

JERARQUÍA DE CLASES

	pag.
<u>Object</u>	1
<u>Boolean</u>	3
True	
False	
<u>Collection</u>	3
Bag	
<u>IndexedCollection</u>	4
<u>FixedSizeCollection</u>	5
Array	
Interval	
<u>String</u>	5
Symbol	
<u>OrderedCollection</u>	6
<u>SortedCollection</u>	7
<u>Set</u>	7
<u>Dictionary</u>	7
<u>Context</u>	8
HomeContext	
<u>Magnitude</u>	8
<u>Association</u>	9
<u>Character</u>	9
Date	
<u>Number</u>	10
Float	
Fraction	
<u>Integer</u>	10
LargeNegativeInteger	
LargePositiveInteger	
SmallInteger	
Time	
<u>Point</u>	11
Behavior	
Class	
MetaClass	

Clase Object

Métodos de clase:

new	Retorna una instancia del receptor. Todas las variables de instancia de la misma se inicializan con nil .
------------	------------------------------------------------------------------------------------------------------------------

Métodos de instancia:

Comparaciones entre objetos:

= anObject	Este es el método para comparar la igualdad de objetos. Por defecto retorna true si el receptor y <i>anObject</i> son el mismo objeto, sino retorna false . Es redefinido por las subclases según sea necesario.
-------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

== <i>anObject</i>	Retorna true si el receptor y <i>anObject</i> son el mismo objeto (misma identidad), sino retorna false .
~= <i>anObject</i>	Retorna true si el receptor y <i>anObject</i> no son iguales, sino retorna false .
~~ <i>anObject</i>	Retorna true si el receptor y <i>anObject</i> no son idénticos, sino retorna false .

Consultas específicas:

isNil	Retorna true si el receptor es el objeto nil, sino retorna false .
notNil	Retorna true si el receptor no es el objeto nil .
isArray isAssociation isBehavior isBoolean isCharacter isClass isCollection isContext isFloat isInteger isNumber isPoint isString isSymbol	Retorna true si el receptor es una instancia de la clase indicada o una de sus subclases, sino retorna false .

Consultas generales:

class	Retorna la clase del receptor.
species	Retorna una clase que es similar (o la misma) a la clase del receptor. Dicha clase puede usarse para generar copias derivadas del receptor.
isKindOf: <i>aClass</i>	Retorna true si el receptor es una instancia de <i>aClass</i> o una de sus subclases, sino retorna false .
isMemberOf: <i>aClass</i>	Retorna true si el receptor es una instancia de <i>aClass</i> , sino retorna false .

Copias:

copy shallowCopy	Retorna una copia del receptor que tiene las mismas variables de instancia (copia las referencias de las mismas en el nuevo objeto).
deepCopy	Retorna una copia del receptor con copias 'shallow' de cada variable de instancia.

Reporte de errores e inspección:

inspect	Abre una ventana para inspeccionar el contenido del receptor.
error: <i>aString</i>	Reporta un error. Crea una ventana con la secuencia de llamados que describe la condición de error con el mensaje <i>aString</i> como título.
implementedBySubclass	Se usa para reportar un error cuando un método de una clase abstracta debe ser redefinido en las subclases.

Variables de instancia indexadas:

at: <i>anInteger</i>	Retorna el objeto almacenado en el receptor en la posición dada por <i>anInteger</i> . Si el receptor no tiene variables de instancia indexadas o <i>anInteger</i> está fuera de rango reporta un error.
at: <i>anInteger</i> put: <i>anObject</i>	Retorna <i>anObject</i> . Reemplaza el objeto en la variable de instancia indexada del receptor en la posición <i>anInteger</i> con <i>anObject</i> . Si el receptor no tiene variables de instancia indexadas o <i>anInteger</i> está fuera de rango reporta un error.
size	Retorna el número de variables de instancia indexadas del receptor.

Varios:

yourself	Retorna el receptor.
asString	Retorna la representación del receptor como cadena de caracteres.

