isEqual:
hash
self

Identifying class and superclass class
superclass

Determining allocation zones zone

Sending messages determined at run time
perform:
perform:withObject:
perform:withObject:withObject:

Identifying proxies isProxy

Testing inheritance relationships isKindOfClass:
isMemberOfClass:

Testing for protocol conformance
conformsToProtocol:

Testing class functionality respondsToSelector:

Managing reference counts retain
release
autorelease
retainCount

autorelease

Adds the receiver to the current autorelease pool and returns self. You add an object to an autorelease pool so that it receives a release messageÐand thus might be deallocatedÐwhen the pool is destroyed. For more information on the autorelease mechanism, see the NSAutoreleasePool class specification.

retain, release

(Class)class

Returns the class object for the receiver's class.

(unsigned int)hash

Returns an unsigned integer that can be used as a table address in a hash table structure. If two obj
determined by the isEqual: method), they must have the same hash value.

isEqual:

(BOOL)isEqual:anObject

Returns YES if the receiver and anObject are equal otherwise returns NO. For the NSObject class
the receiver are compared to determine equality. Subclasses can override this method to redefine w
instance to be equal. For example, a container object might define two containers as equal if they
contents, even though the contains themselves are different objects. See the NSData, NSDictionar
NSString class specifications for examples of the use of this method.

(BOOL)isKindOfClass:(Class)aClass

Returns YES if the receiver is an instance of aClass or an instance of any class that inherits from a
returns NO. For example, in this code isKindOfClass: would return YES because, in the Applicati
inherits from Window:

isMemberOfClass:

(BOOL)isMemberOfClass:(Class)aClass

Returns YES if the receiver is an instance of aClass. Otherwise, it returns NO. For example, in th
isMemberOfClass: would return NO:

isKindOfClass:

(BOOL)isProxy

Returns YES to indicate that the receiver is an NSProxy, rather than an object that descends from N
it returns NO. NSObject's implementation of this method returns NO to indicate that an NSObject
a stand-in for one.

perform:withObject:, perform:withObject:withObject:, methodForSelector: (NSObject class)

perform:(SEL)aSelector withObject:anObject

Sends an aSelector message to the receiver with anObject as an argument. If aSelector is null, an NSInvalidArgumentException is raised.

This method is the same as perform: except that you can supply an argument for the aSelector mes identify a method that takes a single argument of type id.

perform:, perform:withObject:withObject:,
methodForSelector: (NSObject class)

perform:(SEL)aSelector
withObject:anObject
withObject:anotherObject

Sends the receiver an aSelector message with anObject and anotherObject as arguments. If aSelect NSInvalidArgumentException is raised. This method is the same as perform: except that you can s for the aSelector message. aSelector should identify a method that can take two arguments of type

perform:, perform:withObject:, methodForSelector: (NSObject class)

(void)release

As defined in the NSObject class, release decrements the receivers's reference count. When an ob reaches 0, the object is automatically deallocated.

You send release messages only to objects that you ªownº. By definition, you own objects that yo the alloc... or copy... methods. These object creation methods return objects with an implicit refere also own (or perhaps share ownership in) an object that you send a retain message to, since retain i reference count. Each retain message you send an object should be balanced eventually with a rele the object can be deallocated. For more information on the automatic deallocation mechanism, see Foundation Kit.

The application is responsible for determining whether a NO response should be considered an err

Note that if the receiver is able to forward aSelector messages to another object, it will be able to r
albeit indirectly, even though this method returns NO.

forwardInvocation: (NSObject class), + instancesRespondToSelector:(NSObject class)

retain

As defined in the NSObject class, retain increments the receiver's reference count. You send an ob
when you want to prevent it from being deallocated without your express permission.

An object is deallocated automatically when its reference count reaches 0. retain messages increm
and release messages decrement it. For more information on this mechanism, see the introduction

As a convenience, retain returns self, since it is often used in nested expressions:

autorelease, release, retainCount

(unsigned int)retainCount

Returns the receiver's reference count for debugging purposes. You rarely send a retainCount mes
might implement this method in a class to implement your own reference-counting scheme. For ob
released (that is, their release method does nothing), this method should return UINT_MAX, as de

autorelease, release, retain

self

Returns the receiver.

class

superclass

Returns the class object for the receiver's superclass.

(NSZone *)zone

Returns a pointer to the zone from which the receiver was allocated. Objects created without speci
allocated from the default zone, which is returned by NSDefaultMallocZone().