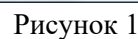


Проект будет реализовывать форму с интерфейсом, показанную на Рисунке 1.



Задание № 1

-
-
- Steps**

 1. Choose Project
 2. **Name and Location**

Name and Location

Project Name:

Project Location:

Project Folder:

Artifact Id:

Group Id:

Version:

Package: (Optional)

Additional Creation Properties:

Key	Value
javaafx-version	13
javaafx-maven-plugin-version	0.0.4
add-debug-configuration	Y

Рисунок 2

2. На втором шаге укажите имя проекта **MyCryptoFX**, расположение, и Group Id - **org.openjfx**.
3. Удалите файл **SystemInfo.java** и переименуйте главный класс проекта – **CryptoFX**.
4. В файле **CryptoFX.java** удалите сгенерированные заготовки кода и наберите новый код, Рисунок 3.

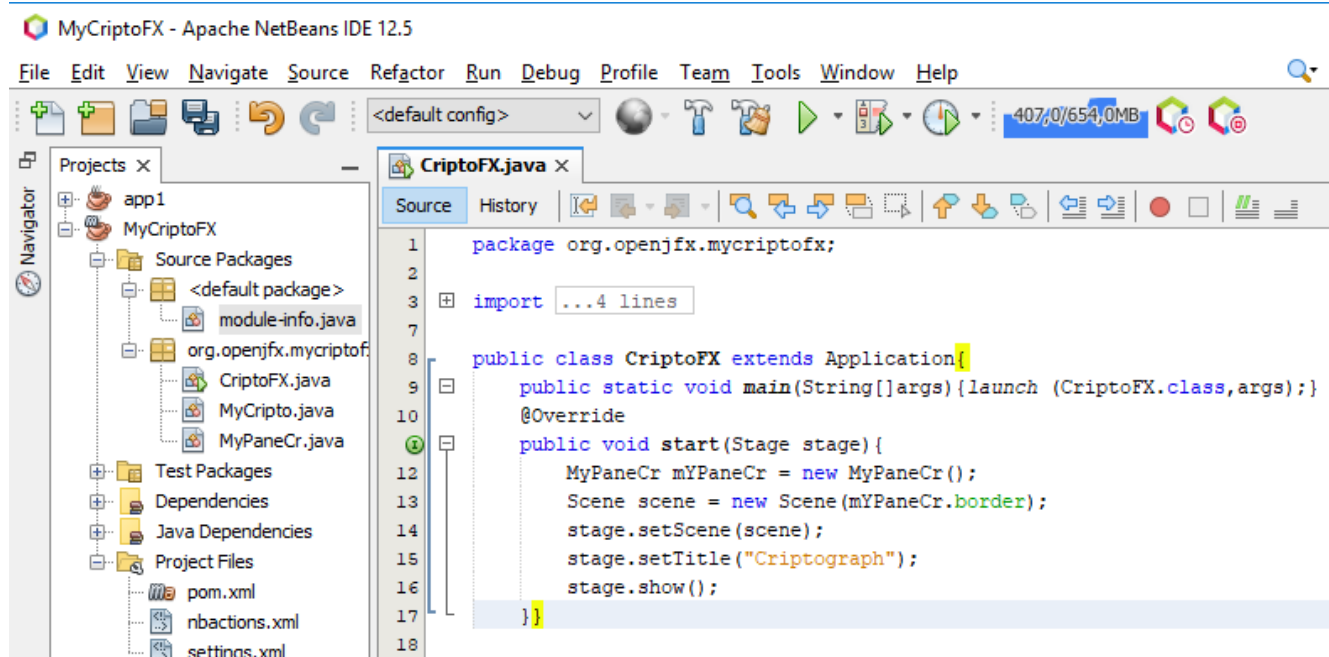
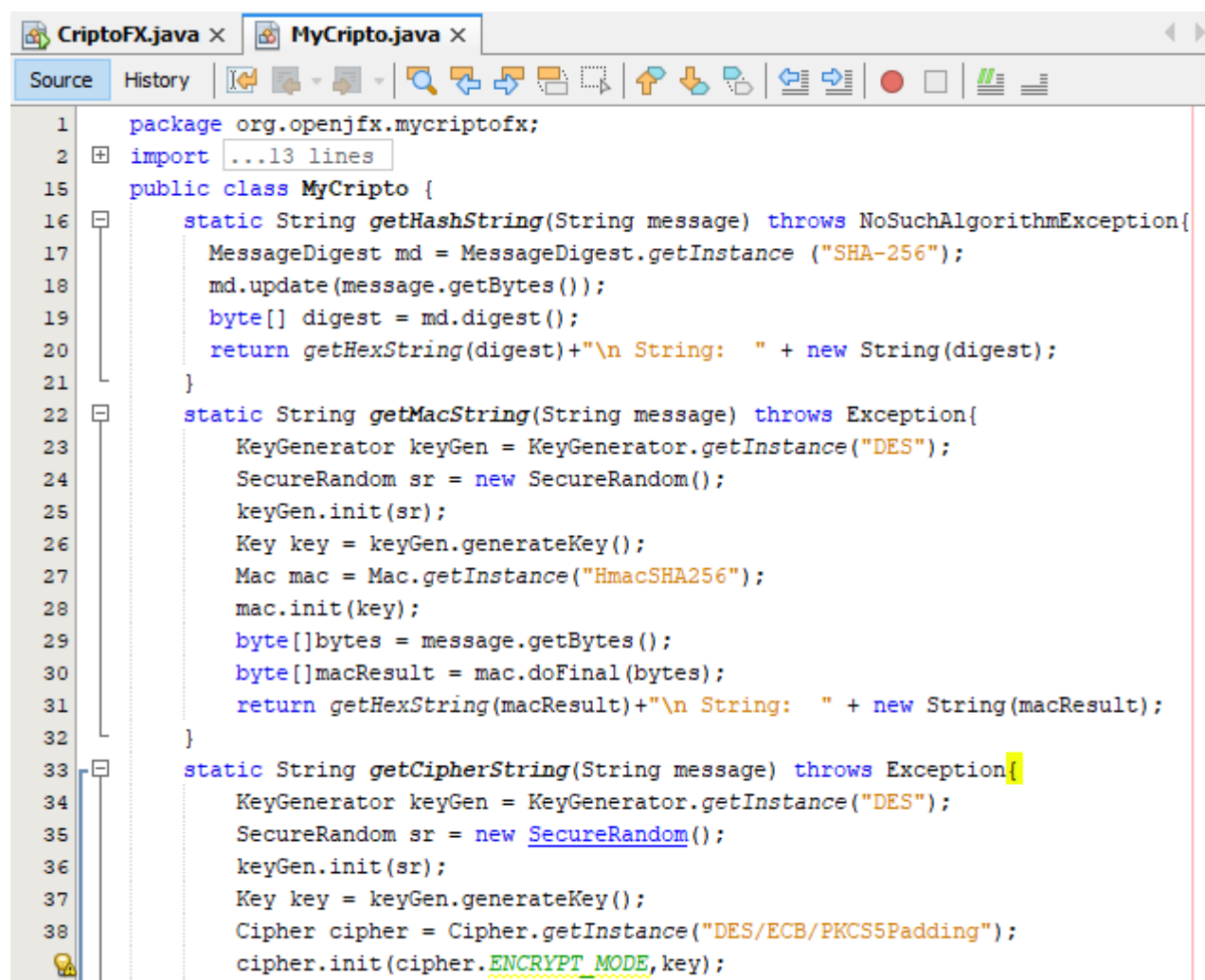


