Overloading

- Overloading names of related classes, disambiguating by class.

## Ideas

*These were moved here from the Projects documentation, to the System documentation, to integrate them with the System documentation. Remaining ideas should be moved back to the Projects documentation after that. The Projects documentation is where Ideas belong.*

Overloading

Als je eindigt met het data type te noemen, krijg je zo'n notatie:

Stop Button

Button is dan de class.

Je kunt het ook zo declareren:

Stop Button Button

Dan is Stop Button de variable identifier en Button de Class,

maar bij:

Stop Button

is Stop de variable identifier en Button de Class. De button heet dus Stop. Maar je zult meer dingen kunnen hebben die Stop heten. Bijvoorbeeld:

Stop Procedure

Als je na de identifier het type benoemt, kun je daarmee wel specifiek identificeren welke Stop je bedoelt.

Zo lok je uit het type altijd te vermelden aan het eind.

En is de declaratie slechts de identificatie zoals het menselijk ook dat wil zien, en is daarbij de class ook toegewezen.

JJ

## Out of the original Symbol documentation

In J Code you could do overloading in which anything of the declaration can differ. If you use named arguments than you can do same parameter types. You can link to any symbol anyway, so it would be just a matter of what you see in text code.