类 CPKTool

java.lang.Object
  └ **CPKTool**

public class **CPKTool**
extends java.lang.Object

构造方法摘要

CPKTool()

方法摘要

private int	**changeDecryptPassword**(java.lang.String oldPassword, java.lang.String newPassword) 更改用于加密的口令。
private int	**changeSignPassword**(java.lang.String oldPassword, java.lang.String newPassword) 更改用于签名的口令。
private java.lang.String	**decryptText**(java.lang.String ciphertext, java.lang.String password) 对密文进行解密
private int	**developeDecryptFile**(java.lang.String inFile, java.lang.String outFile, java.lang.String password) 对已加密的文件进行解密,解密可以有多个接收者
private java.lang.String	**encryptText**(java.lang.String plaintext, java.lang.String recipient) 对文本进行加密
private int	**envelopeEncryptFile**(java.lang.String inFile, java.lang.String outFile, java.lang.String[] recipients) 对文件进行加密,加密可以有多个接收者
private int	**formatPreserveDecryptFile**(java.lang.String inFile, java.lang.String outFile, java.lang.String password) 对输入文件进行保留格式解密，如果输入文件是经过保留格式加密得到的文件，然后保存到输出文件中。
private int	**formatPreserveEncryptFile**(java.lang.String inFile, java.lang.String outFile, java.lang.String[] recipients) 对输入文件进行保留格式加密，保存到输出文件中，可以有多个接收者

private int	**formatPreserveSignFile**(java.lang.String inFile, java.lang.String outFile, java.lang.String password) 对输入文件进行保留格式签名，保存到输出文件中
private int	**formatPreserveVerifyFile**(java.lang.String inFile) 验证由保留格式文件签名得到的签名文件
private java.lang.String	**getIdentity**() 获取用户名称，需要在之前设定用户名称。
private int	**importDecryptKey**(java.lang.String keyFile, java.lang.String password) 导入用于加密的私钥(.pem)，用户之前就已经获取。
private int	**importParameters**(java.lang.String paramFile) 导入公共参数文件
private int	**importSignKey**(java.lang.String keyFile, java.lang.String password) 导入用于签名的私钥(.pem)，用户之前就已经获取。
private int	**setIdentity**(java.lang.String identity) 设置用户名称
private java.lang.String	**signFile**(java.lang.String toBeSignedFile, java.lang.String password) 对文件进行签名
private java.lang.String	**signText**(java.lang.String toBeSignedMessage, java.lang.String password) 对待签名的文本进行签名
private int	**verifyFile**(java.lang.String file, java.lang.String signature, java.lang.String signer) 对由文件签名而得到的签名值进行验证
private int	**verifyText**(java.lang.String signedMessage, java.lang.String signature, java.lang.String signer) 对由文本签名而得到的签名值进行验证

从类 java.lang.Object 继承的方法
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

构造方法详细信息

CPKTool

public **CPKTool**()

方法详细信息

importParameters

```
private int importParameters(java.lang.String paramFile)
```

导入公共参数文件

参数:

`String` - paramFile 公共参数文件的路径

返回:

如果成功则返回0,失败返回 -1

---

setIdentity

```
private int setIdentity(java.lang.String identity)
```

设置用户名称

参数:

`String` - identity 用户身份标识

返回:

如果成功则返回0，失败返回 -1

---

getIdentity

```
private java.lang.String getIdentity()
```

获取用户名称，需要在之前设定用户名称。

返回:

如果成功则返回String类型的用户名称，失败返回 null

---

importSignKey

```
private int importSignKey(java.lang.String keyFile,
                          java.lang.String password)
```

导入用于签名的私钥(.pem)，用户之前就已经获取。

参数:

`String` - keyFile 私钥文件(.pem),
`String` - password 用户设定的口令

返回:

如果成功则返回 0,失败返回 -1

---

importDecryptKey

```
private int importDecryptKey(java.lang.String keyFile,
                             java.lang.String password)
```

导入用于加密的私钥(.pem)，用户之前就已经获取。

参数:

`String` - keyFile 私钥文件(.pem),
`String` - password 用户设定的口令

返回:

如果成功则返回 0,失败返回 -1

---

changeSignPassword

```
private int changeSignPassword(java.lang.String oldPassword,
                               java.lang.String newPassword)
```

更改用于签名的口令。

参数:

`String` - oldPassword 旧的口令
`String` - newPassword 新的口令

返回:

如果成功则返回 0,失败返回 -1

---

changeDecryptPassword

```
private int changeDecryptPassword(java.lang.String oldPassword,
                                  java.lang.String newPassword)
```

更改用于加密的口令。

参数:

`String` - oldPassword 旧的口令
`String` - newPassword 新的口令

返回:

如果成功则返回 0,失败返回 -1

---

signText

```
private java.lang.String signText(java.lang.String toBeSignedMessage,
                                  java.lang.String password)
```

对待签名的文本进行签名

参数:

`String` - toBeSignedMessage 待签名的文本原文
`String` - password 口令

返回:

如果成功则返回 签名值,失败返回 "Failed!"

