là mã zig zag là một hình thức của mã chuyển vị:

- Thông điệp được viết lần lượt từ trái qua phải trên các cột (rail) của một hàng dào tưởng tượng theo đường chéo từ trên xuống dưới.
- Theo đường chéo từ dưới lên khi đạt tới cột thấp nhất.
- Và khi đạt tới cột cao nhất, lại viết theo đường chéo từ trên xuống. Cứ lặp đi lặp lại như thế nào cho đến khi viết hết toàn bộ nội dung của thông điệp.
- Ví dụ: mã hóa chuỗi HUTECH TECHNOLOGY với khóa là 2.

Plaintext:	<u>HUTECHTECHNOLOGY</u>
RailFence:	<div>H T C T C N L G U E H E H O O Y</div>
Ciphertext:	<u>HTCTCNLGUEHEHO OY</u>

Bước 1: Thiết Kế Form :

Thuật Toán Mã Hóa Rail Fence Cipher

Plaintext:

Key:

Ciphertext:

Bước 2: Viết hàm xử lý sự kiện

a. Hàm xử lý sự kiện Encrypt

```
private void mahoabtn(java.awt.event.ActionEvent evt) {
    int k=Integer.valueOf(this.txtkhoa.getText());
    String s=this.txtbri.getText();
    int n=s.length();
    int sd,sc;
    sd=k;
    sc=n/sd+1;
    char hr[][]=new char[sd][sc];
    int c,d;
    c=0;d=0;
    int sodu=n%sd;
    for(int i=0;i<n;i++)
    {
```

```

        hr[d][c]=s.charAt(i);
        d++;
        if(d==k)
        {
            c++; d=0;
        }
    }
    String kq="";
    int sokytu=sc;
    for(int i=0;i<sd;i++)
    {
        if(i>=sodu) sokytu=sc-1;
        for(int j=0;j<sokytu;j++)
            kq=kq+hr[i][j];

    }
    this.txtbm.setText(kq);
}

```

b. Hàm xử lý sự kiện Dencrypt: (Bài tập về nhà Sinh viên)

```

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    int k=Integer.valueOf(this.txtkhoa.getText());
    String s=this.txtbm.getText();
    int n=s.length();
    int sd,sc;
    sd=k;
    sc=n/sd+1;
    char hr[][]=new char[sd][sc];
    for(int i=0;i<sd;i++)
    {
        if(i>=sodu) sokytu=sc-1;
        for(int j=0;j<sokytu;j++)
        {
            hr[i][j]=s.charAt(t);
            t++;
        }
    }
    int c,d;
    c=0;d=0;
}

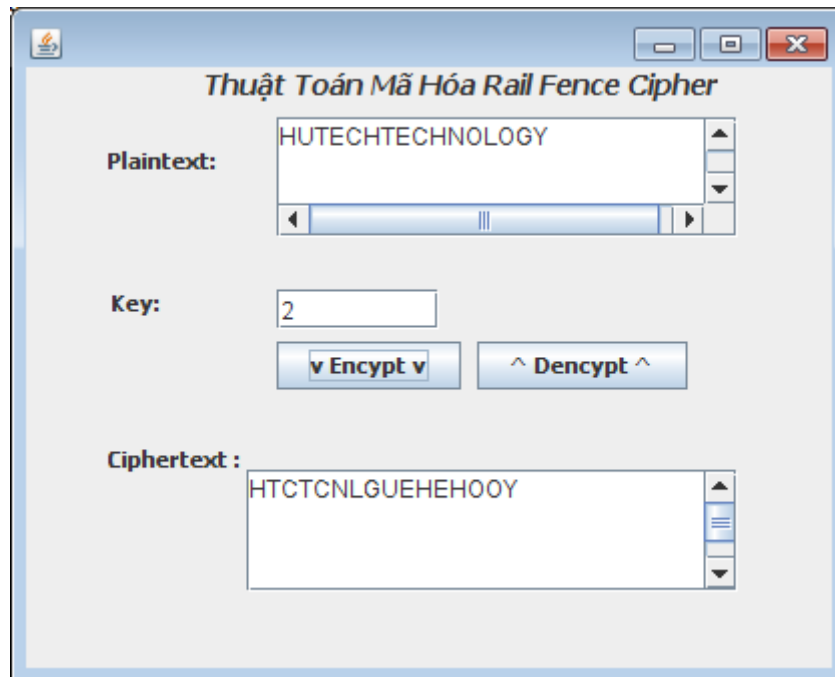
```

```

for(int i=0;i<n;i++)
{
    kq+=hr[d][c];
    d++;
    if(d==k)
    {
        c++; d=0;
    }
}
this.txtbr.setText(kq);
}

