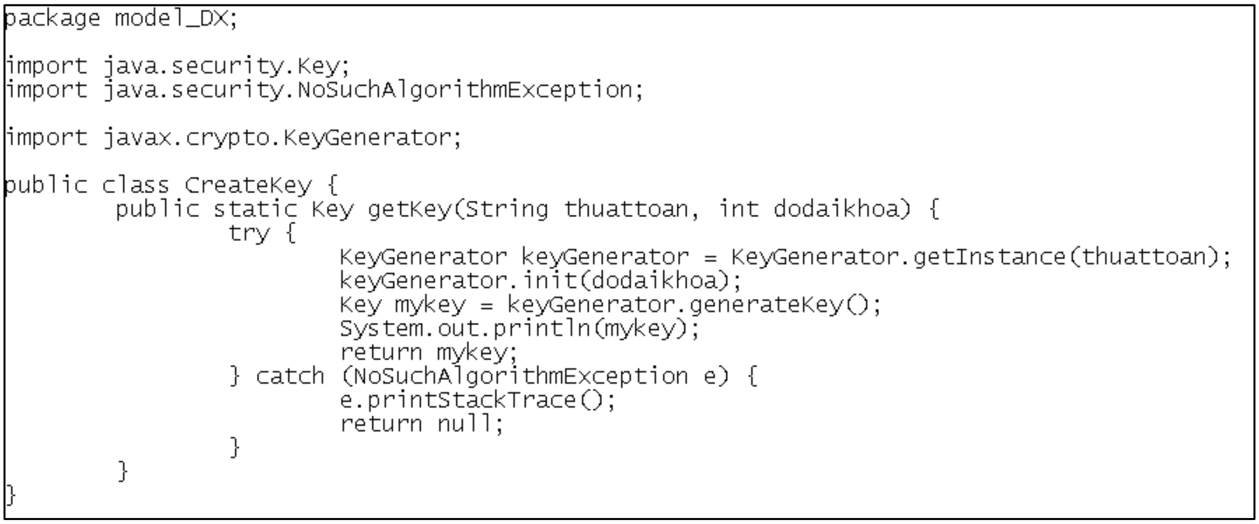
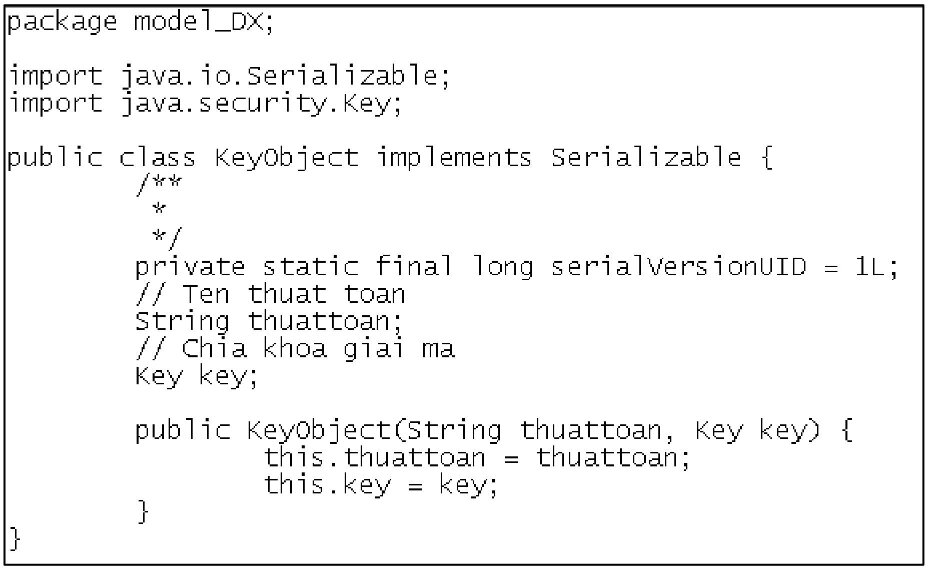
THỰC HÀNH MÃ HÓA ĐỐI XỨNG TRÊN FILE

