

Arocha Juan Carlos 09-10055  
Ontiveros Mary 08-10813

Tarea SmallTalk

1. Para cada una de las siguientes expresiones de SmallTalk, responda:

Expresiones	Mensaje ejecutado	Tipo de Mensaje	Quien recibe el mensaje?	Cuántos argumentos son pasados	Cuáles argumentos son pasados
a + b	+	Binario	a	1	b
3 <= 2	<=	Binario	3	1	2
Array new: 4	new	Nombrado	Array	1	4
Set new	new	Unario	Set	0	
arr at: 4 put: 10	at: put:	Nombrado	arr	2	4 y 10
a at: 'a' add: 3	at: add:	Nombrado	a	2	'a' y 3

2. Considere las siguientes expresiones de smalltalk, Evalúe las expresiones en gst

a. 3 to: 8 do: [ :x | x printN1]

El resultado en pantalla es

3

4

5

6

7

8

3

El orden de evaluación es instanciar por el ciclo el valor actual en la x (no hay orden porque sólo es un parámetro a la vez)

- b. -3 abs factorial + 3 printNI bitAnd: 5

La salida por pantalla es:

3

1

Esto se debe a que el orden de evaluación es primero los operadores unarios (printNI es unario por eso primero muestra un '3'), y luego los binarios de izquierda a derecha (opera -3 abs factorial, lo que da como resultado 6, luego + 3 lo que da 9 y luego bitAnd: 5 lo que da 1, lo que finalmente devuelve por pantalla)

- c. a := Dictionary new.

a at: 'a' put: Set new.

a.

El output por pantalla es:

```
Dictionary (  
    'a' -> Set ()  
)
```

Lo cual se debe a que en la primera instrucción se instancia en la variable 'a' un objeto de tipo Dictionary, luego 'a' recibe el mensaje at:put: donde el argumento de at es 'a' como caracter y el argumento del put es la instanciación de un objeto de tipo Set.

- d. a at: 'a' add: 3

(a at: 'a') add: 3

El output por pantalla es:

Para la primera instrucción , el output es un error dado que como se evalúa de izquierda a derecha, 'a' esta recibiendo el mensaje at:add: lo cual no está definido.

Y para la segunda instrucción, el output es:

3

Lo cual se debe a que al agrupar (a at: 'a'), primero se evalúa ésta expresión dando como resultado el objeto de tipo Set, y luego este objeto recibe el mensaje add: el cual agrega un elemento el Conjunto.

### 3. Clase *OrderedCollection*

Esta clase representa una colección de elementos, objetos de cualquier clase, que se encuentran organizados por medio de claves y que no respetan un orden específico más que el establecido por el programador. Esta clase permite que los objetos estén duplicados y por defecto los elementos se ordenan según sean agregados o eliminados.

Funcionalidades	Mensajes
Crear una nueva colección	new (para tamaño variable) new: (para tamaño fijo)
Insertar un elemento al inicio y al final	addFirst: (inserta un elemento al inicio) addLast: (inserta un elemento al final)
Eliminar el elemento al inicio y al final	removeFirst: (elimina un elemento al inicio) removeLast: (elimina un elemento del final)
Obtener el elemento en una posición dada	at:
Cambiar el elemento en una posición dada por otra	at: put:
El primer y último elemento de la lista	first (devuelve el primer elemento) last (devuelve el último elemento)
El tamaño de la lista	size