| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | [**Class**](http://docs.google.com/java/security/InvalidKeyException.html) | **Use** | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| PREV   NEXT | [**FRAMES**](http://docs.google.com/index.html?java/security//class-useInvalidKeyException.html)    [**NO FRAMES**](http://docs.google.com/InvalidKeyException.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |

**Uses of Class**

**java.security.InvalidKeyException**

| Packages that use [InvalidKeyException](http://docs.google.com/java/security/InvalidKeyException.html) | |
| --- | --- |
| [**java.security**](#3znysh7) | Provides the classes and interfaces for the security framework. |
| [**java.security.cert**](#2et92p0) | Provides classes and interfaces for parsing and managing certificates, certificate revocation lists (CRLs), and certification paths. |
| [**javax.crypto**](#tyjcwt) | Provides the classes and interfaces for cryptographic operations. |
| [**javax.crypto.spec**](#3dy6vkm) | Provides classes and interfaces for key specifications and algorithm parameter specifications. |
| [**javax.security.cert**](#1t3h5sf) | Provides classes for public key certificates. |

| Uses of [InvalidKeyException](http://docs.google.com/java/security/InvalidKeyException.html) in [java.security](http://docs.google.com/java/security/package-summary.html) | |
| --- | --- |

| Methods in [java.security](http://docs.google.com/java/security/package-summary.html) that throw [InvalidKeyException](http://docs.google.com/java/security/InvalidKeyException.html) | |
| --- | --- |
| protected abstract  void | **SignatureSpi.**[**engineInitSign**](http://docs.google.com/java/security/SignatureSpi.html#engineInitSign(java.security.PrivateKey))([PrivateKey](http://docs.google.com/java/security/PrivateKey.html) privateKey)            Initializes this signature object with the specified private key for signing operations. |
| protected  void | **SignatureSpi.**[**engineInitSign**](http://docs.google.com/java/security/SignatureSpi.html#engineInitSign(java.security.PrivateKey,%20java.security.SecureRandom))([PrivateKey](http://docs.google.com/java/security/PrivateKey.html) privateKey, [SecureRandom](http://docs.google.com/java/security/SecureRandom.html) random)            Initializes this signature object with the specified private key and source of randomness for signing operations. |
| protected abstract  void | **SignatureSpi.**[**engineInitVerify**](http://docs.google.com/java/security/SignatureSpi.html#engineInitVerify(java.security.PublicKey))([PublicKey](http://docs.google.com/java/security/PublicKey.html) publicKey)            Initializes this signature object with the specified public key for verification operations. |
| protected abstract  [Key](http://docs.google.com/java/security/Key.html) | **KeyFactorySpi.**[**engineTranslateKey**](http://docs.google.com/java/security/KeyFactorySpi.html#engineTranslateKey(java.security.Key))([Key](http://docs.google.com/java/security/Key.html) key)            Translates a key object, whose provider may be unknown or potentially untrusted, into a corresponding key object of this key factory. |
| void | **Signature.**[**initSign**](http://docs.google.com/java/security/Signature.html#initSign(java.security.PrivateKey))([PrivateKey](http://docs.google.com/java/security/PrivateKey.html) privateKey)            Initialize this object for signing. |
| void | **Signature.**[**initSign**](http://docs.google.com/java/security/Signature.html#initSign(java.security.PrivateKey,%20java.security.SecureRandom))([PrivateKey](http://docs.google.com/java/security/PrivateKey.html) privateKey, [SecureRandom](http://docs.google.com/java/security/SecureRandom.html) random)            Initialize this object for signing. |
| void | **Signature.**[**initVerify**](http://docs.google.com/java/security/Signature.html#initVerify(java.security.cert.Certificate))([Certificate](http://docs.google.com/java/security/cert/Certificate.html) certificate)            Initializes this object for verification, using the public key from the given certificate. |
| void | **Signature.**[**initVerify**](http://docs.google.com/java/security/Signature.html#initVerify(java.security.PublicKey))([PublicKey](http://docs.google.com/java/security/PublicKey.html) publicKey)            Initializes this object for verification. |
| [Key](http://docs.google.com/java/security/Key.html) | **KeyFactory.**[**translateKey**](http://docs.google.com/java/security/KeyFactory.html#translateKey(java.security.Key))([Key](http://docs.google.com/java/security/Key.html) key)            Translates a key object, whose provider may be unknown or potentially untrusted, into a corresponding key object of this key factory. |
| boolean | **SignedObject.**[**verify**](http://docs.google.com/java/security/SignedObject.html#verify(java.security.PublicKey,%20java.security.Signature))([PublicKey](http://docs.google.com/java/security/PublicKey.html) verificationKey, [Signature](http://docs.google.com/java/security/Signature.html) verificationEngine)            Verifies that the signature in this SignedObject is the valid signature for the object stored inside, with the given verification key, using the designated verification engine. |