Рисунок 3

Разработка класса с криптографическими методами

Задание № 2

Создайте в пакете *org.openjfx.mycryptofx* новый класс *MyCrypto* для определения методов криптографии, Рисунок 4.



```

40     byte[] bytes = message.getBytes();
41     byte[] cipherResult = cipher.doFinal(bytes);
42     cipher.init(cipher.DECRYPT_MODE, key);
43     byte[] bytes1 = cipher.doFinal(cipherResult);
44     String str = new String(bytes1);
45     String str1 = getHexString(cipherResult);
46     str=str1+"\n исходный текст: " +str;
47     return str;
48 }
49 static String getPairCipherString(String message) throws Exception {
50     KeyPairGenerator keyPairGen = KeyPairGenerator.getInstance("RSA");
51     keyPairGen.initialize (2048);
52     KeyPair pair = keyPairGen.generateKeyPair();
53     PublicKey publicKey = pair.getPublic();
54     Cipher cipher = Cipher.getInstance("RSA/ECB/PKCS1Padding");
55     cipher.init(Cipher.ENCRYPT_MODE, publicKey);
56     byte[] bytes = message.getBytes();
57     cipher.update(bytes);
58     byte[] cipherResult = cipher.doFinal();
59     cipher.init(Cipher.DECRYPT_MODE, pair.getPrivate());
60     byte[] bytes1=cipher.doFinal(cipherResult);
61     String str = new String(bytes1);
62     String str1=getHexString(cipherResult);
63     str=str1+"\n исходный текст: " + str;
64     return str;
65 }
66 static String getElectSigmatString(String message) throws Exception {
67     KeyPairGenerator keyPairGen = KeyPairGenerator.getInstance("DSA");
68     keyPairGen.initialize (2048);
69     KeyPair pair = keyPairGen.generateKeyPair();
70     PrivateKey privKey = pair.getPrivate();
71     Signature sign = Signature.getInstance("SHA256withDSA");
72     sign.initSign(privKey);
73     byte[] bytes = message.getBytes();
74     sign.update(bytes);
75     byte[] signature = sign.sign();
76     sign.initVerify(pair.getPublic());
77     sign.update(bytes);
78     boolean bool = sign.verify(signature);
79     String str;
80     if(bool) str="Signature verified"; else str="Signature failed";
81     String str1=getHexString(signature);
82     str=str1+"\n"+str;
83     return str;
84 }
85 private static String getHexString(byte[] bytes){
86     StringBuilder hexString = new StringBuilder();
87     for (int i=0;i<bytes.length;i++) {
88         hexString.append(Integer.toHexString(0xFF & bytes[i]));
89     }
90     return "HexStr: " + hexString.toString();
91 }

