

Programación Orientada a Objetos

Profesora Claudia Cappelletti

Colecciones

- 1) Jerarquías Magnitud y Collection en Smalltalk
- 2) Tipos de colecciones:
 - a) Array
 - b) OrderedCollection
 - c) SortedCollection
 - d) Dictionary
- 3) Testeo y conversión de colecciones
- 4) Enumeradores de colección
- 5) Actividad 6

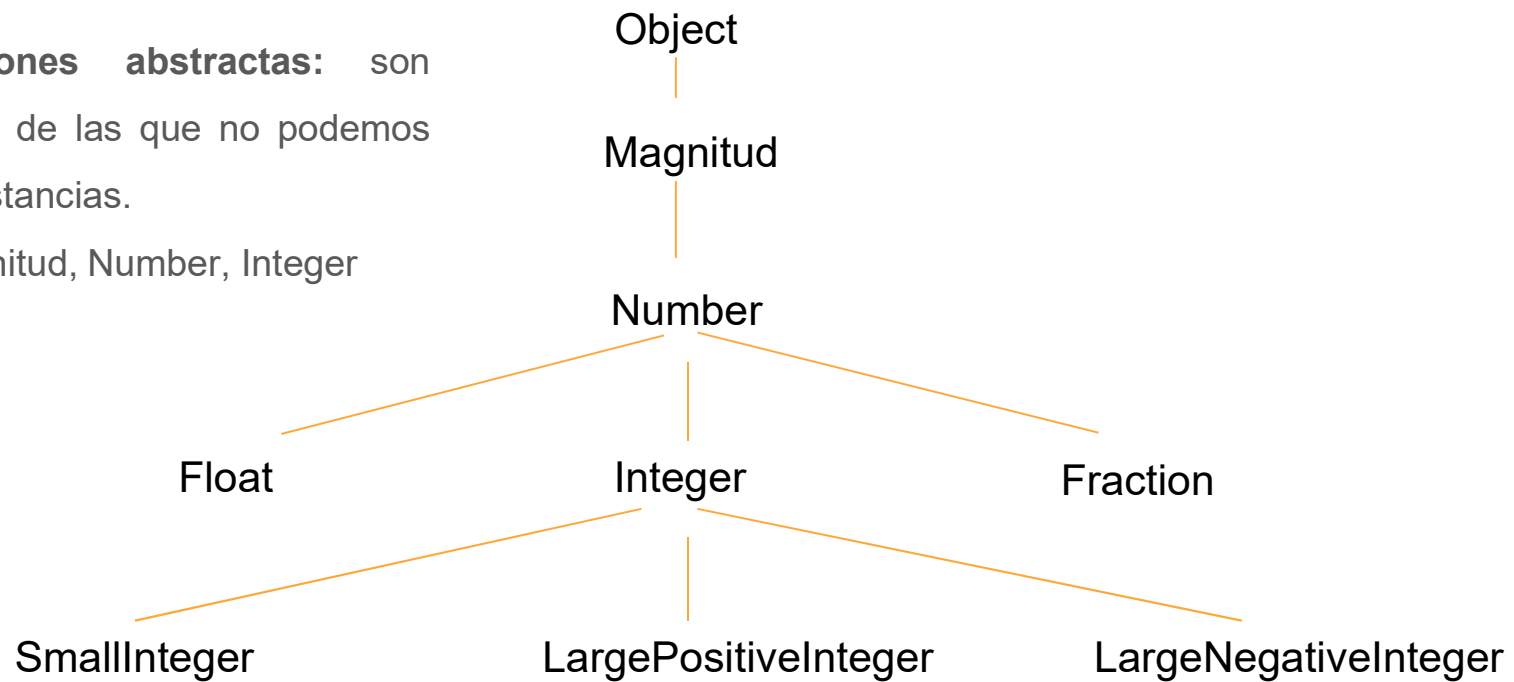


1) Jerarquías en Smalltalk

Jerarquía Magnitud

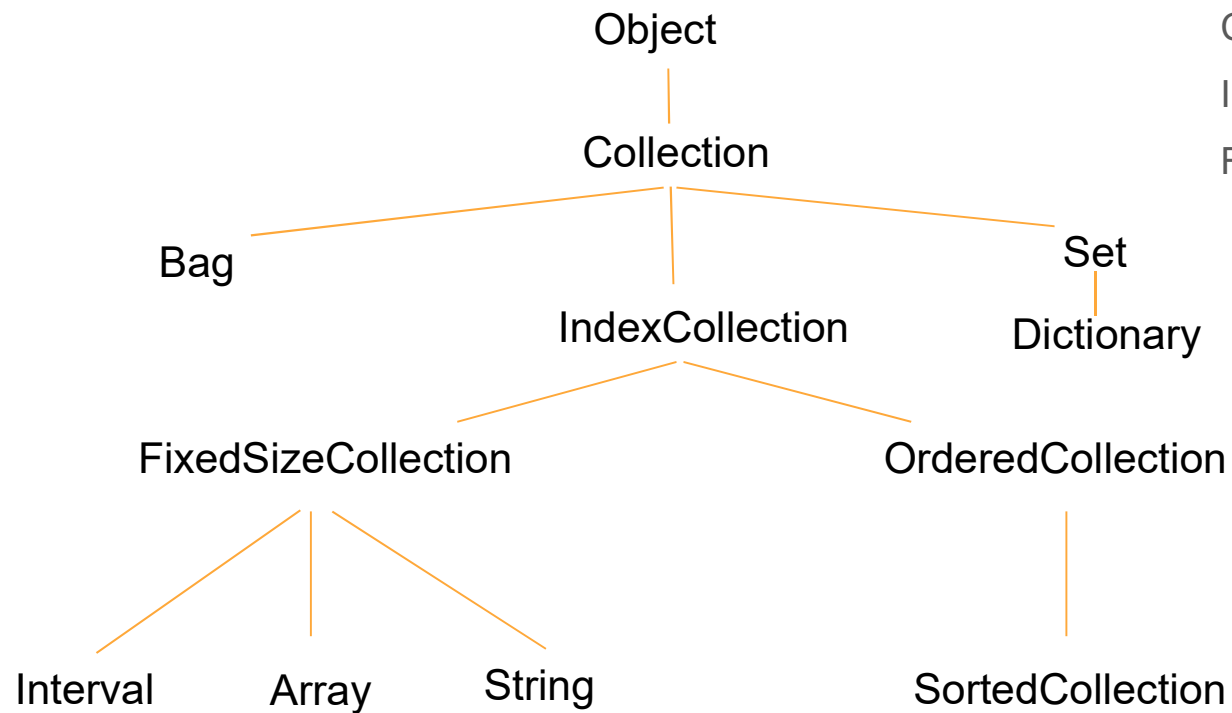
Colecciones abstractas: son aquellas de las que no podemos tener instancias.

Ej: Magnitud, Number, Integer



1) Jerarquías en Smalltalk

Jerarquía Collection



Colecciones abstractas:

Collection,
IndexedCollection,
FixedSizeCollection

2) Tipos de Colecciones

No nos interesa cómo están implementadas las colecciones sino lo que podemos hacer con ellas.

Bag: son colecciones no ordenadas que puede contener elementos repetidos.

Set: son colecciones no ordenadas que no permite elementos repetidos.

Interval: 1 to: 10 by: 2

Algunas colecciones en Smalltalk pueden ser heterogéneas. Por ejemplo: Array, OrderedCollection, SortedCollection.

String es homogéneo, está formado sólo por caracteres.



2) Tipos de Colecciones

a) Array (subclase de FixedSizeCollection)

|a|

a:=Array new:3.

a at:1 put:5.

a at:2 put:8.

a at:3 put:2.

a inspect. Muestra #(5 8 2)

(a at:2) inspect. Muestra 8

El array no admite el add: y tampoco el remove:



2) Tipos de Colecciones

b) OrderedCollection (subclase de IndexedCollection)

oc		Para insertar	Para recuperar	Para eliminar
oc:=OrderedCollection new.		add:unObjeto	first	removeFirst
oc add:5.		addFirst:	last	removeLast
oc add:8.		addLast:	at:	remove:
oc add:2.		add: before:		remove: ifAbsent:
oc inspect.	Muestra #(5 8 2)	add: after:		
(oc at:2) inspect.	Muestra 8	at: put:		

La OrderedCollection admite el **at:** y el **at: put:**, con ellos podemos manejar la colección en forma indexada.