| Constructors in [java.security](http://docs.google.com/java/security/package-summary.html) that throw [InvalidKeyException](http://docs.google.com/java/security/InvalidKeyException.html) | |
| --- | --- |
| [**SignedObject**](http://docs.google.com/java/security/SignedObject.html#SignedObject(java.io.Serializable,%20java.security.PrivateKey,%20java.security.Signature))([Serializable](http://docs.google.com/java/io/Serializable.html) object, [PrivateKey](http://docs.google.com/java/security/PrivateKey.html) signingKey, [Signature](http://docs.google.com/java/security/Signature.html) signingEngine)            Constructs a SignedObject from any Serializable object. |

| Uses of [InvalidKeyException](http://docs.google.com/java/security/InvalidKeyException.html) in [java.security.cert](http://docs.google.com/java/security/cert/package-summary.html) | |
| --- | --- |

| Methods in [java.security.cert](http://docs.google.com/java/security/cert/package-summary.html) that throw [InvalidKeyException](http://docs.google.com/java/security/InvalidKeyException.html) | |
| --- | --- |
| abstract  void | **X509CRL.**[**verify**](http://docs.google.com/java/security/cert/X509CRL.html#verify(java.security.PublicKey))([PublicKey](http://docs.google.com/java/security/PublicKey.html) key)            Verifies that this CRL was signed using the private key that corresponds to the given public key. |
| abstract  void | **Certificate.**[**verify**](http://docs.google.com/java/security/cert/Certificate.html#verify(java.security.PublicKey))([PublicKey](http://docs.google.com/java/security/PublicKey.html) key)            Verifies that this certificate was signed using the private key that corresponds to the specified public key. |
| abstract  void | **X509CRL.**[**verify**](http://docs.google.com/java/security/cert/X509CRL.html#verify(java.security.PublicKey,%20java.lang.String))([PublicKey](http://docs.google.com/java/security/PublicKey.html) key, [String](http://docs.google.com/java/lang/String.html) sigProvider)            Verifies that this CRL was signed using the private key that corresponds to the given public key. |
| abstract  void | **Certificate.**[**verify**](http://docs.google.com/java/security/cert/Certificate.html#verify(java.security.PublicKey,%20java.lang.String))([PublicKey](http://docs.google.com/java/security/PublicKey.html) key, [String](http://docs.google.com/java/lang/String.html) sigProvider)            Verifies that this certificate was signed using the private key that corresponds to the specified public key. |

| Uses of [InvalidKeyException](http://docs.google.com/java/security/InvalidKeyException.html) in [javax.crypto](http://docs.google.com/javax/crypto/package-summary.html) | |
| --- | --- |