Class: CreateKey



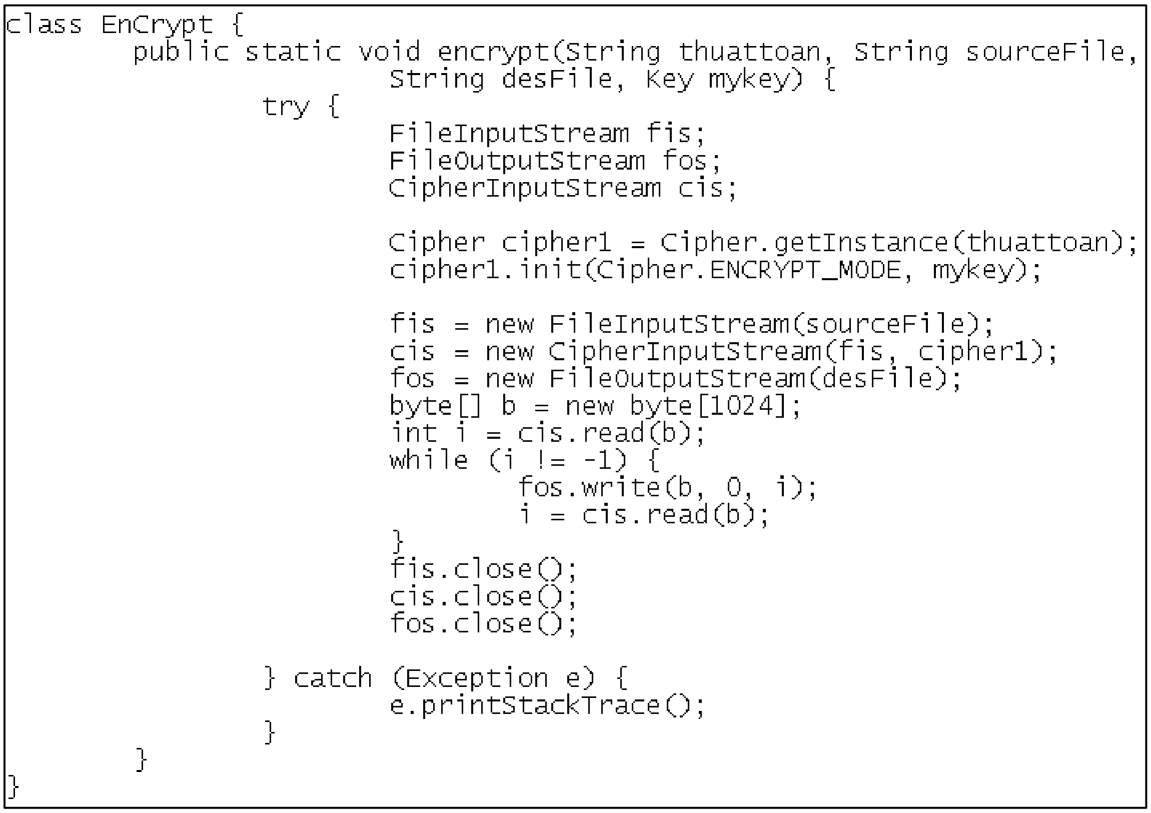
```
package model_DX;

import java.security.Key;
import java.security.NoSuchAlgorithmException;

import javax.crypto.KeyGenerator;

public class CreateKey {
        public static Key getKey(String thuattoan, int dodaikhoa) {
                try {
                        KeyGenerator keyGenerator = KeyGenerator.getInstance(thuattoan);
                        keyGenerator.init(dodaikhoa);
                        Key mykey = keyGenerator.generateKey();
                        System.out.println(mykey);
                        return mykey;
                } catch (NoSuchAlgorithmException e) {
                        e.printStackTrace();
                        return null;
                }
        }
}
```

