Precision about the Cascade

Let's be Precise!

- The semantics of the cascade is to send all the messages in the cascade to the receiver of the FIRST message involved in the cascade.
- Workstation new name: #mac ; nextNode: aNode
- Where the msg name: is sent to the newly created instance of workstation and the msg nextNode: too.

Let's be Precise!

(OrderedCollection with: 1) add: 25; add: 35

In the example the FIRST message involved in the cascade is the first add: msg and not #with: . So all the messages are sent to the result of the parenthesised expression, the newly created instance of anOrderedCollection

One Problem

(OrderedCollection with: 1)
  add: 25;
  add: 35

Prt-> 35

One problem: the expression returns 35 and not the collection object.
Let's analyze a bit...

OrderedCollection>>add: newObject
"Include newObject as one of the receiver's elements. Answer
newObject."

^self addLast: newObject

OrderedCollection>>addLast: newObject
"Add newObject to the end of the receiver. Answer
newObject."

lastIndex = self basicSize ifTrue: [self makeRoomAtLast].
lastIndex := lastIndex + 1.
self basicAt: lastIndex put: newObject.
^newObject

Really got it?

yourself returns the receiver of the cascade:

Here the receiver of the cascade is a newly created
instance anOrderedCollection and not the class
OrderedCollection. The self in the yourself method is
linked to this instance

(OrderedCollection with: 1) add: 25; add: 35; yourself
-> OrderedCollection(1 25 35)

Of course!

Object>>yourself

^ self