| Methods in [javax.crypto](http://docs.google.com/javax/crypto/package-summary.html) that throw [InvalidKeyException](http://docs.google.com/java/security/InvalidKeyException.html) | |
| --- | --- |
| [Key](http://docs.google.com/java/security/Key.html) | **KeyAgreement.**[**doPhase**](http://docs.google.com/javax/crypto/KeyAgreement.html#doPhase(java.security.Key,%20boolean))([Key](http://docs.google.com/java/security/Key.html) key, boolean lastPhase)            Executes the next phase of this key agreement with the given key that was received from one of the other parties involved in this key agreement. |
| protected abstract  [Key](http://docs.google.com/java/security/Key.html) | **KeyAgreementSpi.**[**engineDoPhase**](http://docs.google.com/javax/crypto/KeyAgreementSpi.html#engineDoPhase(java.security.Key,%20boolean))([Key](http://docs.google.com/java/security/Key.html) key, boolean lastPhase)            Executes the next phase of this key agreement with the given key that was received from one of the other parties involved in this key agreement. |
| protected abstract  [SecretKey](http://docs.google.com/javax/crypto/SecretKey.html) | **KeyAgreementSpi.**[**engineGenerateSecret**](http://docs.google.com/javax/crypto/KeyAgreementSpi.html#engineGenerateSecret(java.lang.String))([String](http://docs.google.com/java/lang/String.html) algorithm)            Creates the shared secret and returns it as a secret key object of the requested algorithm type. |
| protected  int | **CipherSpi.**[**engineGetKeySize**](http://docs.google.com/javax/crypto/CipherSpi.html#engineGetKeySize(java.security.Key))([Key](http://docs.google.com/java/security/Key.html) key)            Returns the key size of the given key object in bits. |
| protected abstract  void | **CipherSpi.**[**engineInit**](http://docs.google.com/javax/crypto/CipherSpi.html#engineInit(int,%20java.security.Key,%20java.security.spec.AlgorithmParameterSpec,%20java.security.SecureRandom))(int opmode, [Key](http://docs.google.com/java/security/Key.html) key, [AlgorithmParameterSpec](http://docs.google.com/java/security/spec/AlgorithmParameterSpec.html) params, [SecureRandom](http://docs.google.com/java/security/SecureRandom.html) random)            Initializes this cipher with a key, a set of algorithm parameters, and a source of randomness. |
| protected abstract  void | **CipherSpi.**[**engineInit**](http://docs.google.com/javax/crypto/CipherSpi.html#engineInit(int,%20java.security.Key,%20java.security.AlgorithmParameters,%20java.security.SecureRandom))(int opmode, [Key](http://docs.google.com/java/security/Key.html) key, [AlgorithmParameters](http://docs.google.com/java/security/AlgorithmParameters.html) params, [SecureRandom](http://docs.google.com/java/security/SecureRandom.html) random)            Initializes this cipher with a key, a set of algorithm parameters, and a source of randomness. |
| protected abstract  void | **CipherSpi.**[**engineInit**](http://docs.google.com/javax/crypto/CipherSpi.html#engineInit(int,%20java.security.Key,%20java.security.SecureRandom))(int opmode, [Key](http://docs.google.com/java/security/Key.html) key, [SecureRandom](http://docs.google.com/java/security/SecureRandom.html) random)            Initializes this cipher with a key and a source of randomness. |
| protected abstract  void | **ExemptionMechanismSpi.**[**engineInit**](http://docs.google.com/javax/crypto/ExemptionMechanismSpi.html#engineInit(java.security.Key))([Key](http://docs.google.com/java/security/Key.html) key)            Initializes this exemption mechanism with a key. |
| protected abstract  void | **ExemptionMechanismSpi.**[**engineInit**](http://docs.google.com/javax/crypto/ExemptionMechanismSpi.html#engineInit(java.security.Key,%20java.security.AlgorithmParameters))([Key](http://docs.google.com/java/security/Key.html) key, [AlgorithmParameters](http://docs.google.com/java/security/AlgorithmParameters.html) params)            Initializes this exemption mechanism with a key and a set of algorithm parameters. |
| protected abstract  void | **MacSpi.**[**engineInit**](http://docs.google.com/javax/crypto/MacSpi.html#engineInit(java.security.Key,%20java.security.spec.AlgorithmParameterSpec))([Key](http://docs.google.com/java/security/Key.html) key, [AlgorithmParameterSpec](http://docs.google.com/java/security/spec/AlgorithmParameterSpec.html) params)            Initializes the MAC with the given (secret) key and algorithm parameters. |
| protected abstract  void | **ExemptionMechanismSpi.**[**engineInit**](http://docs.google.com/javax/crypto/ExemptionMechanismSpi.html#engineInit(java.security.Key,%20java.security.spec.AlgorithmParameterSpec))([Key](http://docs.google.com/java/security/Key.html) key, [AlgorithmParameterSpec](http://docs.google.com/java/security/spec/AlgorithmParameterSpec.html) params)            Initializes this exemption mechanism with a key and a set of algorithm parameters. |
| protected abstract  void | **KeyAgreementSpi.**[**engineInit**](http://docs.google.com/javax/crypto/KeyAgreementSpi.html#engineInit(java.security.Key,%20java.security.spec.AlgorithmParameterSpec,%20java.security.SecureRandom))([Key](http://docs.google.com/java/security/Key.html) key, [AlgorithmParameterSpec](http://docs.google.com/java/security/spec/AlgorithmParameterSpec.html) params, [SecureRandom](http://docs.google.com/java/security/SecureRandom.html) random)            Initializes this key agreement with the given key, set of algorithm parameters, and source of randomness. |
| protected abstract  void | **KeyAgreementSpi.**[**engineInit**](http://docs.google.com/javax/crypto/KeyAgreementSpi.html#engineInit(java.security.Key,%20java.security.SecureRandom))([Key](http://docs.google.com/java/security/Key.html) key, [SecureRandom](http://docs.google.com/java/security/SecureRandom.html) random)            Initializes this key agreement with the given key and source of randomness. |
| protected abstract  [SecretKey](http://docs.google.com/javax/crypto/SecretKey.html) | **SecretKeyFactorySpi.**[**engineTranslateKey**](http://docs.google.com/javax/crypto/SecretKeyFactorySpi.html#engineTranslateKey(javax.crypto.SecretKey))([SecretKey](http://docs.google.com/javax/crypto/SecretKey.html) key)            Translates a key object, whose provider may be unknown or potentially untrusted, into a corresponding key object of this secret-key factory. |
| protected  [Key](http://docs.google.com/java/security/Key.html) | **CipherSpi.**[**engineUnwrap**](http://docs.google.com/javax/crypto/CipherSpi.html#engineUnwrap(byte%5B%5D,%20java.lang.String,%20int))(byte[] wrappedKey, [String](http://docs.google.com/java/lang/String.html) wrappedKeyAlgorithm, int wrappedKeyType)            Unwrap a previously wrapped key. |
| protected  byte[] | **CipherSpi.**[**engineWrap**](http://docs.google.com/javax/crypto/CipherSpi.html#engineWrap(java.security.Key))([Key](http://docs.google.com/java/security/Key.html) key)            Wrap a key. |
| [SecretKey](http://docs.google.com/javax/crypto/SecretKey.html) | **KeyAgreement.**[**generateSecret**](http://docs.google.com/javax/crypto/KeyAgreement.html#generateSecret(java.lang.String))([String](http://docs.google.com/java/lang/String.html) algorithm)            Creates the shared secret and returns it as a SecretKey object of the specified algorithm. |
| [PKCS8EncodedKeySpec](http://docs.google.com/java/security/spec/PKCS8EncodedKeySpec.html) | **EncryptedPrivateKeyInfo.**[**getKeySpec**](http://docs.google.com/javax/crypto/EncryptedPrivateKeyInfo.html#getKeySpec(java.security.Key))([Key](http://docs.google.com/java/security/Key.html) decryptKey)            Extract the enclosed PKCS8EncodedKeySpec object from the encrypted data and return it. |
| [PKCS8EncodedKeySpec](http://docs.google.com/java/security/spec/PKCS8EncodedKeySpec.html) | **EncryptedPrivateKeyInfo.**[**getKeySpec**](http://docs.google.com/javax/crypto/EncryptedPrivateKeyInfo.html#getKeySpec(java.security.Key,%20java.security.Provider))([Key](http://docs.google.com/java/security/Key.html) decryptKey, [Provider](http://docs.google.com/java/security/Provider.html) provider)            Extract the enclosed PKCS8EncodedKeySpec object from the encrypted data and return it. |
