

Re: Diamond back refactoring (again, I know...)

Source: <http://coding.derkeiler.com/Archive/Delphi/borland.public.delphi.non-technical/2004-09/4482.html>

From: Jim Cooper (jim.cooper_at_virgin.net)

Date: 09/24/04

Date: Fri, 24 Sep 2004 08:42:34 +0100

> *Especially when writing libraries renaming in a non local scope is more
> damaging than useful.*

I'm sure I've missed your point here. Why is that so? The Rename refactoring only renames what it should. It's not the old search and replace, but a smart, context-aware version. AFAIK it won't cross project boundaries though (or maybe project group boundaries if you have them)

Or are you talking about changing the public interface? In that case, then yes, but that's an issue for you no matter what refactoring you do, either by hand or mechanically. There are a number of issues with refactoring generally. That's one, the other big one for some people is that the Fowler refactorings are not thread-safe (necessarily). There are others.

> *Extracting a method will by default occur on a
> local scope and be of no harm in that context.*

The extracted method is made protected, if that's what you mean. Although adding to a public interface is not necessarily bad.

> *The extract ancestor class should be safe though.*

I must confess I didn't see the difference between that and Extract Superclass :-)

> *Even so I must say I find renaming quite useful, in a non library
> context, since classes are merely specialized fragments of a greater
> whole and thereby easily subject to change (if just by name).*

Yes, it is useful, and it works on much more than class names too.

Cheers,
Jim Cooper

borland.public.delphi.non-technical: Re: Diamond back refactoring (again, I know...)

Jim Cooper jim@falafelsoft.com

Falafel Software <http://www.falafelsoft.co.uk>
