

Investigación

Grupo 11:

Roberto Omaña 06-39990

Leopoldo Pimentel 06-40095

1)

a) $a + b$

Es un mensaje binario.

Este mensaje es recibido por un numero.

Este mensaje recibe 2 argumentos: a y b, los cuales son sumados.

b) $3 \leq 2$

Es un mensaje binario.

Este mensaje es recibido por un booleano.

Este mensaje recibe 2 argumentos: 3 y 2, los cuales son comparados.

c) Array new: 4

Es un mensaje nombrado.

Este mensaje es recibido por un Array.

Este mensaje recibe un argumento: 4, el cual define el tamaño del nuevo Array..

d) Set new

Es un mensaje unario.

Este mensaje es recibido por un objeto de tipo Set.

No recibe argumentos, ya que crea un conjunto vacío.

e) arr at: 4 put: 10

Es un mensaje nombrado.

Este mensaje es recibido por un Array de enteros.

Recibe 2 argumentos: 4, indicando la posición, y 10, el elemento a colocar en dicha posición.

f) a at: 'a' add: 3

Es un mensaje nombrado.

Este mensaje es recibido por un Array de enteros.

Recibe 2 argumentos: 'a', indicando la posición, y 3, el elemento a colocar en dicha posición.

NOTA: En Smalltalk, los caracteres son también valores enteros, como en C.

2)

a) st> 3 to: 8 do: [:x | x printNl]

Salida:

3
4
5
6
7
8
3

Esto imprime la secuencia del 3 al 8, esto se hace a través de una iteración acotada (begin to: end do: []), luego al final vuelve a imprimir el primer número.

b) st> -3 abs factorial + 3 printNl bitAnd: 5

Salida:

3
1

Este resultado es producto de la siguiente secuencia:

Primero obtiene el valor absoluto de 3, luego obtiene su factorial. El paso siguiente es sumarle 3. Con printNl, imprime el segundo número y luego el resultado de la suma, para finalmente aplicarle la operación bitWise con el 5.

c) st> a := Dictionary new.
a at: 'a' put: Set new.
a.

Salida:

Dictionary (
)
st> a at: 'a' put: Set new.
Set ()
st> a.
Dictionary (
 'a'->Set ()
)

Esto crea un Diccionario vacío, luego, a la clave 'a' le asigna como valor un conjunto vacío.

d) `st> a at: 'a' add: 3`

El compilador está tomando la operación 'add' como un parámetro del mensaje 'at', lo cual da error dado que no existe un mensaje con la firma `at: atIndex add: Element`.

`st> (a at: 'a') add: 3`

Salida:

3

Al colocar la operación entre paréntesis, esto devuelve un conjunto, lo que permite añadir un elemento a dicho conjunto, en este caso, se añade el 3.

3)

Esta clase representa colecciones ordenadas de objetos con tipos arbitrarios, los cuales no son accesibles de manera directa, sino a través de un índice.

a) Crear una nueva colección.

`a := OrderedCollection new.`

b) Insertar un elemento al inicio y al final.

`a addFirst` y `a addLast`

c) Eliminar el elemento al inicio y al final.

`a removeFirst` y `a removeLast`

d) Obtener el elemento en una posición dada.

`a at:anIndex`

e) Cambiar el elemento en una posición dada por otra.

`a at:anIndex put:anObject`

f) El primer y último elemento de la lista.

`a first` y `a last`.

g) El tamaño de la lista.

`a size`