| [PKCS8EncodedKeySpec](http://docs.google.com/java/security/spec/PKCS8EncodedKeySpec.html) | **EncryptedPrivateKeyInfo.**[**getKeySpec**](http://docs.google.com/javax/crypto/EncryptedPrivateKeyInfo.html#getKeySpec(java.security.Key,%20java.lang.String))([Key](http://docs.google.com/java/security/Key.html) decryptKey, [String](http://docs.google.com/java/lang/String.html) providerName)            Extract the enclosed PKCS8EncodedKeySpec object from the encrypted data and return it. |
| [Object](http://docs.google.com/java/lang/Object.html) | **SealedObject.**[**getObject**](http://docs.google.com/javax/crypto/SealedObject.html#getObject(java.security.Key))([Key](http://docs.google.com/java/security/Key.html) key)            Retrieves the original (encapsulated) object. |
| [Object](http://docs.google.com/java/lang/Object.html) | **SealedObject.**[**getObject**](http://docs.google.com/javax/crypto/SealedObject.html#getObject(java.security.Key,%20java.lang.String))([Key](http://docs.google.com/java/security/Key.html) key, [String](http://docs.google.com/java/lang/String.html) provider)            Retrieves the original (encapsulated) object. |
| void | **Cipher.**[**init**](http://docs.google.com/javax/crypto/Cipher.html#init(int,%20java.security.cert.Certificate))(int opmode, [Certificate](http://docs.google.com/java/security/cert/Certificate.html) certificate)            Initializes this cipher with the public key from the given certificate. |
| void | **Cipher.**[**init**](http://docs.google.com/javax/crypto/Cipher.html#init(int,%20java.security.cert.Certificate,%20java.security.SecureRandom))(int opmode, [Certificate](http://docs.google.com/java/security/cert/Certificate.html) certificate, [SecureRandom](http://docs.google.com/java/security/SecureRandom.html) random)            Initializes this cipher with the public key from the given certificate and a source of randomness. |
| void | **Cipher.**[**init**](http://docs.google.com/javax/crypto/Cipher.html#init(int,%20java.security.Key))(int opmode, [Key](http://docs.google.com/java/security/Key.html) key)            Initializes this cipher with a key. |
| void | **Cipher.**[**init**](http://docs.google.com/javax/crypto/Cipher.html#init(int,%20java.security.Key,%20java.security.AlgorithmParameters))(int opmode, [Key](http://docs.google.com/java/security/Key.html) key, [AlgorithmParameters](http://docs.google.com/java/security/AlgorithmParameters.html) params)            Initializes this cipher with a key and a set of algorithm parameters. |
| void | **Cipher.**[**init**](http://docs.google.com/javax/crypto/Cipher.html#init(int,%20java.security.Key,%20java.security.spec.AlgorithmParameterSpec))(int opmode, [Key](http://docs.google.com/java/security/Key.html) key, [AlgorithmParameterSpec](http://docs.google.com/java/security/spec/AlgorithmParameterSpec.html) params)            Initializes this cipher with a key and a set of algorithm parameters. |
| void | **Cipher.**[**init**](http://docs.google.com/javax/crypto/Cipher.html#init(int,%20java.security.Key,%20java.security.spec.AlgorithmParameterSpec,%20java.security.SecureRandom))(int opmode, [Key](http://docs.google.com/java/security/Key.html) key, [AlgorithmParameterSpec](http://docs.google.com/java/security/spec/AlgorithmParameterSpec.html) params, [SecureRandom](http://docs.google.com/java/security/SecureRandom.html) random)            Initializes this cipher with a key, a set of algorithm parameters, and a source of randomness. |
| void | **Cipher.**[**init**](http://docs.google.com/javax/crypto/Cipher.html#init(int,%20java.security.Key,%20java.security.AlgorithmParameters,%20java.security.SecureRandom))(int opmode, [Key](http://docs.google.com/java/security/Key.html) key, [AlgorithmParameters](http://docs.google.com/java/security/AlgorithmParameters.html) params, [SecureRandom](http://docs.google.com/java/security/SecureRandom.html) random)            Initializes this cipher with a key, a set of algorithm parameters, and a source of randomness. |
| void | **Cipher.**[**init**](http://docs.google.com/javax/crypto/Cipher.html#init(int,%20java.security.Key,%20java.security.SecureRandom))(int opmode, [Key](http://docs.google.com/java/security/Key.html) key, [SecureRandom](http://docs.google.com/java/security/SecureRandom.html) random)            Initializes this cipher with a key and a source of randomness. |
| void | **ExemptionMechanism.**[**init**](http://docs.google.com/javax/crypto/ExemptionMechanism.html#init(java.security.Key))([Key](http://docs.google.com/java/security/Key.html) key)            Initializes this exemption mechanism with a key. |
| void | **KeyAgreement.**[**init**](http://docs.google.com/javax/crypto/KeyAgreement.html#init(java.security.Key))([Key](http://docs.google.com/java/security/Key.html) key)            Initializes this key agreement with the given key, which is required to contain all the algorithm parameters required for this key agreement. |
| void | **Mac.**[**init**](http://docs.google.com/javax/crypto/Mac.html#init(java.security.Key))([Key](http://docs.google.com/java/security/Key.html) key)            Initializes this Mac object with the given key. |
| void | **ExemptionMechanism.**[**init**](http://docs.google.com/javax/crypto/ExemptionMechanism.html#init(java.security.Key,%20java.security.AlgorithmParameters))([Key](http://docs.google.com/java/security/Key.html) key, [AlgorithmParameters](http://docs.google.com/java/security/AlgorithmParameters.html) params)            Initializes this exemption mechanism with a key and a set of algorithm parameters. |
| void | **ExemptionMechanism.**[**init**](http://docs.google.com/javax/crypto/ExemptionMechanism.html#init(java.security.Key,%20java.security.spec.AlgorithmParameterSpec))([Key](http://docs.google.com/java/security/Key.html) key, [AlgorithmParameterSpec](http://docs.google.com/java/security/spec/AlgorithmParameterSpec.html) params)            Initializes this exemption mechanism with a key and a set of algorithm parameters. |
| void | **KeyAgreement.**[**init**](http://docs.google.com/javax/crypto/KeyAgreement.html#init(java.security.Key,%20java.security.spec.AlgorithmParameterSpec))([Key](http://docs.google.com/java/security/Key.html) key, [AlgorithmParameterSpec](http://docs.google.com/java/security/spec/AlgorithmParameterSpec.html) params)            Initializes this key agreement with the given key and set of algorithm parameters. |
| void | **Mac.**[**init**](http://docs.google.com/javax/crypto/Mac.html#init(java.security.Key,%20java.security.spec.AlgorithmParameterSpec))([Key](http://docs.google.com/java/security/Key.html) key, [AlgorithmParameterSpec](http://docs.google.com/java/security/spec/AlgorithmParameterSpec.html) params)            Initializes this Mac object with the given key and algorithm parameters. |
| void | **KeyAgreement.**[**init**](http://docs.google.com/javax/crypto/KeyAgreement.html#init(java.security.Key,%20java.security.spec.AlgorithmParameterSpec,%20java.security.SecureRandom))([Key](http://docs.google.com/java/security/Key.html) key, [AlgorithmParameterSpec](http://docs.google.com/java/security/spec/AlgorithmParameterSpec.html) params, [SecureRandom](http://docs.google.com/java/security/SecureRandom.html) random)            Initializes this key agreement with the given key, set of algorithm parameters, and source of randomness. |
| void | **KeyAgreement.**[**init**](http://docs.google.com/javax/crypto/KeyAgreement.html#init(java.security.Key,%20java.security.SecureRandom))([Key](http://docs.google.com/java/security/Key.html) key, [SecureRandom](http://docs.google.com/java/security/SecureRandom.html) random)            Initializes this key agreement with the given key and source of randomness. |
| [SecretKey](http://docs.google.com/javax/crypto/SecretKey.html) | **SecretKeyFactory.**[**translateKey**](http://docs.google.com/javax/crypto/SecretKeyFactory.html#translateKey(javax.crypto.SecretKey))([SecretKey](http://docs.google.com/javax/crypto/SecretKey.html) key)            Translates a key object, whose provider may be unknown or potentially untrusted, into a corresponding key object of this secret-key factory. |
| [Key](http://docs.google.com/java/security/Key.html) | **Cipher.**[**unwrap**](http://docs.google.com/javax/crypto/Cipher.html#unwrap(byte%5B%5D,%20java.lang.String,%20int))(byte[] wrappedKey, [String](http://docs.google.com/java/lang/String.html) wrappedKeyAlgorithm, int wrappedKeyType)            Unwrap a previously wrapped key. |
| byte[] | **Cipher.**[**wrap**](http://docs.google.com/javax/crypto/Cipher.html#wrap(java.security.Key))([Key](http://docs.google.com/java/security/Key.html) key)            Wrap a key. |

