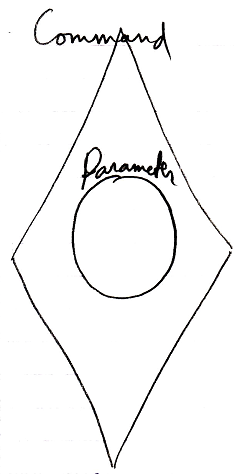
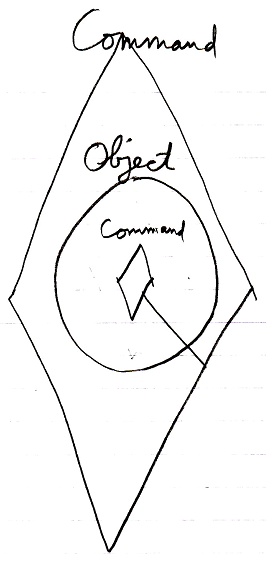
|  |
| --- |
| Circle Language Spec: Parameters |

## Relations Between Commands & Objects in a Diagram

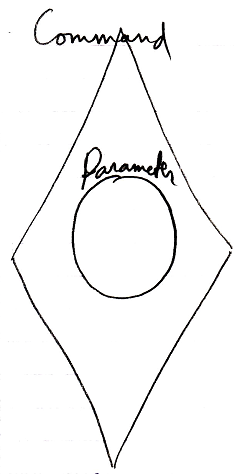
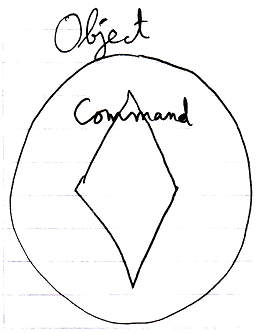
A parameter is an object related to a command.



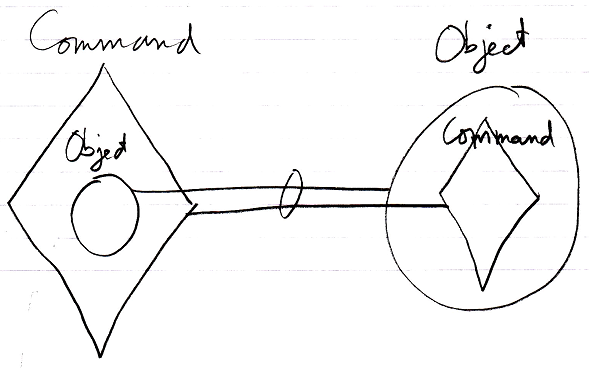
The relation is bidirectional. When an object is a parameter, then the object also contains a reference back to the command.



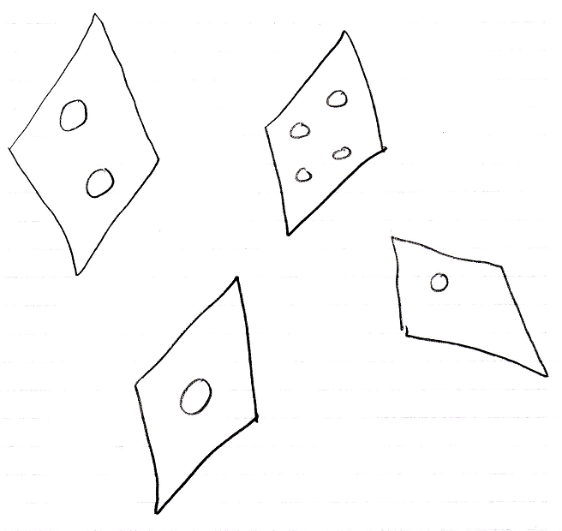
There is complete exchangeability between a command’s parameter and an object’s command.

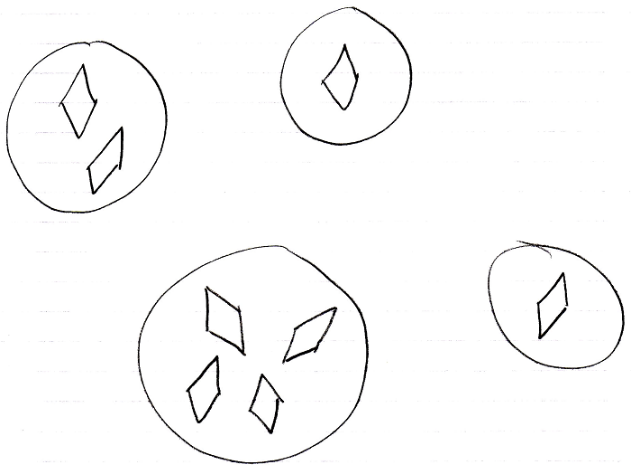
Commands are special objects, that just happen to be executable. That is why the concept of relations applied to commands and objects, result in the concept of parameters.



You can view a system as being a flat set of procedures, that take objects as parameters.

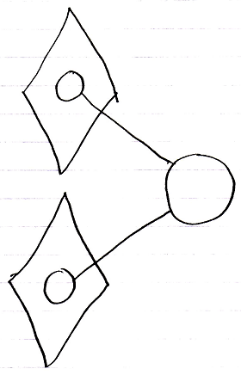


You can also view a system as being a set of objects, that have a set of commands inside of them them.

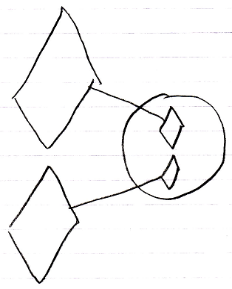


Procedural and object oriented are completely interchangable in this system.

