|  |
| --- |
| Circle Language Spec: Parameters |

## Parameter Passing

*Object*, *class*, *value* and *execution* are called system aspects of a symbol. The basic ones are covered in the articles *System Objects*.

Other programming languages have *parameter passings*. In the new computer language parameter passing is established by access controlling the system aspects of a parameter.

It is basically about being able to separately access control every *system command*.

Just as an example of how access controlling system commands can be seen as parameter passing we consider the ByRef parameter passing of the programming language *Basic*. A ByRef parameter passing’s object target can be set from the outside, and the value of it can be read and written on the outside and on the inside. You make it point to something, and you read and write its value.

Basically the following system commands are accessible:

Object Set Public

Value Get Private

Value Set Private

For a ByRef parameter it is not usual, that you read out the object aspect: meaning you do not *point to* a ByRef parameter, you just *make it point at* something. So the Object Get (‘point at’) command is inaccessible. ByRef parameters are also not Object Set on the inside. So Object Set Private is also inaccessible. So Public does not means that you automatically have Private access as well.

Which system commands are practical to access control, in order to establish the idea of parameter passing, is not that important. The most important thing is, that if you have access control over system commands, you can establish any known parameter passing.

Parameter passing can also be called *parameter access control*.

Examples of practical parameter access control may be covered later.