| Uses of [InvalidKeyException](http://docs.google.com/java/security/InvalidKeyException.html) in [javax.crypto.spec](http://docs.google.com/javax/crypto/spec/package-summary.html) | |
| --- | --- |

| Methods in [javax.crypto.spec](http://docs.google.com/javax/crypto/spec/package-summary.html) that throw [InvalidKeyException](http://docs.google.com/java/security/InvalidKeyException.html) | |
| --- | --- |
| static boolean | **DESKeySpec.**[**isParityAdjusted**](http://docs.google.com/javax/crypto/spec/DESKeySpec.html#isParityAdjusted(byte%5B%5D,%20int))(byte[] key, int offset)            Checks if the given DES key material, starting at offset inclusive, is parity-adjusted. |
| static boolean | **DESedeKeySpec.**[**isParityAdjusted**](http://docs.google.com/javax/crypto/spec/DESedeKeySpec.html#isParityAdjusted(byte%5B%5D,%20int))(byte[] key, int offset)            Checks if the given DES-EDE key, starting at offset inclusive, is parity-adjusted. |
| static boolean | **DESKeySpec.**[**isWeak**](http://docs.google.com/javax/crypto/spec/DESKeySpec.html#isWeak(byte%5B%5D,%20int))(byte[] key, int offset)            Checks if the given DES key material is weak or semi-weak. |