Clases True y False (subclases de Boolean)

Métodos de instancia:

Operaciones lógicas:

not	Retorna true si el receptor es false , y viceversa.
& aBoolean	Retorna true si <i>aBoolean</i> y el receptor son true , caso contrario devuelve false .
 aBoolean	Retorna true si <i>aBoolean</i> o el receptor son true , caso contrario retorna false .
eqv: aBoolean	Retorna true si tanto el receptor como <i>aBoolean</i> son true , sino retorna false .
xor: aBoolean	Retorna true si el receptor y <i>aBoolean</i> no son equivalentes, sino retorna false .

and: aBlock	Si el receptor es true , retorna el resultado de evaluar <i>aBlock</i> (sin argumentos), caso contrario retorna false .
or: aBlock	Si el receptor es false , retorna el resultado de evaluar <i>aBlock</i> , sino retorna true .

Control condicional simple y compuesto:

ifTrue: aBlock	Si el receptor es true , retorna el resultado de evaluar <i>aBlock</i> (sin argumentos), caso contrario retorna nil .
ifFalse: aBlock	Si el receptor es false , retorna el resultado de evaluar <i>aBlock</i> (sin argumentos), caso contrario retorna nil .
ifFalse: falseBlock ifTrue: trueBlock ifTrue: trueBlock ifFalse: falseBlock	Si el receptor es true , retorna el resultado de evaluar <i>trueBlock</i> , sino retorna el resultado de evaluar <i>falseBlock</i> . Ambos bloques serán evaluados sin argumentos.

Clase Collection (subclase de Object)

Métodos de instancia:

Agregar y eliminar elementos:

add: anObject	Retorna <i>anObject</i> . Agrega <i>anObject</i> a la colección receptora.
addAll: aCollection	Retorna <i>aCollection</i> . Agregar cada elemento de <i>aCollection</i> a los elementos del receptor.
remove: anObject	Retorna <i>anObject</i> . Quita el elemento igual a <i>anObject</i> de la colección receptora. Si no lo encuentra reporta un error.
remove: anObject ifAbsent: aBlock	Retorna <i>anObject</i> . Quita el elemento igual a <i>anObject</i> de la colección receptora. Si no lo encuentra evalúa el bloque <i>aBlock</i> (sin argumentos).
removeAll: aCollection	Retorna <i>aCollection</i> . Quita todos los elementos contenidos en <i>aCollection</i> del conjunto de elementos del receptor.

Tamaño de la colección:

size	Retorna el número de elementos contenidos en la colección receptora.
isEmpty	Retorna true si el receptor no contiene elementos (se encuentra vacío), sino retorna false .
notEmpty	Retorna true si el receptor no se encuentra vacío, caso contrario retorna false .

Conversión entre colecciones:

asArray	Retorna una instancia de Array conteniendo los elementos del receptor.
asBag	Retorna una instancia de Bag conteniendo los elementos del receptor.
asOrderedCollection	Retorna una instancia de OrderedCollection conteniendo los elementos del receptor.
asSet	Retorna una instancia de Set conteniendo los elementos del receptor.
asSortedCollection	Retorna una instancia de SortedCollection conteniendo los elementos del receptor en orden ascendente.
asSortedCollection: aBlock	Retorna una instancia de SortedCollection conteniendo los elementos del receptor ordenados de acuerdo al bloque <i>aBlock</i> .

Consultas:

includes: <i>anObject</i>	Retorna true si el receptor contiene un elemento igual a <i>anObject</i> , sino retorna false .
occurrencesOf: <i>anObject</i>	Retorna la cantidad de elementos contenidos en el receptor que son iguales a <i>anObject</i> .
detect: <i>aBlock</i>	Retorna el primer elemento del receptor que causa que <i>aBlock</i> evalúe true , con ese elemento como argumento. Si ningún elemento de la colección receptora cumple esta condición, reporta un error.
detect: <i>aBlock ifNone:</i> <i>exceptionBlock</i>	Retorna el primer elemento del receptor que causa que <i>aBlock</i> evalúe true , con ese elemento como argumento. Si ningún elemento de la colección cumple esta condición, evalúa el bloque <i>exceptionBlock</i> (sin argumentos).

