Шифрование текстовых данных средствами Java

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

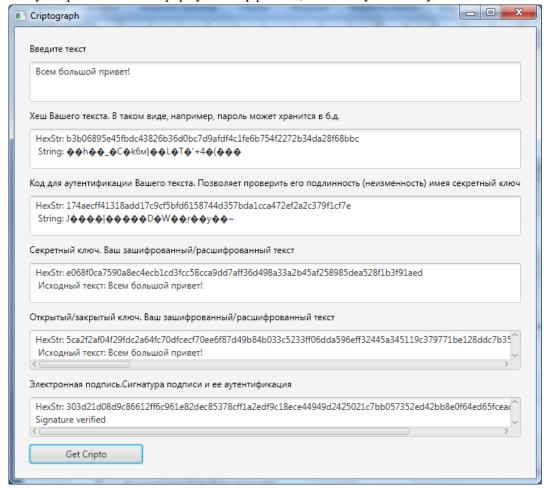


Рисунок 1

Разработка класса запуска проекта

Задание № 1

1. Запустите *NetBeans* и создайте новый проект. На первом шаге выберите Категорию *Maven*, Проекты – Приложение *Simple JavaFX Maven Archetype*.

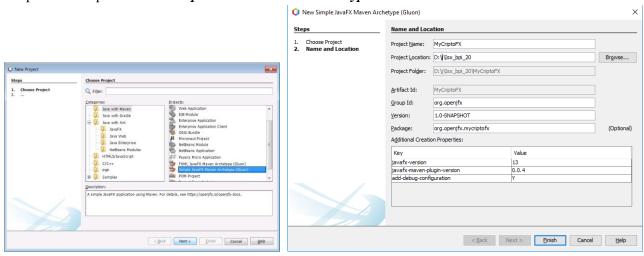


Рисунок 2

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

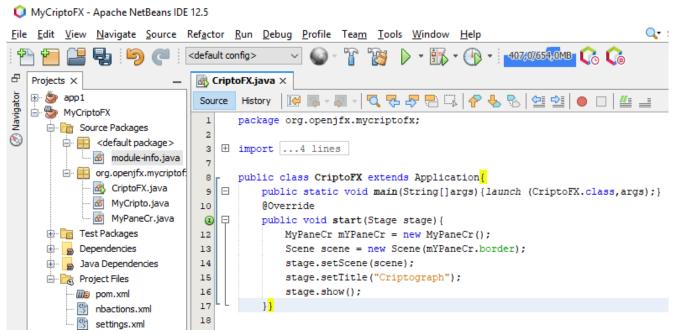


Рисунок 3

Разработка класса с криптографическими методами

Задание № 2

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

```
Source
       package org.openjfx.mycriptofx;

    import | ...13 lines

  2
 15
       public class MyCripto {
 16
    static String getHashString(String message) throws NoSuchAlgorithmException{
             MessageDigest md = MessageDigest.getInstance ("SHA-256");
 17
 18
             md.update(message.getBytes());
 19
             byte[] digest = md.digest();
             return getHexString(digest)+"\n String: " + new String(digest);
 20
 21
           static String getMacString(String message) throws Exception{
 22
 23
               KeyGenerator keyGen = KeyGenerator.getInstance("DES");
 24
               SecureRandom sr = new SecureRandom();
 25
               keyGen.init(sr);
               Key key = keyGen.generateKey();
 26
               Mac mac = Mac.getInstance("HmacSHA256");
 27
               mac.init(kev);
 28
               byte[]bytes = message.getBytes();
 29
               byte[]macResult = mac.doFinal(bytes);
 30
 31
               return getHexString(macResult) + "\n String: " + new String(macResult);
 32
           static String getCipherString(String message) throws Exception[
 33
    - 🖵
               KeyGenerator keyGen = KeyGenerator.getInstance("DES");
 34
               SecureRandom sr = new SecureRandom();
 35
               keyGen.init(sr);
 36
               Key key = keyGen.generateKey();
 37
 38
               Cipher cipher = Cipher.getInstance("DES/ECB/PKCS5Padding");
               cipher.init(cipher.ENCRYPT MODE, key);
```

