|  |
| --- |
| Circle Language Spec: Object Order |

## Object Order

**[Preliminary documentation]**

This is unfinished documentation of lower quality. It only has unfinished descriptions and loose ideas.

### Object Order

*This brainstorm was written in the context of efforts to program a prototype app 'Circle 3'.*

This section is here to augment the functional design of the object order concept. Object order is not a difficult concept. If objects have a certain sequence order, there are several basic data structures facilitating putting the objects in a certain order. That is not the problem here. The problem with object order is the notation. I recall that several notations were proposed, but no definite one was chosen. Actually Object Order is not a problem. It is not like there is no solution for it, it is just that there are too many solutions / too many possible notations. It would indeed be nice to determine the perfect notation.

Ordering is positioning, so the best way of expressing ordering objects would seem to position them in a certain order. The diagram notation is all about expressing systematics with the their most direct depiction possible, so I think the only possible way to express object order is through positioning. This may conflict with automatic positioning of diagram elements. So … then you have to see if it does.

A strategy you could have is to ignore the expression of object order for now, and pick one later when the rest is up and running. You can have to base up and running and ignore the expression of this concept. So the concept is implemented, but the expression is not. Yet another limitation to impose onto the first version of Circle 3.

### Order Matters

*This seems documentation about Creator 0.9: an experimental app, a code generator, in which a program might be based on classes and their relations.*

This is a value you can set to **True** in a 🡪 n relation, that should aid in picking the right data structure for the relation. The only thing it was come up for, was to base picking either hole list or linked list on variables that had no technical meaning, but only functional meaning. I assumed that there will be more list structures that can be chosen if **OrderMatters** or **Not**.

That’s all there is to say about it for now.

The **OrderMatters** properties are not used by J Data Generator.

Apart from **RelationClass**, **Class** also has an **OrderMatters** property intended to set **OrderMatters** for all **RelationClasses** of that **Class**.

### Class . OrderMatters

See the *Order Matters* section.

### Ideas

Object Order,

2008-09-30

Perhaps anything ordered, be it parameters, needs to be part of an ordered list, an ordered array, or otherwise you can't order anything. And perhaps the display of an ordered array is the only display of anything ordered, including the way it would be displayed in text code.

JJ

-----

Object Order,

2009-05-15

There are several candidates for a diagram notation for order. They are out there somewhere among the ideas. Candidates:

- order in a circle, starting at the top, going clockwise, and if there are too many, scroll buttons < >, that make you rotate around the items.

- The items are linked to eachother, like next-command references (look up the idea somewhere.)

JJ