|  |
| --- |
| Circle Language Spec: Interfaces |

## Extensive Classification with Interfaces (Unfinished)

This article is not finished.

Extensive classification through interfaces is about automatic definition of interfaces, as well as manual definition of interfaces. (Consider the imaginary group-by-source interfaces, which could be suggestions for interfaces to form.).

Perhaps there will be other ways interfaces will be automatically defined.

Such as mutuality of commands creates mutuality of interfaces.

A grouping symbol based on what a dependent object uses. The interface organization is a little bit automatic that way. If you do not have an interface organization, then you can still see what the various clients are using and perhaps for the overview you may use an interface organization after all.

## Ideas

- Extensive classification through interfaces

- Each attribute or method can be made an interface

- Etcetera.

- Semantic classification as such.

### Extensive classification through interfaces

You should use interfaces more, to be able to let a new class participate in existing

things, such as the ICollection provides.

Then, for instance, you can program classes that can then be used in J Sound as an operator. That's cool.

I never really organized a system as such, but thinking in that way, does lead to

extensibility.

Perhaps several object classes could have physically the same attribute, so that an attribute can belong to multiple classes. Perhaps that should be part of inheritance, though. That would be less uncontrolled. (For instance the BackColor attribute can be part of multiple classes. Perhaps BackColor should be something they all centrally relate to.)