Iterativos:

do: <i>aBlock</i>	Para cada elemento de la colección receptora, evalúa el bloque <i>aBlock</i> con ese elemento como argumento. (Este método es implementado en las subclases)
collect: <i>aBlock</i>	Para cada elemento de la colección receptora, evalúa <i>aBlock</i> con ese elemento como argumento. Retorna una nueva colección (instancia de la misma clase que el receptor) conteniendo como elementos los resultados de las evaluaciones.
select: <i>aBlock</i>	Para cada elemento de la colección receptora, evalúa <i>aBlock</i> con ese elemento como argumento. Retorna una nueva colección (instancia de la misma clase que el receptor) conteniendo aquellos elementos del receptor cuya evaluación produjo como resultado true .
reject: <i>aBlock</i>	Para cada elemento de la colección receptora, evalúa <i>aBlock</i> con ese elemento como argumento. Retorna una nueva colección (instancia de la misma clase que el receptor) conteniendo aquellos elementos del receptor cuya evaluación produjo como resultado false .
inject: <i>initialValue into:</i> <i>aBinaryBlock</i>	<p>Para cada elemento del receptor, evalúa el bloque de dos argumentos <i>aBinaryBlock</i> con ese elemento como segundo argumento. Para el primer argumento se usa como valor inicial <i>initialValue</i>, y luego el resultado de la evaluación previa del bloque.</p> <p>Retorna finalmente el resultado de la última evaluación. Si el receptor es una colección vacía retorna <i>initialValue</i>.</p> <p>Ejemplo: <code>#(1 2 3 4) inject: 0 into: [:x :y x+y]. → 10</code></p>

Copias de colecciones:

shallowCopy	Retorna una copia del receptor que comparte los mismos elementos.
deepCopy	Retorna una copia del receptor con copias (<i>shallowCopy</i>) de sus elementos.

Clase IndexedCollection (subclase de Collection)

Métodos de instancia:

Igualdad:

= <i>aCollection</i>	Retorna true si los elementos contenidos en el receptor y en <i>aCollection</i> son iguales.
-----------------------------	-----------------------------------------------------------------------------------------------------

Acceder y cambiar elementos:

at: <i>anInteger</i>	Retorna el objeto almacenado en el receptor en la posición dada por el índice <i>anInteger</i> .
at: <i>anInteger put:</i> <i>anObject</i>	Retorna <i>anObject</i> . Reemplaza el objeto almacenado en el receptor en la posición <i>anInteger</i> por <i>anObject</i> .
atAll: <i>aCollection put:</i> <i>anObject</i>	Retorna la colección receptora luego de reemplazar aquellos elementos del receptor indicados por los índices contenidos en <i>aCollection</i> por <i>anObject</i> .
atAllPut: <i>anObject</i>	Retorna el receptor luego de reemplazar cada uno de sus elementos por <i>anObject</i> .
first	Retorna el primer elemento del receptor. Reporta un error si la colección receptora está vacía.
last	Retorna el último elemento del receptor. Reporta un error si no hay elementos.
indexOf: <i>anObject</i>	Retorna el índice o posición del elemento igual a <i>anObject</i> en el receptor. Si no encuentra tal elemento retorna 0 (cero).

indexOf: <i>anObject</i> ifAbsent: <i>aBlock</i>	Retorna el índice o posición del elemento igual a <i>anObject</i> en el receptor. Si no encuentra tal elemento evalúa el bloque <i>aBlock</i> (sin argumentos).
findFirst: <i>aBlock</i>	Retorna el índice del primer elemento del receptor que causa que <i>aBlock</i> evalúe true (con ese elemento como argumento). Si no encuentra el elemento reporta un error.
findLast: <i>aBlock</i>	Retorna el índice del último elemento del receptor que causa que <i>aBlock</i> evalúe true (con ese elemento como argumento). Si no encuentra el elemento reporta un error.

Reemplazar elementos:

replaceFrom: <i>start to: stop</i> with: <i>aCollection</i>	Retorna el receptor. Reemplaza los elementos del receptor desde las posiciones <i>start</i> hasta <i>stop</i> con los elementos de <i>aCollection</i> . El número de elementos a reemplazar debe ser el mismo que el número de elementos en la colección, sino reportar un error.
replaceFrom: <i>start to: stop</i> with: <i>aCollection</i> startingAt: <i>repStart</i>	Retorna el receptor. Reemplaza los elementos del receptor desde las posiciones <i>start</i> hasta <i>stop</i> con los elementos de <i>aCollection</i> , empezando desde la posición <i>repStart</i> de esta última colección.
replaceFrom: <i>start to: stop</i> withObject: <i>anObject</i>	Reemplaza cada uno de los elementos del receptor desde las posiciones <i>start</i> hasta <i>stop</i> por el objeto <i>anObject</i> . Retorna <i>anObject</i> .