Class: KeyObject



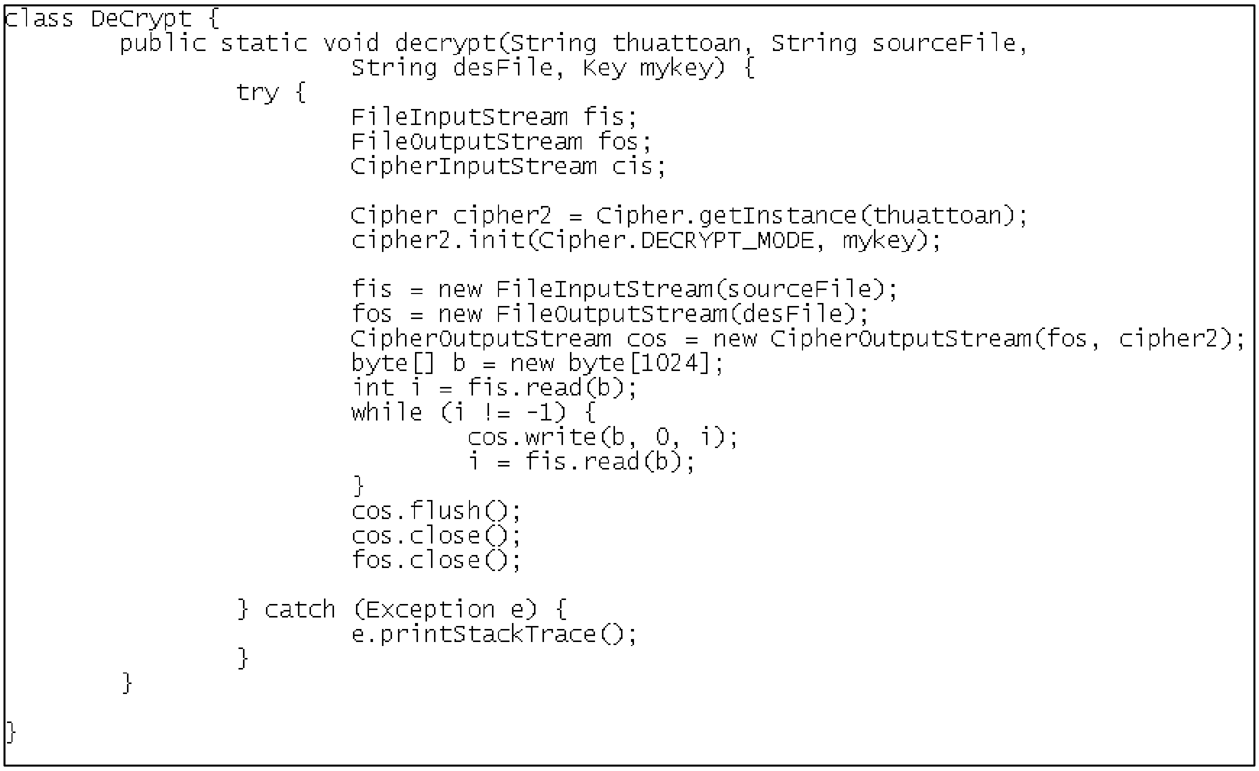
```
package model_DX;

import java.io.Serializable;
import java.security.Key;

public class KeyObject implements Serializable {
        /**
         *
         */
        private static final long serialVersionUID = 1L;
        // Ten thuat toan
        String thuattoan;
        // Chia khoa giai ma
        Key key;

        public KeyObject(String thuattoan, Key key) {
                this.thuattoan = thuattoan;
                this.key = key;
        }
}
```