---

verifyText

```
private int verifyText(java.lang.String signedMessage,
                       java.lang.String signature,
                       java.lang.String signer)
```

对由文本签名而得到的签名值进行验证

参数:

`String` - signedMessage 文本原文
`String` - signature 签名值，通过签名而得到的一组字符串
`String` - signer 签名者，即用户名称

返回:

如果正确则返回 0,失败返回 -1

---

signFile

```
private java.lang.String signFile(java.lang.String toBeSignedFile,
                                  java.lang.String password)
```

对文件进行签名

参数:

`String` - toBeSignedFile 待签名的文件
`String` - password 口令

返回:

如果正确则返回 0,失败返回 -1

---

verifyFile

```
private int verifyFile(java.lang.String file,
                       java.lang.String signature,
                       java.lang.String signer)
```

对由文件签名而得到的签名值进行验证

参数:

`String` - file 待验证的文件
`String` - signature 签名值，通过签名而得到的一组字符串
`String` - signer 签名者，即用户名称

返回:

如果正确则返回 0,失败返回 -1

---

encryptText

```
private java.lang.String encryptText(java.lang.String plaintext,
                                     java.lang.String recipient)
```

对文本进行加密

参数:

`String` - plaintext 明文
`String` - recipient 接收者

返回:

如果正确则返回 加密密文,失败返回 null

---

decryptText

```
private java.lang.String decryptText(java.lang.String ciphertext,
                                     java.lang.String password)
```

对密文进行解密

参数:

`String` - ciphertext 密文
`String` - password 口令

返回:

如果正确则返回解密得到的明文,失败返回 null

---

envelopeEncryptFile

```
private int envelopeEncryptFile(java.lang.String inFile,
                                java.lang.String outFile,
                                java.lang.String[] recipients)
```

对文件进行加密,加密可以有多个接收者

参数:

`String` - inFile 输入文件
`String` - outFile 输出文件
`String[]` - recipients 接受者们

返回:

如果正确则返回 0,失败返回 -1

---

developeDecryptFile

```
private int developeDecryptFile(java.lang.String inFile,
                                java.lang.String outFile,
                                java.lang.String password)
```

对已加密的文件进行解密,解密可以有多个接收者

参数:

`String` - inFile 输入文件
`String` - outFile 输出文件
`String[]` - recipients 接受者们

返回:

如果正确则返回 0,失败返回 -1

---

formatPreserveSignFile

```
private int formatPreserveSignFile(java.lang.String inFile,
                                   java.lang.String outFile,
                                   java.lang.String password)
```

对输入文件进行保留格式签名，保存到输出文件中

参数:

`String` - inFile 输入文件
`String` - outFile 输出文件
`String` - password 口令

返回:

如果正确则返回 0,失败返回 -1

---

formatPreserveVerifyFile

```
private int formatPreserveVerifyFile(java.lang.String inFile)
```

验证由保留格式文件签名得到的签名文件

参数:

`String` - inFile 输入文件

返回:

如果正确则返回 0,失败返回 -1

---

formatPreserveEncryptFile

```
private int formatPreserveEncryptFile(java.lang.String inFile,
                                      java.lang.String outFile,
                                      java.lang.String[] recipients)
```

对输入文件进行保留格式加密，保存到输出文件中，可以有多个接收者

参数:

`String` - inFile 输入文件
`String` - outFile 输出文件
`String[]` - recipients 接收者们

返回:

如果正确则返回 0,失败返回 -1

---

formatPreserveDecryptFile

```
private int formatPreserveDecryptFile(java.lang.String inFile,
                                      java.lang.String outFile,
                                      java.lang.String password)
```

对输入文件进行保留格式解密，如果输入文件是经过保留格式加密得到的文件，然后保存到输出文件中。

参数:

`String` - inFile 输入文件
`String` - outFile 输出文件
`String` - password 口令

返回:

如果正确则返回 0,失败返回 -1