Concatenar, copiar, combinar e invertir colecciones indexadas:

, <i>aCollection</i>	Retorna una nueva colección (instancia de la misma clase que el receptor) conteniendo los elementos del receptor seguidos de los elementos de <i>aCollection</i> . Concatena ambas colecciones.
copyFrom: <i>start to: stop</i>	Retorna una nueva colección (instancia de la misma clase que el receptor) conteniendo los elementos del receptor indexados desde las posiciones <i>start</i> hasta <i>stop</i> .
copyReplaceFrom: <i>start to: stop</i> with: <i>aCollection</i>	Retorna una nueva colección, copia del receptor, en la cual los elementos entre las posiciones <i>start</i> y <i>stop</i> fueron reemplazados por los elementos de <i>aCollection</i> .
copyWith: <i>anObject</i>	Retorna una copia del receptor a la cual se agrega <i>anObject</i> .
copyWithout: <i>anObject</i>	Retorna una copia del receptor excluyendo el primer elemento que es igual a <i>anObject</i> , si hubiera.
reversed	Retorna una nueva colección que contiene los elementos del receptor en orden inverso.

Iteraciones:

reverseDo: <i>aBlock</i>	Para cada elemento del receptor, comenzando por el último, evalúa el bloque <i>aBlock</i> con ese elemento como argumento.
with: <i>aCollection</i> do: <i>aBlock</i>	Para cada par de elementos (el primero del receptor y el segundo de <i>aCollection</i>), evalúa <i>aBlock</i> con esos elementos como argumentos. El receptor y <i>aCollection</i> deben tener el mismo número de elementos, sino reporta un error.

Clase FixedSizeCollection (subclase de IndexedCollection, Collection)

Métodos de clase:

new: <i>anInteger</i>	Retorna una instancia del receptor con espacio asignado para <i>anInteger</i> variables de instancia indexadas.
------------------------------	-----------------------------------------------------------------------------------------------------------------

Dado que no pueden cambiar de tamaño una vez creadas, las instancias de esta clase o de sus subclases **no admiten** el uso de los métodos definidos en **Collection** para agregar y eliminar elementos, tales como **add:**, **remove:**, **removeAll:**, etc

Clase String (subclase de FixedSizeCollection, IndexedCollection, Collection)

Métodos de instancia:

< <i>aString</i> <= <i>aString</i> = <i>aString</i> > <i>aString</i> >= <i>aString</i>	Comparadores para relaciones de mayor, menor o igual entre cadenas de caracteres. Retorna true o false según el caso.
---------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------

asLowerCase	Retorna una instancia de String conteniendo los mismos caracteres que el receptor pero con todos los caracteres alfabéticos en minúscula.
asUpperCase	Retorna una instancia de String conteniendo los mismos caracteres que el receptor pero con todos los caracteres alfabéticos en mayúscula.
asInteger	Retorna el resultado de convertir el texto del receptor en un número entero (se espera que el receptor sea una secuencia de dígitos precedida opcionalmente por un signo menos).
asFloat	Retorna el resultado de convertir el texto del receptor en un número real.
asSymbol	Retorna el símbolo (instancia de Symbol) cuyos caracteres son los mismos que los del receptor.
asDate	Retorna una instancia de Date que representa la fecha descripta por el receptor.
trimBlanks	Retorna un String conteniendo el receptor sin ningún espacio al principio o al final del mismo.

Clase OrderedCollection (subclase de IndexedCollection, Collection)

Métodos de clase:

new	Crea y retorna una instancia que inicialmente tiene capacidad para almacenar 12 elementos.
new: anInteger	Crea y retorna una instancia que inicialmente tiene capacidad para almacenar anInteger elementos.