```

Bước 3: Kiểm Tra



Bài 4: Viết chương trình mã hóa và giải mã văn bản với thuật toán mã hóa PLayer.


Chương trình có thể thực hiện các chức năng sau:

Cho phép nhập văn bản vào hệ thống.

Cho phép nhập khóa bảo vệ văn bản.

Cho phép mở File và Ghi File.

Hướng dẫn:

 **Phương pháp** là lập ma trận 5x5 dựa trên từ khóa cho trước và các ký tự trên bảng chữ cái :

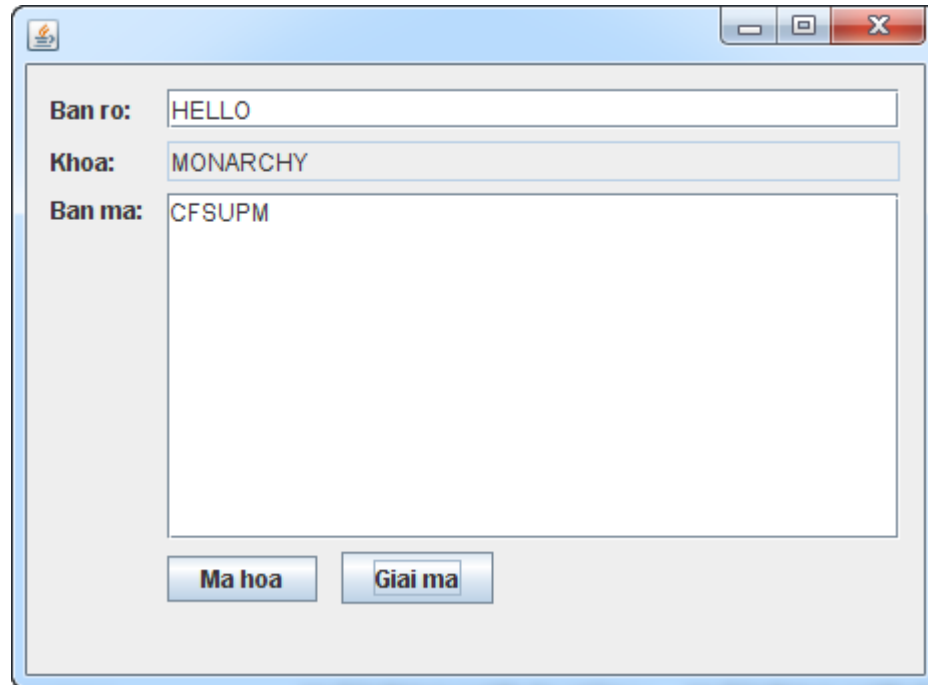
- Trước hết viết các chữ của từ khoá vào các hàng của ma trận bắt từ hàng thứ nhất.
- Nếu ma trận còn trống, viết các chữ khác trên bảng chữ cái chưa được sử dụng vào các ô còn lại. Có thể viết theo một trình tự qui ước trước, chẳng hạn từ đầu bảng chữ cái cho đến cuối.
- Vì có 26 chữ cái tiếng Anh, nên thiếu một ô. Thông thường ta dồn hai chữ nào đó vào một ô chung, chẳng hạn I và J.
- Giả sử sử dụng từ khoá MORNACHY. Lập ma trận khoá Playfair tương ứng như sau:

M	O	N	A	R
C	H	Y	B	D
E	F	G	I,J	K
L	P	Q	S	T
U	V	W	X	Z

Quy tắc mã hóa và giải mã

- Chia bản rõ thành từng cặp chữ. Nếu một cặp nào đó có hai chữ như nhau, thì ta chèn thêm một chữ lọc chẳng hạn X. Ví dụ, trước khi mã **“balloon”** biến đổi thành **“ba lx lo on”**.
- Nếu cả hai chữ trong cặp đều rơi vào cùng một hàng, thì mã mỗi chữ bằng chữ ở phía bên phải nó trong cùng hàng của ma trận khóa (cuộn vòng quanh từ cuối về đầu), chẳng hạn **“ar”** biến đổi thành **“RM”**.
- Nếu cả hai chữ trong cặp đều rơi vào cùng một cột, thì mã mỗi chữ bằng chữ ở phía bên dưới nó trong cùng cột của ma trận khóa (cuộn vòng quanh từ cuối về đầu), chẳng hạn **“mu”** biến đổi thành **“CM”**.
- Trong các trường hợp khác, mỗi chữ trong cặp được mã bởi chữ cùng hàng với nó và cùng cột với chữ cùng cặp với nó trong ma trận khóa. Chẳng hạn, **“hs”** mã thành **“BP”**, và **“ea”** mã thành **“IM”** hoặc **“JM”**.

Bước 1: Thiết Kế Form :



The screenshot shows a Java Swing window with a light blue title bar and standard Windows-style window controls (minimize, maximize, close). The window contains a form with three text input fields. The first field is labeled 'Ban ro:' and contains the text 'HELLO'. The second field is labeled 'Khoa:' and contains the text 'MONARCHY'. The third field is labeled 'Ban ma:' and contains the text 'CFSUPM'. Below the input fields, there are two buttons: 'Ma hoa' and 'Giai ma'.

Khai báo bảng khóa playfair ứng với khóa MONARCH khóa form:

```
public class Form3 extends javax.swing.JFrame {  
    char pf[][]={{'M','O','N','A','R'},  
                  {'C','H','Y','B','D'},  
                  {'E','F','G','I','K'},  
                  {'L','P','Q','S','T'},  
                  {'U','V','W','X','Z'}};  
};
```

Bước 2: Viết hàm xử lý sự kiện

a. Hàm xử lý sự kiện nút mã hóa

```
private void btnmahoa(java.awt.event.ActionEvent evt) {  
    String banro=this.txtbr.getText();  
    banro=banro.toUpperCase();  
    banro=banro.replace('J', 'I');  
    String banma=mahoa(banro);  
    this.txtbm.setText(banma);  
}
```

```

private String mahoa(String banro) {
    int n=banro.length();
    int i=0;
    String banma="";
    char a,b;
    while (i<n)
    {
        if(i==n-1)
        {
            a=banro.charAt(i);
            b='X';
            i++;
        }else{
            a=banro.charAt(i);
            b=banro.charAt(i+1);

            if(a==b)
            {
                b='X';
                i++;
            }else
                i+=2;

        }
        //thay the trong ban playfair
        banma+=thayThe(a,b);
    }
    return banma;
}

