

Re: what's the future of Object Oriented Programming

Source: <http://coding.derkeiler.com/Archive/General/comp.object/2006-09/msg00174.html>

- *From:* "topmind" <topmind@xxxxxxxxxxxxxxxxxxxx>
 - *Date:* 12 Sep 2006 10:16:36 -0700
-

Patrick May wrote:

"topmind" <topmind@xxxxxxxxxxxxxxxxxxxx> writes:

H. S. Lahman wrote:

Reuse is nice but not a major consideration in using OO techniques.

The real objective of OO development is to create maintainable applications in the face of volatile requirements over the product life cycle.

To answer your title question, OO programming is still the only game in town for producing maintainable software.

I beg to differ. There is no evidence for this, at least outside of systems software.

Actually, there is considerable evidence that some of the concepts typically grouped together under the term "object oriented" provide significant benefits in managing dependencies between software components, which results in more maintainable software.

The only examples I've seen are from people who didn't know how to use procedural and databases well.

I don't know of anyone who has used OO in a large project and regretted having those tools available.

Re: what's the future of Object Oriented Programming

I've heard mumblings of things not going so smooth. Satisfaction surveys by Yourdon showed no improvement. Further, OOP tends to *create* large projects. Procedural/relational projects tend to break big problems into smaller applications that feed off of databases. The database is the Nile and small apps are villages on the shore. Thus, one mostly only has to know about the schemas and a few sanctioned library routines.

"The only game in town" is overstating the case a bit, because similar techniques can be applied in most non-OO languages and functional languages, for example, offer still other approaches.

Out of curiosity, what do you mean by "systems software?"

Tool building, such as OS's, RDBMS engines, compilers, device drivers, network engines, etc. Polymorphism seems to have more use there.

Sincerely,

Patrick

-T-

.