```

Рисунок 4

Разработка классов реализации панелей для отображения данных

Задание № 3

1. Создайте в пакете *org.openjfx.mycryptofx* новый класс *MyPaneGr* для отображения данных, см. Рисунок 5.

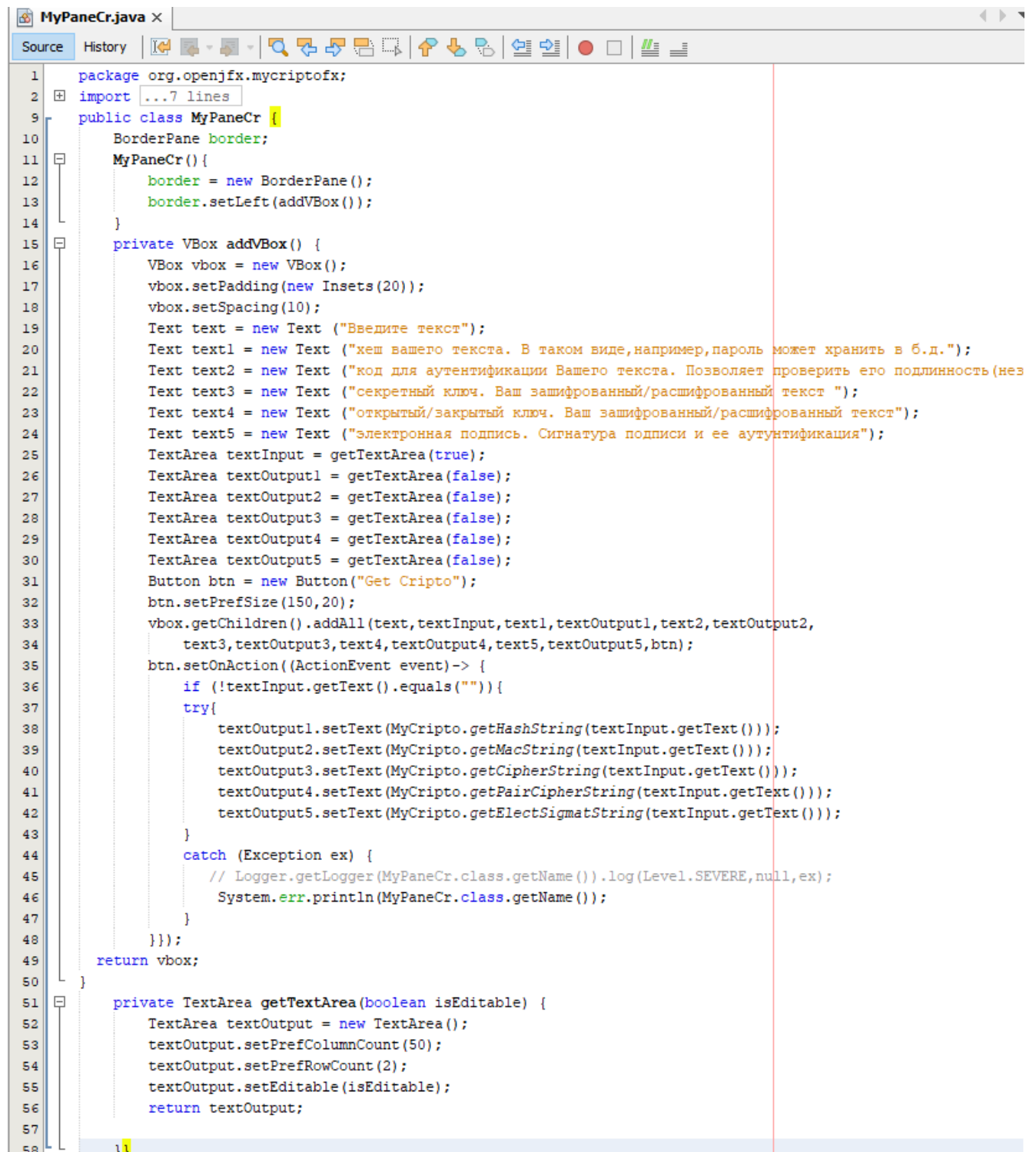


Рисунок 5

2. Проверьте версии зависимостей в файле *pom.xml* (папка *Project Files*) созданным средой программирования, и подправьте имя главного класса, Рисунок 6.
3. Запустите проект на выполнение.

