

Programación Orientada a Objetos

Profesora Claudia Cappelletti

Smalltalk

- 1) Características de Smalltalk
- 2) Mensajes
 - a) Tipos de Mensajes
 - b) Orden de ejecución de mensajes
- 3) Variable y asignación
- 4) Estructuras de control
 - a) Selección condicional
 - b) Repetición condicional
 - c) Repetición de longitud fija
- 5) Actividad 3



Smalltalk. Características

1) Características de Smalltalk

- Es considerado el primero de los lenguajes orientados a objetos, aunque en realidad el primero en implementar la programación orientada a objetos fue Simula.
- Ha tenido gran influencia sobre otros lenguajes como **Java** o **Ruby**, y de su entorno han surgido muchas de las prácticas y herramientas de desarrollo promulgadas actualmente por las metodologías ágiles (refactorización, desarrollo incremental, desarrollo dirigido por tests, etc.).
- Es un lenguaje de programación reflexivo y con tipado dinámico.



Smalltalk. Características

- **Smalltalk** es un lenguaje **orientado a objetos puro**, todas las entidades que maneja son objetos. El lenguaje se basa en conceptos tales como objetos y mensajes.
- Es mucho más que un lenguaje de programación, es un ambiente completo de desarrollo de programas.
- Algunos de los Entornos de trabajo son: Smalltalk Express, Pharo, Squeak, VisualWorks, Dolphin.



Smalltalk. Características

- Diseñar nuevas aplicaciones Smalltalk, requiere de conocimientos sobre las clases existentes en el sistema Smalltalk.

Se dice que programar en Smalltalk es programar por extensión.

- Las nuevas aplicaciones son construidas por extensión de las librerías de clases de Smalltalk.
- El costo de aprendizaje se recupera por ser sus aplicaciones de alta productividad por el “reuso” y “extensión del código”.



Smalltalk. Mensajes

2) Mensajes

Componentes: receptor, selector, argumentos.

Tipos de mensaje: unarios, binarios y de palabra clave.

Ejemplos:

5 factorial.

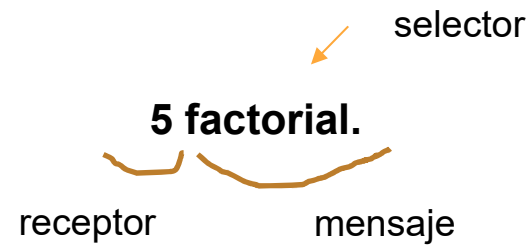
3 < 5.

#(4 3 8 1) at: 4 put: 2.



Smalltalk. Mensajes

a) **Mensajes unarios:** no tienen argumentos



Otros ejemplos:

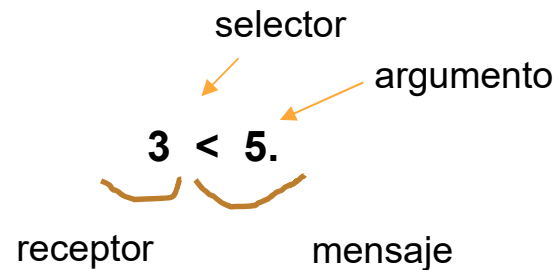
19,76 rounded.

'abcd' size.



Smalltalk. Mensajes

b) **Mensajes binarios:** tienen un solo argumento. Se utilizan para operaciones lógico, matemáticas.



Otros ejemplos:

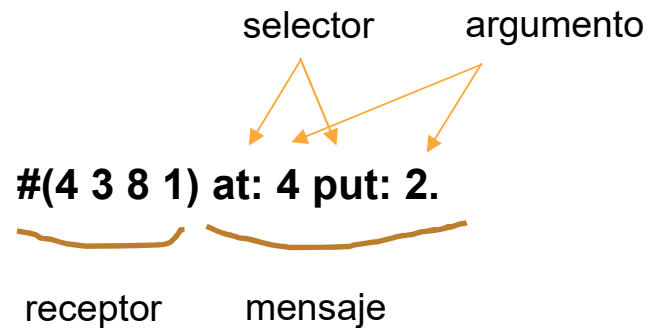
`'abc' ~= 'def'.`

`true and false.`



Smalltalk. Mensajes

c) **Mensajes de palabra clave:** son mensajes con una o más palabras clave, cada palabra clave tiene un argumento asociado. Se reconoce por los dos puntos (:).



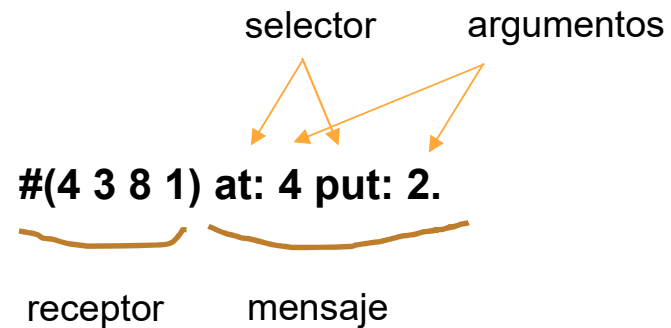
Otros ejemplos:

5 between:8 and:10.