| Constructors in [javax.crypto.spec](http://docs.google.com/javax/crypto/spec/package-summary.html) that throw [InvalidKeyException](http://docs.google.com/java/security/InvalidKeyException.html) | |
| --- | --- |
| [**DESedeKeySpec**](http://docs.google.com/javax/crypto/spec/DESedeKeySpec.html#DESedeKeySpec(byte%5B%5D))(byte[] key)            Creates a DESedeKeySpec object using the first 24 bytes in key as the key material for the DES-EDE key. |
| [**DESedeKeySpec**](http://docs.google.com/javax/crypto/spec/DESedeKeySpec.html#DESedeKeySpec(byte%5B%5D,%20int))(byte[] key, int offset)            Creates a DESedeKeySpec object using the first 24 bytes in key, beginning at offset inclusive, as the key material for the DES-EDE key. |
| [**DESKeySpec**](http://docs.google.com/javax/crypto/spec/DESKeySpec.html#DESKeySpec(byte%5B%5D))(byte[] key)            Creates a DESKeySpec object using the first 8 bytes in key as the key material for the DES key. |
| [**DESKeySpec**](http://docs.google.com/javax/crypto/spec/DESKeySpec.html#DESKeySpec(byte%5B%5D,%20int))(byte[] key, int offset)            Creates a DESKeySpec object using the first 8 bytes in key, beginning at offset inclusive, as the key material for the DES key. |

