

## Re: Yet another OO question...

---

*Source:* <http://coding.derkeiler.com/Archive/Java/comp.lang.java.programmer/2007-03/msg03118.html>

---

- *From:* "Adam Maass" <[adam.nospam.maass@xxxxxxxxxxx](mailto:adam.nospam.maass@xxxxxxxxxxx)>
  - *Date:* Tue, 27 Mar 2007 00:08:49 -0700
- 

"ChrisW" <[c.c.wood@xxxxxxxxxx](mailto:c.c.wood@xxxxxxxxxx)> wrote :

This might seem like a really stupid question, but I've realised that one of the main problems I've got understanding OO (and teaching myself) is that I don't actually know what to put into different methods. I'm relatively happy doing basic procedural stuff, and I've got a ~300 line program which I'm trying to make OO, but I don't know where to start because I don't know where to split my program into different methods (if that makes sense...). I'm about a third of the way through Thinking In Java, and it's a lot better than most of the other books I've read, but until I understand \*what\* to actually put into a method I guess I'm not going to get very far! If anybody's got any suggestions where I can get more info, or people can explain how they got through this (surely everyone must go through this mind-set when they start OO stuff, or do some people just get it naturally?!). I can explain the program I'm trying to convert and / or post some pseudo code if that helps,

TIA,  
Chris

Umm... some general pointers:

- 1) First examine your data items, and do something akin to a database normalization on them. Each entity becomes a class; the relations between entities become either references to instances or collections of instances. (Realize also that m:n relationships are possible.)
- 2) Start writing methods. Methods belong to the class that contain the majority of the data necessary to complete the computation.
- 3) Tell, don't ask. Tell an object to complete some computation and return the result. Don't ask an object for its data and complete the computation outside of that object.
- 4) Talk only to yourself and your neighbors. That is, the only values that should be (directly) used in any method are the instance (and static) members of the class you're operating in, and the parameters to the method. In particular, do not call methods on the return values of methods you call in your method.

Re: Yet another OO question...

5) Keep your methods short and to the point.

6) Beware of adding too many methods to a class; a class that has lots of methods for different purposes has lost cohesion and is begging for a refactor. Try to keep all of the members of a class focused on a single purpose.

I hope this is helpful!

.