Національний технічний університет України

«Київський Політехнічний Інститут»

Факультет інформатики і обчислювальної техніки

Кафедра обчислювальної техніки

**Лабораторна робота №7**

**З предмету «Захист інформації в комп’ютерних системах»**

Виконав:

Студент  
IV курсу ФІОТ  
групи ІО-12  
Бута С. О.

Залікова книжка №1205

Київ-2015

**Завдання**

Алгоритм шифрування даних RSA.

**Звіт**

Лістинг:

**package** sample;  
  
**import** javafx.event.Event;  
**import** javafx.scene.control.TextArea;  
**import** javafx.scene.control.TextField;  
**import** sun.misc.BASE64Decoder;  
**import** sun.misc.BASE64Encoder;  
  
**import** javax.crypto.BadPaddingException;  
**import** javax.crypto.Cipher;  
**import** javax.crypto.IllegalBlockSizeException;  
**import** javax.crypto.NoSuchPaddingException;  
**import** java.io.IOException;  
**import** java.nio.charset.Charset;  
**import** java.security.InvalidKeyException;  
**import** java.security.KeyPair;  
**import** java.security.KeyPairGenerator;  
**import** java.security.NoSuchAlgorithmException;  
  
**public class** Controller {  
 **public** TextField **publicKeyTextField**;  
 **public** TextField **privateKeyTextField**;  
 **public** TextArea **simpleTextArea**;  
 **public** TextArea **encryptedTextArea**;  
  
 KeyPair **keys** = **null**;  
  
 **public void** generateKeysClick(Event event) {  
 **try** {  
 **keys** = KeyPairGenerator.*getInstance*(**"RSA"**).generateKeyPair();  
  
 BASE64Encoder encoder = **new** sun.misc.BASE64Encoder();  
 **publicKeyTextField**.setText(encoder.encode(**keys**.getPublic().getEncoded()));  
 **privateKeyTextField**.setText(encoder.encode(**keys**.getPrivate().getEncoded()));  
 } **catch** (NoSuchAlgorithmException e) {  
 e.printStackTrace();  
 }  
 }  
  
 **public void** encryptClick(Event event) {  
 **if**(**keys** == **null**){  
 generateKeysClick(event);  
 }  
 **byte**[] toEncrypt = **simpleTextArea**.getText().getBytes(Charset.*forName*(**"UTF-8"**));  
 **try** {  
 Cipher cipher = Cipher.*getInstance*(**"RSA"**);  
 cipher.init(Cipher.***ENCRYPT\_MODE***, **keys**.getPublic());  
 **byte**[] encrypted = cipher.doFinal(toEncrypt);  
 BASE64Encoder encoder = **new** sun.misc.BASE64Encoder();  
 String toArea = encoder.encode(encrypted);  
 **encryptedTextArea**.setText(toArea);  
 System.***out***.println(**"Encrypted length: "** + **new** String(encrypted).length());  
 } **catch** (NoSuchAlgorithmException e) {  
 e.printStackTrace();  
 } **catch** (NoSuchPaddingException e) {  
 e.printStackTrace();  
 } **catch** (IllegalBlockSizeException e) {  
 e.printStackTrace();  
 } **catch** (BadPaddingException e) {  
 e.printStackTrace();  
 } **catch** (InvalidKeyException e) {  
 e.printStackTrace();  
 }  
 }  
  
 **public void** decryptClick(Event event) **throws** IOException {  
 **if**(**keys** == **null**){  
 **return**;  
 }  
 BASE64Decoder decoder = **new** BASE64Decoder();  
 **byte**[] toDecrypt = decoder.decodeBuffer(**encryptedTextArea**.getText());  
 System.***out***.println(**"To decrypt length: "** + **new** String(toDecrypt).length());  
 **try** {  
 Cipher cipher = Cipher.*getInstance*(**"RSA"**);  
 cipher.init(Cipher.***DECRYPT\_MODE***, **keys**.getPrivate());  
 **byte**[] decrypted = cipher.doFinal(toDecrypt);  
 **simpleTextArea**.setText(**new** String(decrypted));  
 } **catch** (NoSuchAlgorithmException e) {  
 e.printStackTrace();  
 } **catch** (NoSuchPaddingException e) {  
 e.printStackTrace();  
 } **catch** (InvalidKeyException e) {  
 e.printStackTrace();  
 } **catch** (BadPaddingException e) {  
 e.printStackTrace();  
 } **catch** (IllegalBlockSizeException e) {  
 e.printStackTrace();  
 }  
 }  
}

Вікно програми:

