/\*

Program to use the Caesar cipher and encrypt a set of passwords.

Programmer: Hong Zhang, File Name: Encryptor .java

\*/

**Original source code:**

**import** javax.swing.\*;

**public** **class** Encryptor {

**public** String cipherText;

**private** String plainText;

**private** **int** shift;

**public** Encryptor() {

plainText = **null**;

shift = 0;

}

**public** **static** **void** main(String [] args) {

// encryption block

Encryptor d = **new** Encryptor();

String strCipherText = d.Encrypt();

System.***out***.println(strCipherText);

// decrypt block

Encryptor d = **new** Encryptor();

// cipher text becomes the input text to the Decrypt method

d.cipherText = strCipherText;

String strPlainText = d.Decrypt();

System.***out***.println(strPlainText);

System.*exit*(0);

}

**public** String Encrypt() {

plainText =

((String)JOptionPane.*showInputDialog*("enter words " +

"to encrypt")).toLowerCase().trim();

shift =

Integer.*parseInt*(JOptionPane.*showInputDialog*("enter offset"));

**int** offset = 0;

**int** newOffset = 0;

String alphabet = "abcdefghijklmnopqrstuvwxyz";

StringBuffer sb = **new** StringBuffer();

**int** index = plainText.length();

**for**(**int** i = 0; i < index; i++) {

String temp = "" + plainText.charAt(i);

offset = alphabet.indexOf(temp);

offset += shift;

**if**(offset > 25){

newOffset = offset % 26;

sb.append(alphabet.charAt(newOffset));

}

**else** {

sb.append(alphabet.charAt(offset));

}

}

**return** sb.toString();

}

**public** String Decrypt() {

plainText = cipherText;

shift =

Integer.*parseInt*(JOptionPane.*showInputDialog*("enter offset"));

**int** offset = 0;

**int** newOffset = 0;

String alphabet = "abcdefghijklmnopqrstuvwxyz";

StringBuffer sb = **new** StringBuffer();

**int** index = plainText.length();

**for**(**int** i = 0; i < index; i++) {

String temp = "" + plainText.charAt(i);

offset = alphabet.indexOf(temp);

offset -= shift;

**if**(offset > 25){

newOffset = offset % 26;

sb.append(alphabet.charAt(newOffset));

}

**else** {

sb.append(alphabet.charAt(offset));

}

}

**return** sb.toString();

}

}

**Modified source code:**

**import** javax.swing.\*;

**public** **class** Encryptor {

**public** String cipherText;

**public** **static** String *alphabet* = "abcdefghijklmnopqrstuvwxyz";

**private** String plainText;

**private** **int** shift;

**public** Encryptor() {

plainText = **null**;

shift = 0;

}

**public** **static** **void** main(String [] args) {

// encryption block

Encryptor e = **new** Encryptor();

String strCipherText = e.Encrypt();

System.***out***.println("encrypted text");

System.***out***.println(strCipherText);

// decrypt block

Encryptor d = **new** Encryptor();

// cipher text becomes the input text to the Decrypt method

d.cipherText = strCipherText;

String strPlainText = d.Decrypt();

System.***out***.println("decrypted text");

System.***out***.println(strPlainText);

System.*exit*(0);

}

**public** String Encrypt() {

plainText =

((String)JOptionPane.*showInputDialog*("enter words " +

"to encrypt")).toLowerCase().trim();

shift =

Integer.*parseInt*(JOptionPane.*showInputDialog*("enter "

+ "offset to encrypt"));

**int** offset = 0;

**int** newOffset = 0;

StringBuffer sb = **new** StringBuffer();

**int** index = plainText.length();

**for**(**int** i = 0; i < index; i++) {

String temp = "" + plainText.charAt(i);

offset = *alphabet*.indexOf(temp);

offset += shift;

**if**(offset > 25){

newOffset = offset % 26;

sb.append(*alphabet*.charAt(newOffset));

}

**else** {

sb.append(*alphabet*.charAt(offset));

}

}

**return** sb.toString();

}

**public** String Decrypt() {

plainText = cipherText;

shift =

Integer.*parseInt*(JOptionPane.*showInputDialog*("enter "

+ "offset to decrypt"));

**int** offset = 0;

**int** newOffset = 0;

StringBuffer sb = **new** StringBuffer();

**int** index = plainText.length();

**for**(**int** i = 0; i < index; i++) {

String temp = "" + plainText.charAt(i);

offset = *alphabet*.indexOf(temp);

offset -= shift;

**if**(offset < 0){

newOffset = offset + 26;

sb.append(*alphabet*.charAt(newOffset));

}

**else** {

sb.append(*alphabet*.charAt(offset));

}

}

**return** sb.toString();

}

}

Sample Output:







