# Circle Language Spec Inheritance Spec Strategy & Steps

***-***

*Author: JJ van Zon*

*Location: Oosterhout, The Netherlands*

*Date: May 18, 2010 – July 16, 2010*

*Status: Finished*

## Purpose of this Document

This document is not meant as reading material. It is a list of steps undertaken in the project, later serving as a reference for how the project was executed.

## Strategy

- Improvise

- Use same strategy as the previous project and see if it works.

- Also consider saving the hardest topics for later.

- The inheritance documentation does something with object resolution topics.  
First it was thought that object resolution might also be finished up in this project, but this idea was let go of.

- The next project will be about working towards an implementation phase as soon as possible.

## Steps

- Short Brainstorm

- Read over strategy from previous project

- Compose topic list

- Start making a topic list off the top of your head

- Extend topic list going through the short existing texts about inheritance

- Look at object resolution concepts for Overriding, Shadowing, etcetera

- Extend topic list going through the long existing texts about inheritance.

- Look at Black Boxing for Exclusion topics

- Brainstorm about unclarities

- Finish up topic list

- Read over ‘writing efficiently’

- Finish up topic list

- Start writing already

- Main Concepts

- Work out the List concept:

> In the System Objects chapter

A list’s control over aspects of its items & adressing the list as a normal object should go through its system interface, because that is the internal working of the list (the system object).

- Specialization:

- Altering the Member Set

- Detouring Members

- Altering Command Implementations

- Miscellaneous Situations

- Make Notation for ‘Required’

> In the Parameters chapter.

X Enforcing & Preventing Specialization

… Deeper Specialization

- Change approach:

- Remove intrinsic overriding notation

- Add data replacement versus specialization

- More work:

- Reorient in the plan

- Read the articles

- Read the brainstorms

- Make evident changes, that are mentioned in brainstorm text

- Add missing stuff to articles, such as missing pictures

> Split up the Specialization article already

> Changed the Interface section of Enforcing & Preventing Specialization

> I also have an idea for specifying rules that apply to all sub-objects. Maybe that can be derived from the list concept.

- Set up open issue list

- Add one Miscellaneous Inheritance Situation:

- One object inheriting multiply from the same class.

- Propose solutions for open issues & adapt articles to them

- Deeper Specialization Structure

- Deeper Member Addition

- Deeper Specialization is now finished

- Other open issues are postponed

… Write Misc Deeper Specialization Situations

- Deeper Additional Members in an Object - List - Object – Object situation

- Change System Inheritance:

It used to be normal object and class inheritance, but now it is changed so that you can specialize any deeper object, so that you can specialize as deep into the hierarchy as you want.

- Something to consider though, is that when a much deeper object is specialized, all the levels that lead to the deeper object will also get a specialized version.

> Add deeper member addition example with a triangle in each layer in it

- In Misc Deeper Specialization Situations

> Removed this article degrading it to an idea

- And in Deeper Specialization, Deeper Member Addition.

- And in System Inheritance

> Left a remark in it about specializing each layer, and referred to the Deeper Specialization article.

- Postpone the other deeper specialization situations

- Wrap up:

- Change reference to Deeper Specialization in System Inheritance to a reference to Even Deeper Member Addition.

- Split up into more articles

> I create a lot more articles this way

> I split up the following:

- Main Concepts

- Altering the Member Set

- Altering Command Implementations

> but I did not split up the following ones:

Enforcing & Preventing Specialization

Deeper Specialization

- Cross out

- Ideas document in chapter

- Ideas in this project documentation

- Consider Creator 0.9 concepts

> They mostly have to do with generic user interfaces. I will not cover that here.

- Read over

> Only did half of that because I want to move on.

- And finish up project

- Tidy up software development plan documents

- Isolate version a version

- Make entry in project list tidy

- Update article in parent folder