```

```

String thayThe(char a, char b)
{
    String vta=timViTri(a);
    String vtb=timViTri(b);
    char x,y;
    if(vta.charAt(0)==vtb.charAt(0)) //cung dong
    {
        x=pf[vta.charAt(0)-'0'][(vta.charAt(1)-'0')+1%5];
        y=pf[(vtb.charAt(0)-'0')][(vtb.charAt(1)-'0')+1%5];
        return x+""+y;
    }
    if(vta.charAt(1)==vtb.charAt(1)) //cung dong
    {
        x=pf[(vta.charAt(0)-'0')+1%5][vta.charAt(1)-'0'];
        y=pf[(vtb.charAt(0)-'0')+1%5][vtb.charAt(1)-'0'];
        return x+""+y;
    }
    x=pf[(vta.charAt(0)-'0')][(vtb.charAt(1)-'0')];
    y=pf[(vtb.charAt(0)-'0')][vta.charAt(1)-'0'];
    return x+""+y;
}

private String timViTri(char a) {
    for(int i=0;i<5;i++)
        for(int j=0;j<5;j++)
            if(pf[i][j]==a)
                return i+""+j;
    return "";
}

```

b. Hàm xử lý sự kiện nút giải mã

```

private void btngiaima(java.awt.event.ActionEvent evt) {
    String banma=this.txtbm.getText();
    String banro=giaima(banma);
    int n=banro.length();
    String br="";
    for(int i=0;i<n-2;i+=2)
        if(banro.charAt(i)==banro.charAt(i+2))
            br+=banro.charAt(i);
        else
            br+=banro.charAt(i)+""+banro.charAt(i+1);
    if(banro.charAt(n-1)=='X')
        br+=banro.charAt(n-2);
    else
        br+=banro.charAt(n-2); br+=banro.charAt(n-1);
    this.txtbr.setText(br);
}

```



```

private String giaima(String banma) {
    int n=banma.length();
    String banro="";
    char a,b;
    for(int i=0;i<n;i+=2)
    {
        a=banma.charAt(i);
        b=banma.charAt(i+1);
        //thay the trong ban playfair
        banro+=thayTheNguoc(a,b);
    }
    return banro;
}

String thayTheNguoc(char a, char b)
{
    String vta=timViTri(a);
    String vtb=timViTri(b);
    char x,y;
    if(vta.charAt(0)==vtb.charAt(0)) //cung dong
    {
        x=pf[vta.charAt(0)-'0'][(vta.charAt(1)-'0')-1+5)%5];
        y=pf[vtb.charAt(0)-'0'][(vtb.charAt(1)-'0')-1+5)%5];
        return x+""+y;
    }
    if(vta.charAt(1)==vtb.charAt(1)) //cung dong
    {
        x=pf[(vta.charAt(0)-'0')-1+5)%5][vta.charAt(1)-'0'];
        y=pf[(vtb.charAt(0)-'0')-1+5)%5][vtb.charAt(1)-'0'];
        return x+""+y;
    }
    x=pf[(vta.charAt(0)-'0')][vtb.charAt(1)-'0'];
    y=pf[vtb.charAt(0)-'0'][vta.charAt(1)-'0'];
    return x+""+y;
}

```

Bài 5: Viết chương trình mã hóa và giải mã văn bản với thuật toán mã hóa Transposition cipher.

Chương trình có thể thực hiện các chức năng sau:

Cho phép nhập văn bản vào hệ thống.

Cho phép nhập khóa bảo vệ văn bản.

Cho phép mở File và Ghi File.

Hướng dẫn :Hệ mã hóa đổi chỗ (Transposition Cipher)

Là hệ mã hóa trong đó các kí tự của bản gốc được giữ nguyên, nhưng vị trí bị thay đổi.

Đảo ngược toàn bộ plaintext : nghĩa là bản gốc được viết theo thứ tự ngược lại từ sau ra trước.

Ví dụ Plaintext: SECURE EMAIL

Bản mã: LIAMEERUCES

Mã hóa theo mẫu hình học: bản gốc được sắp xếp lại theo một mẫu hình học nào đó, thường là một mảng hoặc ma trận hai chiều.

Ví dụ: bản gốc ban đầu là BAO MAT

Ví dụ mã hoá theo mẫu hình học.

