Module java.base
Package java.security.interfaces

Interface DSAKeyPairGenerator

`public interface DSAKeyPairGenerator`

An interface to an object capable of generating DSA key pairs.

The `initialize` methods may each be called any number of times. If no `initialize` method is called on a DSAKeyPairGenerator, each provider that implements this interface should supply (and document) a default initialization. Note that defaults may vary across different providers. Additionally, the default value for a provider may change in a future version. Therefore, it is recommended to explicitly initialize the DSAKeyPairGenerator instead of relying on provider-specific defaults.

Users wishing to indicate DSA-specific parameters, and to generate a key pair suitable for use with the DSA algorithm typically

1. Get a key pair generator for the DSA algorithm by calling the KeyPairGenerator `getInstance` method with "DSA" as its argument.
2. Check if the returned key pair generator is an instance of DSAKeyPairGenerator before casting the result to a DSAKeyPairGenerator and calling one of the `initialize` methods from this DSAKeyPairGenerator interface.
3. Generate a key pair by calling the `generateKeyPair` method of the KeyPairGenerator class.

Note: it is not always necessary to do algorithm-specific initialization for a DSA key pair generator. That is, it is not always necessary to call an `initialize` method in this interface. Algorithm-independent initialization using the `initialize` method in the KeyPairGenerator interface is all that is needed when you accept defaults for algorithm-specific parameters.

Note: Some earlier implementations of this interface may not support larger values of DSA parameters such as 3072-bit.

Since:

1.1

See Also:

`KeyPairGenerator`

Method Summary

All Methods | Instance Methods | Abstract Methods

Modifier and Type	Method	Description
void	**initialize**(int modlen, boolean genParams, **SecureRandom** random)	Initializes the key pair generator for a given modulus length (instead of parameters), and an optional SecureRandom bit source.
void	**initialize**(**DSAParams** params, **SecureRandom** random)	Initializes the key pair generator using the DSA family parameters (p,q and g) and an optional SecureRandom bit source.

Method Details

initialize

```
void initialize(DSAParams params,
                SecureRandom random)
```

Initializes the key pair generator using the DSA family parameters (p,q and g) and an optional SecureRandom bit source. If a SecureRandom bit source is needed but not supplied, i.e. null, a default SecureRandom instance will be used.

Parameters:

`params` - the parameters to use to generate the keys.

`random` - the random bit source to use to generate key bits; can be null.

Throws:

`InvalidParameterException` - if the `params` value is invalid, null, or unsupported.

initialize

```
void initialize(int modlen,
                boolean genParams,
                SecureRandom random)
```

Initializes the key pair generator for a given modulus length (instead of parameters), and an optional SecureRandom bit source. If a SecureRandom bit source is needed but not supplied, i.e. null, a default SecureRandom instance will be used.

If `genParams` is true, this method generates new p, q and g parameters. If it is false, the method uses precomputed parameters for the modulus length requested. If there are no precomputed parameters for that modulus length, an exception will be thrown.

Parameters:

`modlen` - the modulus length in bits. Valid values are any multiple of 64 between 512 and 1024, inclusive, 2048, and 3072.

`genParams` - whether to generate new parameters for the modulus length requested.

`random` - the random bit source to use to generate key bits; can be null.

Throws:

`InvalidParameterException` - if `modlen` is invalid, or unsupported, or if `genParams` is false and there are no precomputed parameters for the requested modulus length.