Nota: cada instancia de OrderedCollection guarda internamente su contenido en una instancia de Array, el tamaño mencionado corresponde al tamaño del mismo. Para crear instancias de OrderedCollection debe usarse el método **new**.

Métodos de instancia:

Agregar y eliminar elementos:

add: anObject	Agrega <i>anObject</i> después del último elemento del receptor. Retorna <i>anObject</i> .
add: anObject beforeIndex: anInteger	Retorna <i>anObject</i> . Agrega <i>anObject</i> en la posición <i>anInteger</i> - 1 de la colección receptora. Si <i>anInteger</i> esta fuera de los límites de la colección reporta un error.
add: anObject afterIndex: anInteger	Retorna <i>anObject</i> . Agrega <i>anObject</i> en la posición <i>anInteger</i> + 1 de la colección receptora. Si <i>anInteger</i> esta fuera de los límites de la colección reporta un error.
add: newObject before: oldObject	Retorna <i>newObject</i> . Agrega <i>newObject</i> inmediatamente antes del elemento <i>oldObject</i> en la colección receptora. Si <i>oldObject</i> no esta en la colección reporta un error.
add: newObject after: oldObject	Retorna <i>newObject</i> . Agrega <i>newObject</i> inmediatamente después del elemento <i>oldObject</i> en la colección receptora. Si <i>oldObject</i> no esta en la colección reporta un error.
addFirst: anObject	Retorna <i>anObject</i> . Agrega <i>anObject</i> antes del primer elemento del receptor.
addLast: anObject	Retorna <i>anObject</i> . Agrega <i>anObject</i> después del último elemento del receptor.
addAllFirst: aCollection	Retorna <i>aCollection</i> . Agrega todos los elementos de <i>aCollection</i> al receptor antes del primer elemento del mismo.
addAllLast: aCollection	Retorna <i>aCollection</i> . Agrega todos los elementos de <i>aCollection</i> al receptor después del último elemento del mismo.
removeFirst	Quita y devuelve el primer elemento del receptor. Reporta un error si es la colección está vacía.
removeLast	Quita y devuelve el último elemento del receptor. Reporta un error si es la colección está vacía.
removeIndex: anInteger	Retorna el receptor. Quita el elemento del mismo en la posición <i>anInteger</i> . Si <i>anInteger</i> es un índice inválido para el receptor, reporta un error.

Acceder y cambiar elementos:

at: anInteger	Retorna el elemento del receptor en la posición <i>anInteger</i> . Si <i>anInteger</i> es un índice inválido para el receptor reporta un error.
at: anInteger put: anObject	Retorna <i>anObject</i> . Reemplaza el objeto almacenado en el receptor en la posición <i>anInteger</i> por <i>anObject</i> . Si <i>anInteger</i> es un índice inválido para la colección receptora reporta un error.

before: <i>anObject</i>	Retorna el objeto anterior a <i>anObject</i> en la colección receptora. Si <i>anObject</i> no es un elemento de la colección reporta un error.
before: <i>anObject</i> ifNone: <i>aBlock</i>	Retorna el objeto anterior a <i>anObject</i> en la colección receptora. Si <i>anObject</i> no es un elemento de la colección evalúa el bloque <i>aBlock</i> (sin argumentos).
after: <i>anObject</i>	Retorna el objeto siguiente a <i>anObject</i> en la colección receptora. Si <i>anObject</i> no es un elemento de la colección reporta un error.
after: <i>anObject</i> ifNone: <i>aBlock</i>	Retorna el objeto siguiente a <i>anObject</i> en la colección receptora. Si <i>anObject</i> no es un elemento de la colección evalúa el bloque <i>aBlock</i> (sin argumentos).

Clase SortedCollection (subclase de OrderedCollection, IndexedCollection, Collection)

Métodos de clase:

new	Retorna una instancia de SortedCollection cuyos elementos están ordenados en forma ascendente.
sortBlock: <i>aBlock</i>	Retorna una instancia de SortedCollection cuyo ordenamiento estará dado por el bloque <i>aBlock</i> .

