### Cascading Overriding

J Graphics was the program that made me write the first functional piece of J Data Generator. The thing that seemed hardest to generate is the cascading overriding of the styles of J Graphics objects. Actually, it never *got* generated.

The basic idea is the following. A Line object would contain two Points. The Point objects have their own style. The Line can choose to override the style of the Point objects. The point would then get the PointStyle specified in the Line object. A Polygon object consists of Line objects. The Polygon object specifies a LineStyle and a PointStyle, which can override the Styles of the Line objects.

Actually it was even more refined. Each value of the style, for instance the ForeColor, BorderColor, FontName or FontSize, could be individually overridden.

Dang, does not seem that simple. Hmmm, what to do with that.

The Point object’s styles would actually return the values as overridden.

#### Old Brainstorm

* Styles (= overriding dedimensionality (het ergste van allemaal))
* A parent overriding default values of sub objects

I want Overriding Styles to be generated by Sigma Data

Overriders are private collections in which you can publicly register or unregister.

Bij styles moet je:

- per soort style een klasse hebben

- zeggen dat die style klasse z'n ValuesOverridable zijn

- Een klasse die een directe relatie heeft met een style, krijgt dan automatisch   
 een overriders collectie

- Per klasse moet je kunnen zeggen of ie een overrider is

- Met dat de klasse een overrider is is ook de collectie van die klasse een   
 overrider