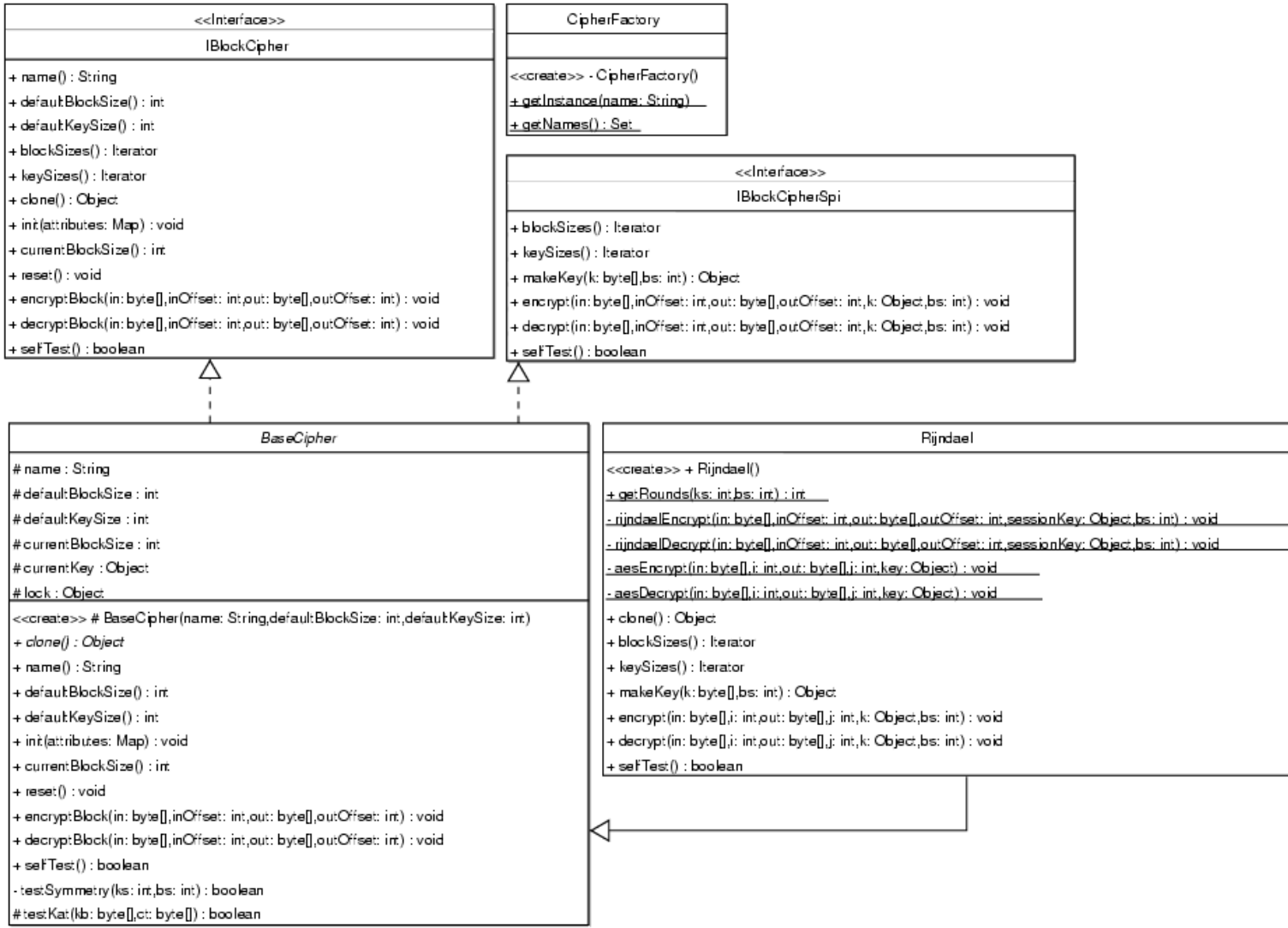
<<Interface>>
IBlockCipher
+ name() : String
+ defaultBlockSize() : int
+ defaultKeySize() : int
+ blockSizes() : Iterator
+ keySizes() : Iterator
+ clone() : Object
+ init(attributes: Map) : void
+ currentBlockSize() : int
+ reset() : void
+ encryptBlock(in: byte[],inOffset: int,out: byte[],outOffset: int) : void
+ decryptBlock(in: byte[],inOffset: int,out: byte[],outOffset: int) : void
+ selfTest() : boolean
CipherFactory
<<create>> - CipherFactory()
+ getInstance(name: String)
+ getNames() : Set
<<Interface>>
IBlockCipherSpi
+ blockSizes() : Iterator
+ keySizes() : Iterator
+ makeKey(k: byte[],bs: int) : Object
+ encrypt(in: byte[],inOffset: int,out: byte[],outOffset: int,k: Object,bs: int) : void
+ decrypt(in: byte[],inOffset: int,out: byte[],outOffset: int,k: Object,bs: int) : void
+ selfTest() : boolean
BaseCipher
name : String
defaultBlockSize : int
defaultKeySize : int
currentBlockSize : int
currentKey : Object
lock : Object
<<create>> # BaseCipher(name: String,defaultBlockSize: int,defaultKeySize: int)
+ clone() : Object
+ name() : String
+ defaultBlockSize() : int
+ defaultKeySize() : int
+ init(attributes: Map) : void
+ currentBlockSize() : int
+ reset() : void
+ encryptBlock(in: byte[],inOffset: int,out: byte[],outOffset: int) : void
+ decryptBlock(in: byte[],inOffset: int,out: byte[],outOffset: int) : void
+ selfTest() : boolean
- testSymmetry(ks: int,bs: int) : boolean
testKat(kb: byte[],ct: byte[]) : boolean
Rijndael
<<create>> + Rijndael()
+ getRounds(ks: int,bs: int) : int
- rijndaelEncrypt(in: byte[],inOffset: int,out: byte[],outOffset: int,sessionKey: Object,bs: int) : void
- rijndaelDecrypt(in: byte[],inOffset: int,out: byte[],outOffset: int,sessionKey: Object,bs: int) : void
- aesEncrypt(in: byte[],i: int,out: byte[],j: int,key: Object) : void
- aesDecrypt(in: byte[],i: int,out: byte[],j: int,key: Object) : void
+ clone() : Object
+ blockSizes() : Iterator
+ keySizes() : Iterator
+ makeKey(k: byte[],bs: int) : Object
+ encrypt(in: byte[],i: int,out: byte[],j: int,k: Object,bs: int) : void
+ decrypt(in: byte[],i: int,out: byte[],j: int,k: Object,bs: int) : void
+ selfTest() : boolean