Métodos de instancia:

sortBlock	Retorna el bloque que determina el orden de la colección.
sortBlock: <i>aBlock</i>	Retorna el receptor. Asigna el bloque de ordenamiento del receptor como <i>aBlock</i> y reordena la colección.

add: <i>anObject</i> addAll: <i>aCollection</i>	A diferencia de OrderedCollection , agregan los elementos en forma ordenada a la colección.
at: <i>anInteger</i> put: <i>anObject</i> add: <i>newObject</i> before: <i>oldObject</i> add: <i>newObject</i> after: <i>oldObject</i> addFirst: <i>anObject</i> addLast: <i>anObject</i> addAllFirst: <i>aCollection</i> addAllLast: <i>aCollection</i>	Estos métodos reportan un error dado que el sortBlock define el orden de los elementos de la colección.

Clase Set (subclase de Collection)

- Un **Set** representa un conjunto. Es decir una colección de objetos sin orden, sin claves externas.
- Todos los elementos ocurren una sola vez en el conjunto (son únicos).

Clase Dictionary (subclase de Set, Collection)

- **Dictionary** es una colección de pares de objetos **clave/valor** (key/value).
- Las claves en una instancia de **Dictionary** deben ser únicas.
- Internamente una instancia de **Dictionary** almacena pares clave/valor como un conjunto de instancias de **Association**.

Métodos de instancia:

Agregar asociaciones clave/valor:

at: <i>aKey</i> put: <i>anObject</i>	Retorna <i>anObject</i> . Si el receptor contiene un par clave/valor cuya clave es <i>aKey</i> , reemplaza el valor del par por <i>anObject</i> . Caso contrario agrega un nuevo par con <i>aKey/anObject</i> al receptor.
add: <i>anAssociation</i>	Retorna <i>anAssociation</i> . Agrega <i>anAssociation</i> al receptor.

Acceder a los elementos:

at: <i>aKey</i>	Retorna el valor del par clave/valor del receptor cuya clave es igual a <i>aKey</i> . Si no lo encuentra reporta un error.
at: <i>aKey</i> ifAbsent: <i>aBlock</i>	Retorna el valor del par clave/valor del receptor cuya clave es igual a <i>aKey</i> . Si no lo encuentra evalúa <i>aBlock</i> (sin argumentos).

keyAtValue: <i>anObject</i>	Retorna la clave en el receptor cuyo valor es igual a <i>anObject</i> . Si no lo encuentra retorna nil .
keyAtValue: <i>anObject</i> ifAbsent: <i>aBlock</i>	Retorna la clave en el receptor cuyo valor es igual a <i>anObject</i> . Si no lo encuentra evalúa <i>aBlock</i> (sin argumentos).

Quitar asociaciones clave-valor:

removeKey: <i>aKey</i>	Retorna el receptor. Quita el par clave/valor cuya clave es igual a <i>aKey</i> del mismo. Si no lo encuentra reporta un error.
removeKey: <i>aKey</i> ifAbsent: <i>aBlock</i>	Retorna <i>aKey</i> . Quita el par clave/valor cuya clave es igual a <i>aKey</i> del receptor. Si no lo encuentra evalúa <i>aBlock</i> (sin argumentos).
remove: <i>anObject</i> ifAbsent: <i>aBlock</i>	Quitar un par en base al valor del mismo. Este método reporta un error dado que los valores de un diccionario no son únicos.

Obtener todas las claves o valores:

keys	Retorna una instancia de Set que contiene todas las claves del receptor.
values	Retorna una instancia de Bag que contiene todos los valores de los pares del receptor.

Consultas:

includes: <i>anObject</i>	Retorna true si el receptor contiene un par clave/valor cuyo valor sea igual a <i>anObject</i> , sino retorna false .
includesKey: <i>aKey</i>	Retorna true si el receptor contiene la clave <i>aKey</i> , sino retorna false .
occurrencesOf: <i>anObject</i>	Retorna el número de pares clave/valor en el receptor cuyo valor es igual a <i>anObject</i> .

Iterativos:

do: <i>aBlock</i>	Retorna el receptor. Para cada valor en el receptor, evalúa <i>aBlock</i> con dicho valor como argumento.
keysDo: <i>aBlock</i>	Retorna el receptor. Para cada clave en el receptor, evalúa <i>aBlock</i> con dicha clave como argumento.
select: <i>aBlock</i>	Para cada par clave/valor en el receptor, evalúa <i>aBlock</i> con el valor del par como argumento. Retorna un nuevo diccionario conteniendo los pares para los cuales la evaluación de <i>aBlock</i> da como resultado true .

