|  |
| --- |
| Circle Language Spec: Black Boxes |

## Public & Private Connectors

The following articles display the various access connectors an object can have.

#### Object Connectors

|  |  |  |
| --- | --- | --- |
| Public Object Get | Private Object Get | Friend Object Get |
|  |  |  |
|  |  |  |
| Public Object Set | Private Object Set | Friend Object Set |
|  |  |  |

#### Class Connectors

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Public Use As Class | | Private Use As Class | | Friend Use As Class | |
|  | |  | |  | |
|  | |  | |  | |
| Public Class Set | | Private Class Set | | Friend Class Set | |
|  | |  | |  | |
|  | |  | |  | |
| Public Reference Class Get | | Private Reference Class Get | | Friend Reference  Class Get | |
|  | |  | |  | |
|  | |  | |  | |
| Public Object Class Get | Private Object Class Get | | Friend Object Class Get | |
|  |  | |  | |

#### Value Connectors

|  |  |  |
| --- | --- | --- |
| Public Value Get | Private Value Get | Friend Value Get |
|  |  |  |
|  |  |  |
| Public Value Set | Private Value Set | Friend Value Set |
|  |  |  |

#### Data Connectors

|  |  |  |
| --- | --- | --- |
| Public Data Get | Private Data Get | Friend Data Get |
|  |  |  |
|  |  |  |
| Public Data Set | Private Data Set | Friend Data Set |
|  |  |  |

#### Clone Connectors

|  |  |  |
| --- | --- | --- |
| Public Clone (2) Get | Private Clone (2) Get | Friend Clone (2) Get |
|  |  |  |
|  |  |  |
| Public Clone (2) Set | Private Clone (2) Set | Friend Clone (2) Set |
|  |  |  |

#### Existance Connectors

|  |  |  |
| --- | --- | --- |
| Public New | Private New | Friend New |
|  |  |  |
|  |  |  |
| Public Annul | Private Annul | Friend Annul |
|  |  |  |

#### Execute Connectors

|  |  |  |
| --- | --- | --- |
| Public Execute | Private Execute | Friend Execute |
|  |  |  |

#### Add And Remove Connectors …

…

#### Remarks

Object-Bound & Reference Bound

In the overviews, object-bound connectors are displayed with a separate shape without a parent, while reference-bound connectors are displayed with the target inside a parent.

Object Class Connectors

There is no Object Class Set connector, because the object class can not be freely set; it can only be set on object creation.

Class Get and Use As Class

For the Class aspect, there is a distinction between Use As Class and Class Get. They are different. Use As Class makes you use one symbol as the class for another symbol. That is how classes are most commonly used. That would be the result of an Object Get on the source object, followed by a Class Set on the target object. No Class Get is involved in a Use As Class action, which you might expect if you draw a parallel to the Object aspect.

Class Get gets a reference to the class object of an object. Traditionally this would be considered getting an object’s reflective data. You can use the reference to the class to your own discretion. Class Get is far less common. Use As Class gets a fundamental notation, whereas Class Get has to make dues with the ‘system command’ notation.

This is due to the behavior of the Class aspect. This is different behavior compared to the Object aspect. You just have to remember, that each aspect has different behavior, because it has a different purpose.

The funny thing about access controlling Use As Class, is that you are actually access controlling the Object Get for specific use in a class assignment.

Value Connectors

Note that for the Value aspect the directions of Get and Set are switched compared to the Object aspect

|  |  |
| --- | --- |
| Object Get | Value Get |
|  |  |

This is because for the Object aspect, the access mark indicates the direction of the *resultant line*, while for the Value aspect, the access mark indicates the direction of the *assignment*. You have to remember, that each aspect has its own unique behavior.

Data Connectors

You can access control sub-objects, to apply some sort of write-protection to the sub-object. But that does not write-protect the sub-objects of the sub-objects. To collectively write-protect any sub-object, you can access control the object’s Data aspect.



A preliminary notation for Data access is a *triple wavy line* (‘triple’ expressing multitude, ‘wavy’ expressing values or data).

Clone Connectors

The Value aspect is about copying the direct value of the object. But you can also copy the values of sub-objects up to a certain cloning depth. The could be any number of levels up to which the values of sub-objects are cloned. The diagrams in the overview show access connectors for cloning at a depth of 2. That is just an example. Any cloning depth other than two can also be separately access controlled.

Cloning has the exact same notation as the Value aspect, but then with a number near the access mark, to indicate the cloning depth.

|  |  |
| --- | --- |
| Value Get | Clone (2) Get |
|  |  |

Clone is simply thatclosely related to the Value aspect.

Interface Connectors & Other Connectors

There are also interface connectors, but *interfaces* as a whole will be cought up with in the *Interfaces* article group. Interface connectors work roughly the same as class connectors. In fact, any other aspect that will be introduced in the future will extend the set of possible connectors.

Preliminary System Command Notation

The access connectors using the system command notation have a preliminary notation. The exact notation system commands will get an update in the future.