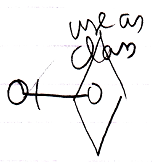
|  |
| --- |
| Circle Language Spec: System Objects |

## Explicit Get & Set Arguments Notation

This article sheds light on *explicit Get & Set argument* notations. Several notations are introduced. It is not clear yet which is the best one.

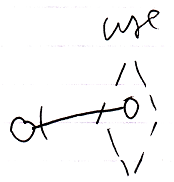
In the diagrams below things are often implicitly notated. However, the notation of the Get and Set arguments is always *ex*plicit.

The most explicit way to draw out a call to a system command with the argument explicitly notated is as follows:



It is notated as calling the overloaded Use As Class command with its argument filled in.

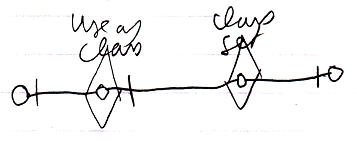
But this notation is also possible:



Then you express the aspect *implicitly* using a different line type for the diamond. You also leave out the aspect from the identifier.

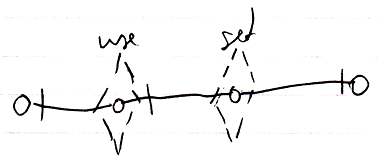
So the aspect can be expressed with a diamond line type or by explicitly expressing the aspect in the identifier.

Get, Set and Use commands are never used on their own. They always need another counterpart. The notation below shows two system command counterparts with explicit arguments:



The Class Set argument is connected to the Use As Class argument.

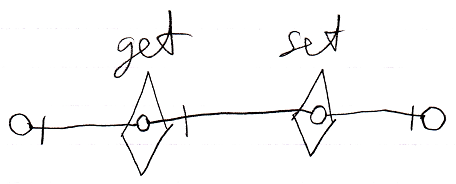
The following notation expresses the same thing:



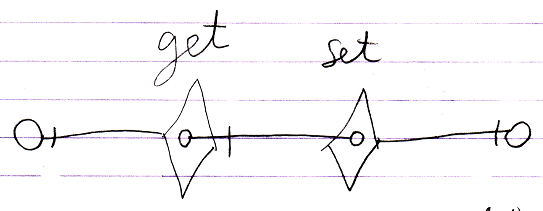
But now the aspect is implicitly expressed by the line type of the command calls.

Note that all explicit Get and Set argument notations use a solid *object* line between arguments, because that is the part that is *explicit*. Other parts of the notation are implicit sometimes.

If only the object aspect is adressed, the picture looks like this:



There is also the following weirder possibility to display the Get and Set arguments:



But the odd thing about that notation is that there is no connection between the Get and Set arguments and the real Get and Set targets. The notation looks so strange, that we will probably never go with that.

The main point is: the diagrams above show the explicit expression of the arguments of Get and Set calls.

Explicit display of Get and Set arguments is hardly ever used, because there are simpler notations available, such as the assignment notation or if you must, the explicit Get & Set notation:

|  |
| --- |
| Assignment |
|  |
| Object Get 🡨  Object Set 🡪 |
|  |
| Explicit Get & Set |
|  |
| Object Get 🡨  Object Set 🡪 |
|  |
| Explicit Get & Set Arguments |
|  |
| Object Get 🡨  Object Set 🡪 |

You would only use explicit display of Get and Set arguments in abstract diagram expression when you just want to make a point.