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

Рисунок 4

Разработка классов реализации панелей для отображения данных

Задание № 3

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

```
  MyPaneCr.java 

  X

                          package org.openjfx.mycriptofx;
 2
   9
      public class MyPaneCr {
10
          BorderPane border;
11 👨
          MyPaneCr() {
12
              border = new BorderPane();
13
              border.setLeft(addVBox());
14
           }
15
          private VBox addVBox() {
16
              VBox vbox = new VBox();
              vbox.setPadding(new Insets(20));
17
18
              vbox.setSpacing(10);
19
              Text text = new Text ("Введите текст");
              Text text1 = new Text ("хеш вашего текста. В таком виде, например, пароль может хранить в б.д.");
20
21
              Text text2 = new Text ("код для аутентификации Вашего текста. Позволяет проверить его подлинность (нез
22
              Text text3 = new Text ("секретный ключ. Ваш зашифрованный/расшифрованный текст ");
              Text text4 = new Text ("открытый/закрытый ключ. Ваш зашифрованный/расшифрованный текст");
23
24
              Text text5 = new Text ("электронная подпись. Сигнатура подписи и ее аутунтификация");
25
              TextArea textInput = getTextArea(true);
              TextArea textOutputl = getTextArea(false);
26
27
              TextArea textOutput2 = getTextArea(false);
28
              TextArea textOutput3 = getTextArea(false);
              TextArea textOutput4 = getTextArea(false);
29
30
              TextArea textOutput5 = getTextArea(false);
31
              Button btn = new Button("Get Cripto");
              btn.setPrefSize(150.20):
32
33
              vbox.getChildren().addAll(text,textInput,text1,textOutput1,text2,textOutput2,
34
                  text3, textOutput3, text4, textOutput4, text5, textOutput5, btn);
              btn.setOnAction((ActionEvent event)-> {
35
                  if (!textInput.getText().equals("")){
36
37
                      textOutputl.setText(MyCripto.getHashString(textInput.getText()));
38
                      textOutput2.setText(MyCripto.getMacString(textInput.getText()));
39
40
                      textOutput3.setText(MyCripto.getCipherString(textInput.getText()));
                      textOutput4.setText(MyCripto.getPairCipherString(textInput.getText()));
41
                      textOutput5.setText(MyCripto.getElectSigmatString(textInput.getText()));
42
43
44
                   catch (Exception ex) {
                      // Logger.getLogger(MyPaneCr.class.getName()).log(Level.SEVERE,null,ex);
45
                      System.err.println(MyPaneCr.class.getName());
46
47
48
              11);
49
        return vbox:
50
    口
          private TextArea getTextArea(boolean isEditable) {
51
52
              TextArea textOutput = new TextArea();
              textOutput.setPrefColumnCount(50);
53
              textOutput.setPrefRowCount(2);
54
55
              textOutput.setEditable(isEditable);
56
              return textOutput;
57
58
```

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.

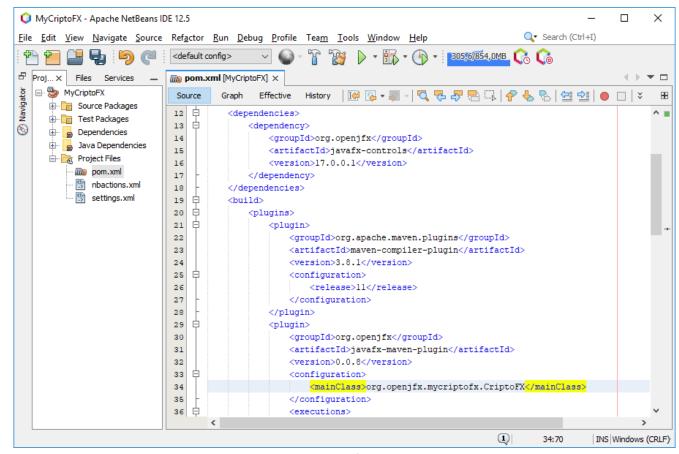


Рисунок 6

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