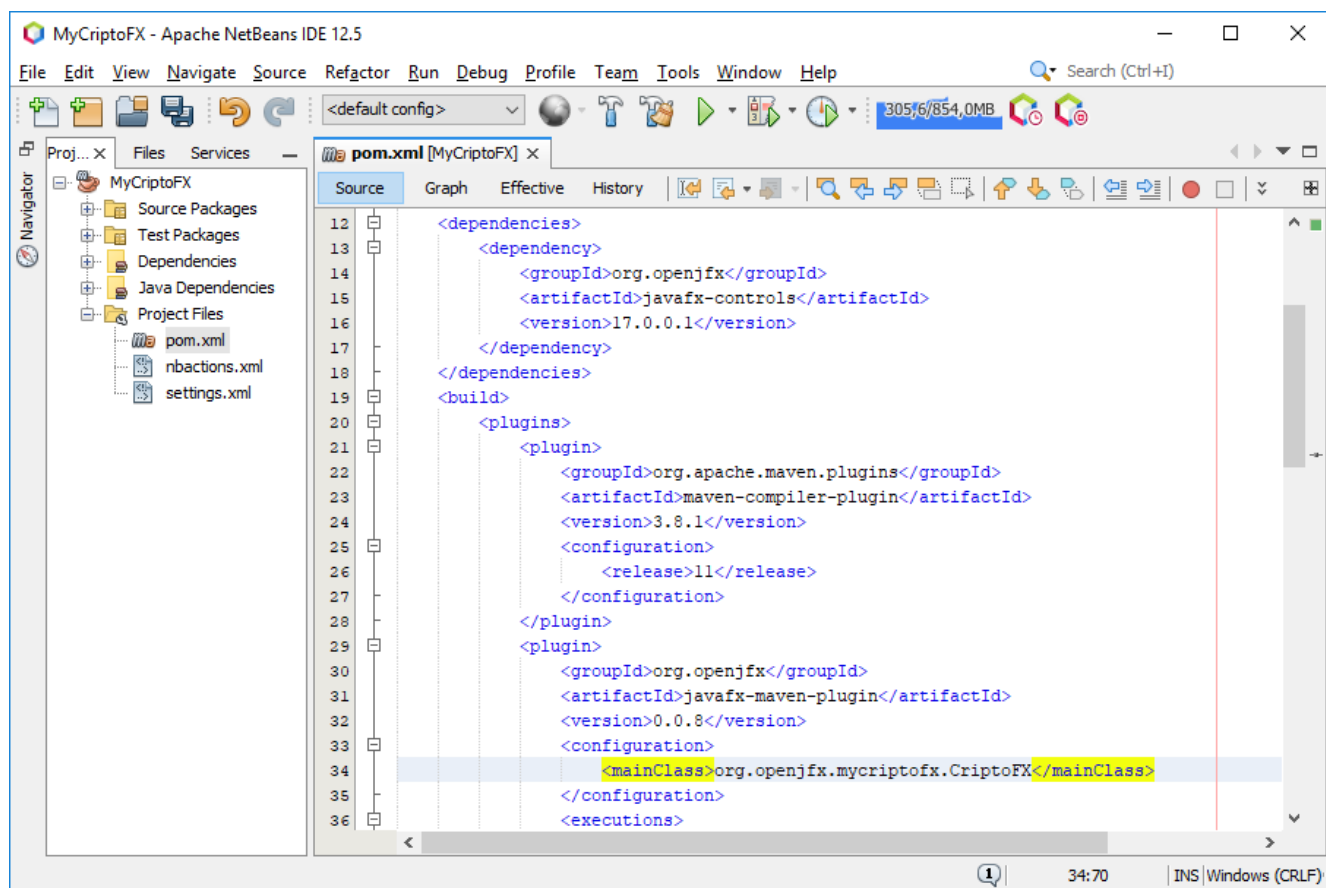


Рисунок 6

Более подробно о разработке **Maven** проектов с использованием **JavaFX** вы можете почитать на сайте: <https://openjfx.io/openjfx-docs/> в разделе JavaFX and NetBeans, [Non-modular with Maven](#).