Clase Context (subclase de Object)

Métodos de instancia:

value	Retorna el resultado de evaluar el bloque receptor (el cual no debe tener argumentos).
value: <i>anObject</i>	Retorna el resultado de evaluar el bloque receptor con <i>anObject</i> . El bloque receptor debe tener un solo argumento.
value: <i>arg1</i> value: <i>arg2</i>	Retorna el resultado de evaluar el bloque receptor con los argumentos <i>arg1</i> y <i>arg2</i> . El bloque receptor debe tener dos argumentos.
whileFalse: <i>aBlock</i>	Evalua repetitivamente el bloque receptor y <i>aBlock</i> , hasta que el resultado de la evaluación del bloque receptor sea true . Retorna nil .
whileTrue: <i>aBlock</i>	Evalua repetitivamente el bloque receptor y <i>aBlock</i> , hasta que el resultado de la evaluación del bloque receptor sea false . Retorna nil .

Clase Magnitude (subclase de Object)

Métodos de instancia:

< <i>aMagnitude</i> <= <i>aMagnitude</i> = <i>aMagnitude</i> > <i>aMagnitude</i> >= <i>aMagnitude</i>	Comparadores para relaciones de mayor, menor o igual entre magnitudes. Retorna true o false según el caso.
------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------

between: <i>min</i> and: <i>max</i>	Retornar true si el receptor es mayor o igual a <i>min</i> y menor o igual a <i>max</i> , caso contrario retornar false .
max: <i>aMagnitude</i>	Retornar el receptor si es mayor que <i>aMagnitude</i> , sino retornar <i>aMagnitude</i> .
min: <i>aMagnitude</i>	Retornar el receptor si es menor que <i>aMagnitude</i> , sino retornar <i>aMagnitude</i> .

Clase Association (subclase de Magnitude)

Métodos de clase:

key: <i>aKey</i> value: <i>anObject</i>	Retorna una instancia de Association cuya clave es inicializada como <i>aKey</i> y cuyo valor es inicializado como <i>anObject</i> .
-------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------

Métodos de instancia:

key	Retorna la clave del receptor.
key: <i>anObject</i>	Asigna <i>anObject</i> como clave del receptor. Retorna el receptor.
value	Retorna el valor del receptor.
value: <i>anObject</i>	Asigna <i>anObject</i> como valor del receptor. Retorna el receptor.

Clase Character (subclase de Magnitude)

Métodos de instancia:

asciiValue	Retorna el número correspondiente al código ASCII del receptor.
asLowerCase	Si el receptor es una letra, retorna el caracter en letra minúscula correspondiente al mismo, sino retorna el receptor.
asString	Retorna una instancia de String de longitud 1 que contiene el receptor.
asUpperCase	Si el receptor es una letra, retorna el caracter en letra mayúscula correspondiente al mismo, sino retorna el receptor.
isAlphaNumeric	Retorna true si el receptor es un caracter alfanumérico, sino retorna false .
isDigit	Retorna true si el receptor está en el rango de los caracteres 0 a 9, sino retorna false .
isLetter	Retorna true si el receptor está en el rango de los caracteres a hasta z o A hasta Z, sino retorna false .
isLowerCase	Retorna true si el receptor está en el rango de los caracteres a hasta z, sino retorna false .
isUpperCase	Retorna true si el receptor está en el rango de los caracteres A hasta Z, sino retorna false .
isVowel	Retorna true si el receptor es una vocal (mayúscula o minúscula), sino retorna false .