Cột 1	Cột 2	Cột 3
B	A	O
M	A	T

Nếu lấy các cột theo thứ tự 2, 3, 1. Bản mã sẽ là AAOTBM

Đổi chỗ cột: đổi chỗ các kí tự trong plaintext thành dạng hình chữ nhật theo cột.

Ví dụ : Bản gốc BAO MAT THU DIEN TU

Bản gốc được chuyển thành ma trận 3x5 như sau:

Bảng ví dụ mã hóa bằng phương pháp đổi chỗ cột

Cột 1	Cột 2	Cột 3	Cột 4	Cột 5
B	M	T	D	N
A	A	H	I	T
O	T	U	E	U

Vì có 5 cột nên chúng có thể được sắp lại theo $5! = 120$ cách khác nhau

Nếu ta chuyển vị các cột theo thứ tự 3, 5, 2, 4, 1 rồi lấy các kí tự theo hàng ngang ta sẽ thu được bản mã: TNMDBHTAIAUUTEO.

Hoán vị các kí tự của bản gốc theo chu kỳ cố định d: Nếu hàm f là hoán vị của một khối gồm d kí tự thì khóa mã hóa được biểu diễn bởi K(d, f)

Ví dụ: với d = 5, f hoán vị của dãy 12345 thành 35142

Bảng Hoán vị các kí tự của bản gốc theo chu kỳ cố định d

Vị trí ban đầu	Vị trí hoán vị	Nội dung mã hóa	Mã hóa
1	3	G	O
2	5	R	P
3	1	O	G
4	4	U	U
5	2	P	R

Theo bảng trên bản gốc ban đầu được mã hóa thành OPGUP.

Bước 1: Thiết Kế Form :

Bước 2: Viết hàm xử lý sự kiện

a. Hàm xử lý sự kiện Encrypt

```
private void mahoabtn(java.awt.event.ActionEvent evt) {  
    String k=this.txtkhoa.getText();  
    String ks[]=new String[6];  
    ks=k.split(",");  
    //System.out.println(ks.length);  
    int key[]=new int[6];  
    for(int i=0;i<6;i++)  
        key[i]=Integer.valueOf(ks[i])-1;  
    String sa=this.txtbr.getText();  
    String kq="";  
    int na=sa.length();  
    int d=0;  
    int c;  
    String s="";  
    int thieu=6-na%6;  
    for(int i=0;i<thieu;i++) sa=sa+" ";  
    while (d<na)  
    {  
        c=d+6;  
        s=sa.substring(d, c);  
        for(int i=0;i<6;i++)  
            kq=kq+s.charAt(key[i]);  
        d=d+6;  
    }  
    this.txtbm.setText(kq);  
}
```

b. Hàm xử lý sự kiện Dencrypt

```

private void btngiaima(java.awt.event.ActionEvent evt) {
    String k=this.txtkhoa.getText();
    String ks[]=new String[6];
    ks=k.split(",");
    int key[]=new int[6];
    for(int i=0;i<6;i++)
        key[i]=Integer.valueOf(ks[i])-1;
    int keyt1[]=new int[6];
    for(int i=0;i<6;i++)
        keyt1[key[i]]=i;
    String sa=this.txtbm.getText();
    String kq="";
    int na=sa.length();
    int d=0;
    int c;
    String s="";
    while (d<na)
    {
        c=d+6;
        s=sa.substring(d, c);
        for(int i=0;i<6;i++)
            kq=kq+s.charAt(keyt1[i]);
        d=d+6;
    }
    this.txtbr.setText(kq);
}

```

Bài Tập: Viết phần mềm mã hóa văn bản với các thuật toán mã hóa trên.

Chương trình có thể thực hiện các chức năng sau:

Cho phép nhập văn bản vào hệ thống.

Cho phép nhập khóa bảo vệ văn bản.

Cho phép mở File và Ghi File.

Cho phép bên gửi mã hóa dữ liệu và bên nhận mã hóa dữ liệu với khóa K.

