Ministerul Educaţiei al Republicii Moldova

Universitatea Tehnică a Moldovei

Facultatea Calculatoare Informatică şi Microelectronică

Departamentul Ingineria Software și Automatică

**Raport**

Disciplina: Securitatea informațională.

Lucrarea de laborator nr. 1

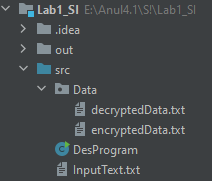
**Tema:** Securitatea informațională.

A efectuat: st.gr. TI-194, Zavorot Daniel

A verificat: asist. univ. Răducanu Octavian

Chișinău - 2022

Structura poriectului:



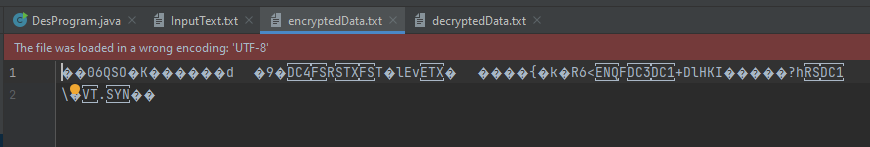
DesProgram.java:

import java.io.IOException;  
import java.io.InputStream;  
import java.io.OutputStream;  
import java.nio.file.Files;  
import java.nio.file.Paths;  
import java.security.InvalidAlgorithmParameterException;  
import java.security.InvalidKeyException;  
import java.security.NoSuchAlgorithmException;  
import java.security.spec.AlgorithmParameterSpec;  
import javax.crypto.Cipher;  
import javax.crypto.CipherInputStream;  
import javax.crypto.CipherOutputStream;  
import javax.crypto.KeyGenerator;  
import javax.crypto.NoSuchPaddingException;  
import javax.crypto.SecretKey;  
import javax.crypto.spec.IvParameterSpec;  
  
public class DesProgram {  
 private static Cipher *encrypt*;  
 private static Cipher *decrypt*;  
 private static final byte[] *initialization\_vector* = {22, 33, 11, 44, 55, 99, 66, 77};  
  
 public static void main(String[] args) {  
  
 String textFile = "src/InputText.txt";  
 String encryptedData = "src/Data/encryptedData.txt";  
 String decryptedData = "src/Data/decryptedData.txt";  
 try {  
 SecretKey secretKey = KeyGenerator.*getInstance*("DES").generateKey();  
 AlgorithmParameterSpec aps = new IvParameterSpec(*initialization\_vector*);  
  
 *encrypt* = Cipher.*getInstance*("DES/CBC/PKCS5Padding");  
 *encrypt*.init(Cipher.*ENCRYPT\_MODE*, secretKey, aps);  
  
 *decrypt* = Cipher.*getInstance*("DES/CBC/PKCS5Padding");  
 *decrypt*.init(Cipher.*DECRYPT\_MODE*, secretKey, aps);  
  
 *encryption*(Files.*newInputStream*(Paths.*get*(textFile)), Files.*newOutputStream*(Paths.*get*(encryptedData)));  
 *decryption*(Files.*newInputStream*(Paths.*get*(encryptedData)), Files.*newOutputStream*(Paths.*get*(decryptedData)));  
  
 System.*out*.println("The encrypted and decrypted files have been created successfully.");  
 } catch (NoSuchAlgorithmException | NoSuchPaddingException | InvalidKeyException |  
 InvalidAlgorithmParameterException | IOException e) {  
 e.printStackTrace();  
 }  
 }  
  
 private static void encryption(InputStream input, OutputStream output) throws IOException {  
 output = new CipherOutputStream(output, *encrypt*);  
 *writeBytes*(input, output);  
 }  
  
 private static void decryption(InputStream input, OutputStream output) throws IOException {  
 input = new CipherInputStream(input, *decrypt*);  
 *writeBytes*(input, output);  
 }  
  
 private static void writeBytes(InputStream input, OutputStream output) throws IOException {  
 byte[] writeBuffer = new byte[512];  
 int readBytes = 0;  
 while ((readBytes = input.read(writeBuffer)) >= 0) {  
 output.write(writeBuffer, 0, readBytes);  
 }  
 output.close();  
 input.close();  
 }  
}

InputText.txt:



encryptedData.txt:



decryptedData.txt

