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. 3

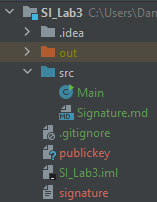
**Tema:** Securitatea informațională.

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

A verificat: asist. univ. Răducanu Octavian

Chișinău - 2022

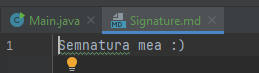
Structura poriectului:



Codul:

import java.nio.file.Files;  
import java.nio.file.Paths;  
import java.security.KeyPair;  
import java.security.KeyPairGenerator;  
import java.security.PrivateKey;  
import java.security.SecureRandom;  
import java.security.Signature;  
  
public class Main {  
 public static void main(String[] args) {  
 try {  
 // Get instance and initialize a KeyPairGenerator object.  
 KeyPairGenerator keyGen = KeyPairGenerator.*getInstance*("DSA", "SUN");  
 SecureRandom random = SecureRandom.*getInstance*("SHA1PRNG", "SUN");  
 keyGen.initialize(1024, random);  
  
 // Get a PrivateKey from the generated key pair.  
 KeyPair keyPair = keyGen.generateKeyPair();  
 PrivateKey privateKey = keyPair.getPrivate();  
  
 // Get an instance of Signature object and initialize it.  
 Signature signature = Signature.*getInstance*("SHA1withDSA", "SUN");  
 signature.initSign(privateKey);  
  
 // Supply the data to be signed to the Signature object  
 // using the update() method and generate the digital  
 // signature.  
 byte[] bytes = Files.*readAllBytes*(Paths.*get*("src/Signature.md"));  
 signature.update(bytes);  
 byte[] digitalSignature = signature.sign();  
  
 // Save digital signature and the public key to a file.  
 Files.*write*(Paths.*get*("signature"), digitalSignature);  
 Files.*write*(Paths.*get*("publickey"), keyPair.getPublic().getEncoded());  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
}

Signature.md:



signature:



publickey:

