/\*字符串 DESede(3DES) 加密\*/  
import java.security.Security;  
  
import javax.crypto.Cipher;  
import javax.crypto.SecretKey;  
import javax.crypto.spec.SecretKeySpec;  
  
public class DES3 {  
      
    private static final String Algorithm = "DESede"; // 定义 加密算法,可用  
  
    // DES,DESede,Blowfish  
  
    // keybyte为加密密钥，长度为24字节  
    // src为被加密的数据缓冲区（源）  
    public static byte[] encryptMode(byte[] keybyte, byte[] src) {  
        try {  
            // 生成密钥  
            SecretKey deskey = new SecretKeySpec(keybyte, Algorithm);  
            // 加密  
            Cipher c1 = Cipher.getInstance(Algorithm);  
            c1.init(Cipher.ENCRYPT\_MODE, deskey);  
            return c1.doFinal(src);  
        } catch (java.security.NoSuchAlgorithmException e1) {  
            e1.printStackTrace();  
        } catch (javax.crypto.NoSuchPaddingException e2) {  
            e2.printStackTrace();  
        } catch (java.lang.Exception e3) {  
            e3.printStackTrace();  
        }  
        return null;  
    }  
  
    // keybyte为加密密钥，长度为24字节  
    // src为加密后的缓冲区  
    public static byte[] decryptMode(byte[] keybyte, byte[] src) {  
        try {  
            // 生成密钥  
            SecretKey deskey = new SecretKeySpec(keybyte, Algorithm);  
            // 解密  
            Cipher c1 = Cipher.getInstance(Algorithm);  
            c1.init(Cipher.DECRYPT\_MODE, deskey);  
            return c1.doFinal(src);  
        } catch (java.security.NoSuchAlgorithmException e1) {  
            e1.printStackTrace();  
        } catch (javax.crypto.NoSuchPaddingException e2) {  
            e2.printStackTrace();  
        } catch (java.lang.Exception e3) {  
            e3.printStackTrace();  
        }  
        return null;  
    }  
  
    // 转换成十六进制字符串  
    public static String byte2hex(byte[] b) {  
        String hs = "";  
        String stmp = "";  
        for (int n = 0; n < b.length; n++) {  
            stmp = (java.lang.Integer.toHexString(b[n] & 0XFF));  
            if (stmp.length() == 1)  
                hs = hs + "0" + stmp;  
            else  
                hs = hs + stmp;  
            if (n < b.length - 1)  
                hs = hs + ":";  
        }  
        return hs.toUpperCase();  
    }  
  
    public static void main(String[] args) {  
  
        // 添加新安全算法,如果用JCE就要把它添加进去  
        Security.addProvider(new com.sun.crypto.provider.SunJCE());  
        final byte[] keyBytes = { 0x11, 0x22, 0x4F, 0x58, (byte) 0x88, 0x10,  
                0x40, 0x38, 0x28, 0x25, 0x79, 0x51, (byte) 0xCB, (byte) 0xDD,  
                0x55, 0x66, 0x77, 0x29, 0x74, (byte) 0x98, 0x30, 0x40, 0x36,  
                (byte) 0xE2 }; // 24字节的密钥  
  
        String szSrc = "This is a 3DES test. 测试";  
        System.out.println("加密前的字符串:" + szSrc);  
  
        byte[] encoded = encryptMode(keyBytes, szSrc.getBytes());  
        System.out.println("加密后的字符串:" + new String(encoded));  
  
        byte[] srcBytes = decryptMode(keyBytes, encoded);  
        System.out.println("解密后的字符串:" + (new String(srcBytes)));  
    }  
}