

## Re: OOP – a question about database access

**Source:** <http://coding.derkeiler.com/Archive/General/comp.object/2003-11/0247.html>

---

**From:** Uncle Bob (Robert C. Martin) (*u.n.c.l.e.b.o.b\_at\_objectmentor.com*)

**Date:** 11/06/03

Date: Thu, 06 Nov 2003 11:37:46 -0600

Tim <nocomment@rogers.com> might (or might not) have written this on (or about) Fri, 31 Oct 2003 18:49:07 GMT, :

>If this is the wrong newsgroup, can someone please point me to the  
>correct one and disregard this.  
>  
>I've been programming for 20 years but I'm trying to understand OOA and  
>OOD now. What I've decided to do is try to take a little database I have  
>that includes a table with a sports league's schedule and try to use  
>this to develop a little application to help me understand the concepts  
>and proper techniques of OOA, OOD and OOP.  
>The columns in the table are date, home team, visitor team and starting  
>time.  
>(I'm going to use Java in a JDeveloper environment and the database is  
>an MSAccess using a JDBC/ODBC bridge)  
>  
>So, how would you divide the various functions regarding interfaces,  
>inheritance, etc.

First determine what this system has to do. Break this down into very small tasks. (Add team, delete team, generate schedule report, generate league statistics, add game outcome, etc, etc.)

Completely ignore the database. Don't think about it. There are no tables, there are no rows, there is no SQL. Instead, implement each feature, one at a time without any database. Use data-structures in RAM instead. As you implement a feature, try to let the code tell you what kind of data structure it wants. Create that kind of data structure.

Keep adding new features, one at a time. Scan the code for duplication, and eliminate it wherever it occurs. Use inheritance, polymorphism, and delegation to help you eliminate this duplication. Extend this elimination to stretches of code that are similar, though not truly duplicate. Find abstractions that allow you to merge similar modules into base classes with the deviations in subclasses.

comp.object: Re: OOP – a question about database access

Keep adding feature after feature, one at a time. Don't be afraid to make sweeping changes to the design. Write enough unit tests so that whenever you make a change you can be sure you haven't broken any other part of the system. Keep the code as clean and expressive as you can.

After you have all the features implemented, then find a way to store the data structures into a database. Keep the database code as far away from the rest of the code as possible.

Robert C. Martin | "Uncle Bob"  
Object Mentor Inc. | [unclebob @ objectmentor . com](mailto:unclebob@objectmentor.com)  
501 N. Riverside Dr. | Tel: (800) 338-6716  
Suite 206 | Fax: (847) 775-8174 | [www.objectmentor.com](http://www.objectmentor.com)  
| | [www.XProgramming.com](http://www.XProgramming.com)  
Gurnee, IL, | Training and Mentoring | [www.junit.org](http://www.junit.org)  
60031 | OO, XP, Agile, C++, Java, C# | <http://fitnesse.org>