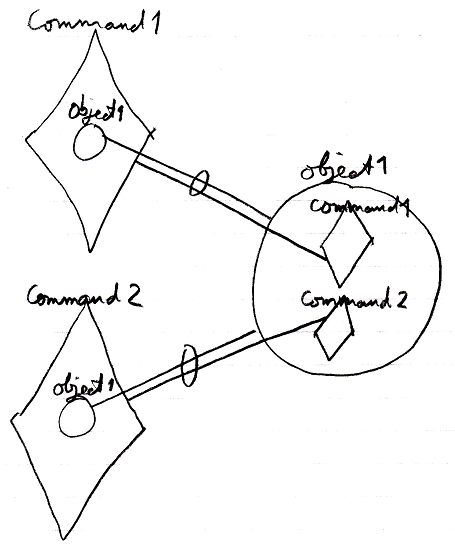
Multiple commands can have the same object as a parameter.



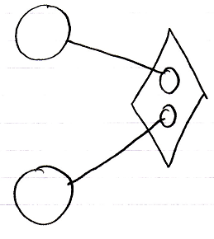
Then the object will contain all of those commands.



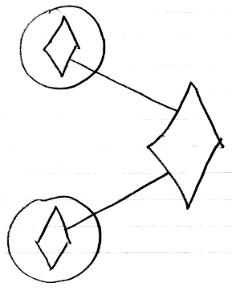
These two perspectives on it are completely interchangable.



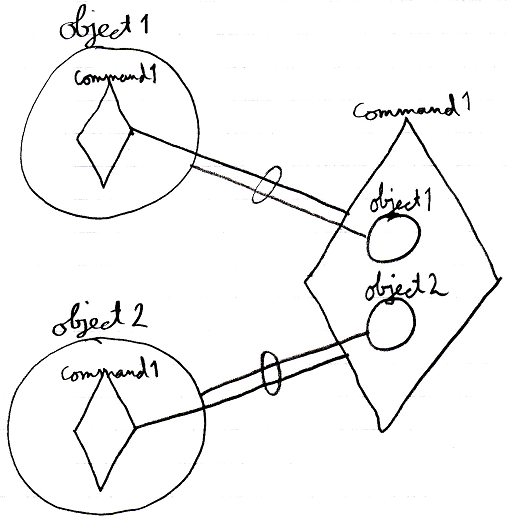
A command can contain multiple objects.



Then the command becomes part of all of those objects.



These two perspectives on it are completely interchangable.



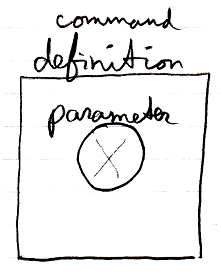
In this new computer language both points of view, commands being part of objects, and commands being independent entities with parameters, are equally visible in the language.

### Object relations

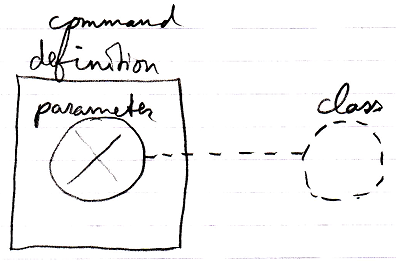
So when you’ve passed an object to a command, the command automatically becomes visible inside the object as a runnable command.

### Class relations

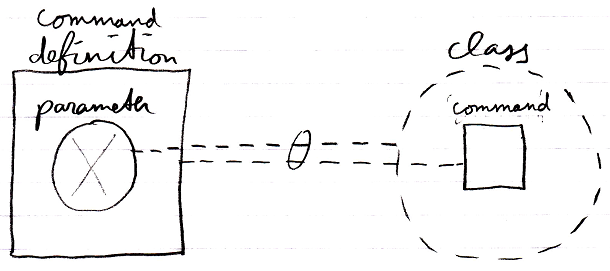
But in case of a command definition, a parameter is usually not filled in yet.



But the class of the parameter is determined.



It is a relation between classes, instead of a relation between objects. This automatically makes the command definition available from the parameter’s class.



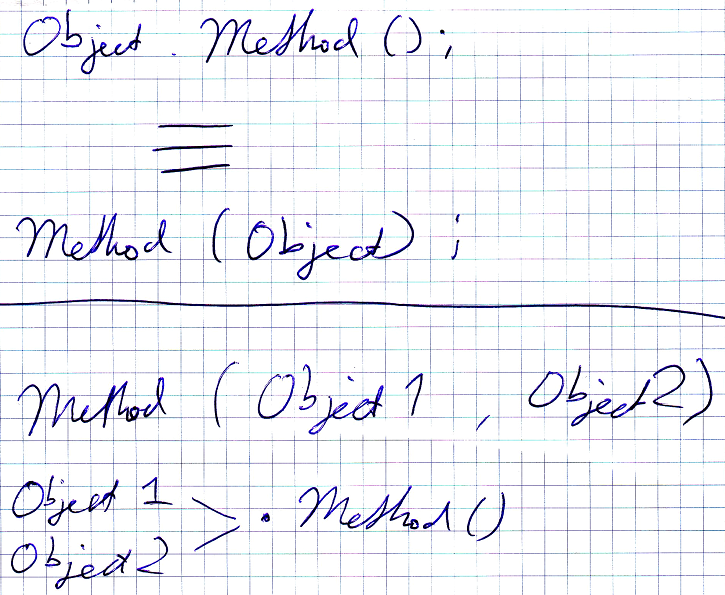
There is complete exchangability between the class of a parameter and a command of a class.

## Ideas

Commands,

± 2008

The idea of multi-methods and exchangeability between class methods and method parameters is the expressed in textual pseudo-code below:



JJ