Clase Number (subclase de Magnitude)

Métodos de instancia:

<i>* aNumber</i> <i>+ aNumber</i> <i>- aNumber</i> <i>/ aNumber</i>	Operadores aritméticos.
<i>// aNumber</i>	Retorna el número entero que resulta de la división entera entre el receptor y <i>aNumber</i> .
<i>\ aNumber</i>	Retorna el resto de la división entera entre el receptor y <i>aNumber</i> .

asInteger	Retorna el número entero que es resultado de redondear el receptor (el entero más cercano).
negated	Retorna el número que resulta de negar el receptor.
numerator	Retorna el numerador del receptor. Por defecto es el receptor, se redefine en la clase Fraction .
denominator	Retorna el denominador del receptor. Por defecto es 1, se redefine en la clase Fraction .
even	Retorna true si la parte entera del receptor es par, sino retorna false .
odd	Retorna true si la parte entera del receptor es impar, sino retorna false .
negative	Retorna true si el receptor es menor que cero, sino retorna false .
positive	Retorna true si el receptor es mayor que cero, sino retorna false .
sign	Retorna 1 si el receptor es mayor que cero, -1 si es menor que cero, sino retorna 0.
abs	Retorna el valor absoluto del receptor.
squared	Retorna el receptor multiplicado por sí mismo.
sqrt	Retorna una instancia de Float que es la raíz cuadrada del receptor.

to: aNumber	Retorna una instancia de Interval definida entre los números dados por el receptor y <i>aNumber</i> , con incremento 1.
to: sNumber by: inc	Retorna una instancia de Interval definida entre el receptor y <i>sNumber</i> , con el incremento dado por <i>inc</i> .
to: aNumber do: aBlock	Evalúa el bloque de un argumento dado por <i>aBlock</i> , para los números comprendidos entre el receptor y <i>aNumber</i> , con incremento 1.
to: sNumber by: inc do: aBlock	Evalúa el bloque de un argumento dado por <i>aBlock</i> , para los números comprendidos entre el receptor y <i>sNumber</i> , con el incremento dado por <i>inc</i> .

@ aNumber	Retorna una instancia de la clase Point en la cual las coordenadas (x, y) están dadas por el receptor y <i>aNumber</i> .
------------------	---------------------------------------------------------------------------------------------------------------------------------

Clase Integer (subclase de Number, Magnitude)

Métodos de instancia:

asBoolean	Retorna true si el receptor es distinto de cero.
asCharacter	Retorna la instancia de Character cuyo código ASCII coincide con el valor del receptor.
factorial	Devuelve el factorial del receptor.
gcd: anInteger	Retorna el máximo divisor común entre el receptor y <i>anInteger</i> .
timesRepeat: aBlock	Evalúa el bloque <i>aBlock</i> <i>n</i> veces, donde <i>n</i> es el receptor.

Clase Point (subclase de Object)

Métodos de instancia:

<i>< aPoint</i> <i><= aPoint</i> <i>= aPoint</i> <i>> aPoint</i> <i>>= aPoint</i>	Comparadores para relaciones de mayor, menor o igual entre puntos. Una relación dada es verdadera cuando se verifica para ambas coordenadas: <i>x</i> e <i>y</i> . Retorna true o false según el caso.
<i>+ delta</i> <i>- delta</i> <i>* scale</i> <i>/ scale</i>	Operaciones aritméticas entre puntos (pares coordenados). Tanto <i>delta</i> como <i>scale</i> pueden ser instancias de Number o Point .
negated	Retorna una instancia de Point con las coordenadas <i>x</i> e <i>y</i> del receptor negadas.
transpose	Retorna una instancia de Point con las coordenadas <i>x</i> e <i>y</i> del receptor intercambiadas.
between: aPoint and: bPoint	Retorna true si el receptor es mayor o igual a <i>aPoint</i> y menor o igual a <i>bPoint</i> , sino retorna false .
max: aPoint	Retorna un punto formado con el máximo de las coordenadas <i>x</i> y el máximo de las coordenadas <i>y</i> entre el receptor y el argumento <i>aPoint</i> .
min: aPoint	Retorna un punto formado con el mínimo de las coordenadas <i>x</i> y el mínimo de las coordenadas <i>y</i> entre el receptor y el argumento <i>aPoint</i> .
x	Retorna la coordenada <i>x</i> del receptor.
y	Retorna la coordenada <i>y</i> del receptor.
x: aNumber	Asigna el valor <i>aNumber</i> a la coordenada <i>x</i> del receptor. Retorna el receptor.
y: aNumber	Asigna el valor <i>aNumber</i> a la coordenada <i>y</i> del receptor. Retorna el receptor.