Class: Encrypt



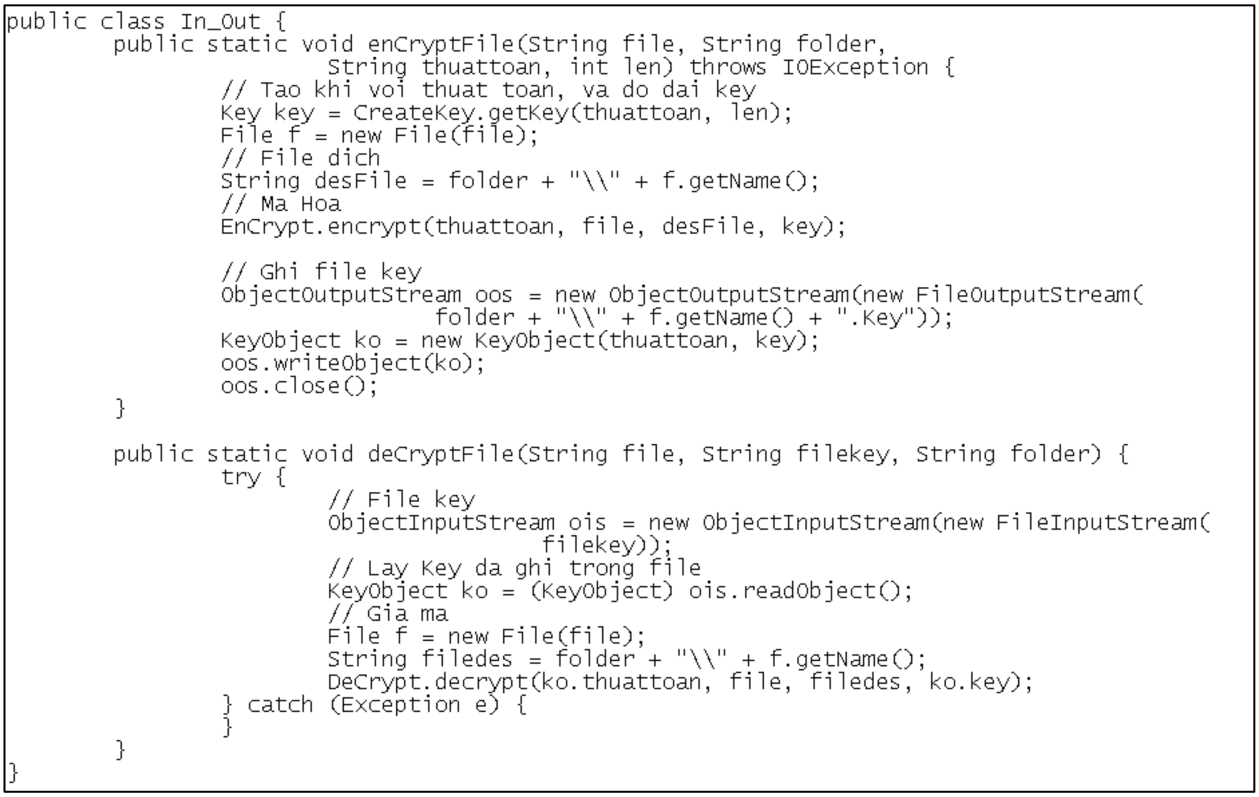
```
class EnCrypt {
        public static void encrypt(String thuattoan, String sourceFile,
                        String desFile, Key mykey) {
                try {
                        FileInputStream fis;
                        FileOutputStream fos;
                        CipherInputStream cis;

                        Cipher cipher1 = Cipher.getInstance(thuattoan);
                        cipher1.init(Cipher.ENCRYPT_MODE, mykey);

                        fis = new FileInputStream(sourceFile);
                        cis = new CipherInputStream(fis, cipher1);
                        fos = new FileOutputStream(desFile);
                        byte[] b = new byte[1024];
                        int i = cis.read(b);
                        while (i != -1) {
                                fos.write(b, 0, i);
                                i = cis.read(b);
                        }
                        fis.close();
                        cis.close();
                        fos.close();

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

Class: Decrypt



```
class DeCrypt {
        public static void decrypt(String thuattoan, String sourceFile,
                        String desFile, Key mykey) {
                try {
                        FileInputStream fis;
                        FileOutputStream fos;
                        CipherInputStream cis;

                        Cipher cipher2 = Cipher.getInstance(thuattoan);
                        cipher2.init(Cipher.DECRYPT_MODE, mykey);

                        fis = new FileInputStream(sourceFile);
                        fos = new FileOutputStream(desFile);
                        CipherOutputStream cos = new CipherOutputStream(fos, cipher2);
                        byte[] b = new byte[1024];
                        int i = fis.read(b);
                        while (i != -1) {
                                cos.write(b, 0, i);
                                i = fis.read(b);
                        }
                        cos.flush();
                        cos.close();
                        fos.close();

                } catch (Exception e) {
                        e.printStackTrace();
                }
        }

}
```

Class: In_Out



```
public class In_Out {
        public static void enCryptFile(String file, String folder,
                        String thuattoan, int len) throws IOException {
                // Tao khi voi thuat toan, va do dai key
                Key key = CreateKey.getKey(thuattoan, len);
                File f = new File(file);
                // File dich
                String desFile = folder + "\\" + f.getName();
                // Ma Hoa
                EnCrypt.encrypt(thuattoan, file, desFile, key);

                // Ghi file key
                ObjectOutputStream oos = new ObjectOutputStream(new FileOutputStream(
                                folder + "\\" + f.getName() + ".Key"));
                KeyObject ko = new KeyObject(thuattoan, key);
                oos.writeObject(ko);
                oos.close();
        }

        public static void deCryptFile(String file, String filekey, String folder) {
                try {
                        // File key
                        ObjectInputStream ois = new ObjectInputStream(new FileInputStream(
                                        filekey));
                        // Lay Key da ghi trong file
                        KeyObject ko = (KeyObject) ois.readObject();
                        // Gia ma
                        File f = new File(file);
                        String filedes = folder + "\\" + f.getName();
                        DeCrypt.decrypt(ko.thuattoan, file, filedes, ko.key);
                } catch (Exception e) {
                }
        }
}
```