Smalltalk. Mensajes

c) **Mensajes de palabra clave:** son mensajes con una o más palabras clave, cada palabra clave tiene un argumento asociado. Se reconoce por los dos puntos (:).



Otros ejemplos:

5 between:8 and:10.



Smalltalk. Mensajes

Orden de ejecución de los mensajes:

Los mensajes en Smalltalk se ejecutan en el siguiente orden:

1°) las expresiones que están entre paréntesis ().

2°) las expresiones unarias.

3°) las expresiones binarias.

4°) las expresiones de palabra clave.

Todas de izquierda a derecha.



Smalltalk. Mensajes

Ejemplos de mensajes anidados:

2 factorial negated.

$3 + 4 * 6 + 3$.

5 between:1 and:3 squared + 4.

Ejemplo de polimorfismo:

1) $5 + 100$.

valor de retorno= 105

2) $(200 @ 200) + 100$.

valor de retorno= $(300@300)$



Smalltalk. Variable y asignación

3) Variable y asignación

Una variable en Smalltalk representa un puntero a un objeto.

$x \leftarrow x + 1$ El mensaje +1 es enviado al objeto referido por x. La variable x luego apunta a la expresión resultado de evaluar $x+1$.

Una asignación en Smalltalk tiene el siguiente formato:

var := expresión

Ejemplo: $x := x + 1$.

Todas las sentencias en Smalltalk terminan en . , salvo algunas excepciones.



Smalltalk. Estructuras de control

4) Estructuras de control

En Smalltalk no existen las estructuras de control, se simulan, están implementadas en términos de objetos y mensajes.

a) Selección condicional:

- (expresión booleana) `ifTrue:[TrueBlock]`
`ifFalse:[FalseBlock].`
- (expresión booleana) `ifTrue:[TrueBlock].`
- (expresión booleana) `ifFalse:[FalseBlock].`

Ejemplo:

```
|a b|
a:= 3.
b:= 8.
(a < b) ifTrue:[a:=a +1]
        ifFalse:[a:= b*2].
a inspect.
```



Smalltalk. Estructuras de control

Interpretación del ifTrue: ifFalse:

Los objetos boolean true y false aceptan los mensajes de palabra clave ifTrue: ifFalse:

- 1) La expresión booleana es evaluada, dará un objeto true o false como resultado.
- 2) Si el resultado es true el mensaje ifTrue: ifFalse: con sus argumentos será enviado al objeto true, sino será enviado al objeto false.
- 3) El objeto true responde el mensaje evaluando la expresión TrueBlock.

Ejemplo:

```
|a b|  
a:= 3.  
b:= 8.  
(a < b) ifTrue:[a:=a +1]  
          ifFalse:[a:= b*2].  
a inspect.
```

Smalltalk. Estructuras de control

b) Repetición condicional:

- [expresión booleana] whileTrue:[cuerpo del loop].
- [expresión booleana] whileFalse:[cuerpo del loop].

Ejemplo:

```
|suma i|  
i:=1.  
suma:= 0.  
[i <=10] whileTrue:[suma:= suma + i.  
                    i:=i+1].  
suma inspect.
```



Smalltalk. Estructuras de control

Interpretación del whileTrue:

- 1) El mensaje whileTrue [] es enviado al bloque [expresión booleana].
- 2) En respuesta el bloque se evalúa.
- 3) Si el bloque retorna el objeto true se evalúa el cuerpo del loop y el mensaje whileTrue es nuevamente enviado al bloque [expresión booleana] y se repiten los pasos 1), 2) y 3).

Ejemplo:

```
|suma i|  
i:=1.  
suma:= 0.  
[i <=10] whileTrue:[suma:= suma + i.  
                    i:=i+1].  
suma inspect.
```

Smalltalk. Estructuras de control

c) Repetición de longitud fija:

- ValorInicial to: valorFinal do: [variable del loop | cuerpo del loop].

Interpretación del to:do:

El mensaje to: valorFinal do:unBlock evalúa el argumento unBloque para cada entero que esté en el intervalo dado por el valor del receptor hasta el valor final incluido.

Ejemplo:

```
|suma|  
suma:= 0.  
1 to: 10 do: [:i | suma:= suma + i ].  
suma inspect.
```

5) Actividad 3

1) Resolver los siguientes mensajes anidados, indicando objeto receptor, mensaje, selector y argumentos, de cada uno de los mensajes. Indicar en cada caso también el valor de retorno. Resolver paso a paso indicando el procedimiento utilizado.

a) 2 factorial negated.

b) $3 + 4 * 6 + 3$.

c) 5 between:1 and:3 squared + 4.