2) Tipos de Colecciones

c) SortedCollection (subclase de OrderedCollection)

|sc|

sc:=SortedCollection new.

sc add:5.

El add agrega un elemento a la SortedCollection en forma ordenada.

sc add:8.

sc add:2.

sc inspect.

Muestra #(2 5 8)

(sc at:2) inspect.

Muestra 5

Para agregar elementos en forma ordenada, en algunas ocasiones se necesita el sortBlock[] para indicar el criterio de ordenación. Si sólo se hace el new, asume que el sortBlock por defecto ordena de menor o igual a mayor [:x :y|x<=y].

En otros casos siempre deberemos indicar el criterio de ordenación:

S:=SortedCollection new sortBlock [:x :y| x verNombre <= y verNombre].



2) Tipos de Colecciones

d) Dictionary (subclase de Set)

|d|

```
d:=Dictionary new at:'lunes' put:'Física';  
                at:'martes' put:'Computación';  
                at:'miércoles' put:'Matemática';  
                at:'jueves' put:'Computación';  
                at:'viernes' put:'Matemática';  
                yourself.
```

Puedo también agregar una Asociación creada previamente. Una Asociación está formada por una clave y un valor. La forma de agregar una asociación es: d add:unaAsociación.

d inspect. Muestra la asociación entre clave y valor

Física	Computación	Matemática	Computación	Matemática	← Valores, pueden repetirse
lunes	martes	miércoles	jueves	viernes	← Claves, no deben repetirse

3) Testeos y Conversión de colecciones

Testeos de colecciones:

size

isEmpty

includes:unObjeto

occurrencesOf: un Objeto

Conversión de colecciones:

asSet

asBag

asArray

asOrderedCollection

asSortedCollection

Ejemplo de conversión de colecciones:

```
|oc|
```

```
oc:=OrderedCollection new.
```

```
oc add:5.
```

```
oc add:8.
```

```
oc add:2.
```

```
oc inspect.
```

```
sc:= oc asSortedCollection.
```

```
sc inspect. Muestra #(2 5 8)
```



4) Enumeradores de colecciones

Enumeradores de colecciones:

do: itera sobre cada elemento de la colección.

Ejemplo: col **do:** [:pieza| pieza reset].

↑
es un objeto

to: do: repite un número conocido de veces un proceso.

Ejemplo: |suma col|

suma:=0.

col:=#(1 3 6 8).

1 **to:**(col size) **do:**[:i | suma:=suma + col at:i].

↑
es un índice

No se debe agregar ni remover un elemento dentro de un **do:**

Porque trabaja internamente con el tamaño de la colección.



4) Enumeradores de colecciones

Uso del select, reject, collect y detect

Todos ellos están implementados internamente con un do:

select col1:= alumnos select:[:alu| alu verNota=8].

Retorna en otra colección los alumnos con nota igual a 8

reject col2:= alumnos reject:[:alu| alu verNota=8].

Retorna en otra colección los alumnos con nota distinto a 8

collect col3:= alumnos collect:[:alu| alu verNombre].

Retorna en otra colección los nombres de los alumnos

detect a:= alumnos detect:[:alu| alu verNota=8] ifNone:[^nil].

Retorna el primer alumno con nota igual a 8 o nil si no hay ninguno



5) Actividad 6

Dado el siguiente diccionario:

|d|

d:=Dictionary new at:'gorrión' put:'pájaro';

at:'olmo' put:'árbol';

at:'golondrina' put:'pájaro';

at:'sauce' put:'árbol';

yourself.

Investigar que retornan o qué modifican las siguientes líneas de código

rta1:=d at:'golondrina'.

rta2:=d at:'gorrión'.

rta3:=d at:'pavo'.

d at:'fuego' ifAbsent:['no lo se'].

d removeKey:'fuego' ifAbsent:['^nil].