| Uses of [InvalidKeyException](http://docs.google.com/java/security/InvalidKeyException.html) in [javax.security.cert](http://docs.google.com/javax/security/cert/package-summary.html) | |
| --- | --- |

| Methods in [javax.security.cert](http://docs.google.com/javax/security/cert/package-summary.html) that throw [InvalidKeyException](http://docs.google.com/java/security/InvalidKeyException.html) | |
| --- | --- |
| abstract  void | **Certificate.**[**verify**](http://docs.google.com/javax/security/cert/Certificate.html#verify(java.security.PublicKey))([PublicKey](http://docs.google.com/java/security/PublicKey.html) key)            Verifies that this certificate was signed using the private key that corresponds to the specified public key. |
| abstract  void | **Certificate.**[**verify**](http://docs.google.com/javax/security/cert/Certificate.html#verify(java.security.PublicKey,%20java.lang.String))([PublicKey](http://docs.google.com/java/security/PublicKey.html) key, [String](http://docs.google.com/java/lang/String.html) sigProvider)            Verifies that this certificate was signed using the private key that corresponds to the specified public key. |

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | [**Class**](http://docs.google.com/java/security/InvalidKeyException.html) | **Use** | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| PREV   NEXT | [**FRAMES**](http://docs.google.com/index.html?java/security//class-useInvalidKeyException.html)    [**NO FRAMES**](http://docs.google.com/InvalidKeyException.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |

[Submit a bug or feature](http://bugs.sun.com/services/bugreport/index.jsp)

For further API reference and developer documentation, see [Java SE Developer Documentation](http://docs.google.com/webnotes/devdocs-vs-specs.html). That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.

Copyright 2006 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](http://docs.google.com/legal/license.html). Also see the [documentation redistribution policy](http://java.sun.com/docs/redist.html).