

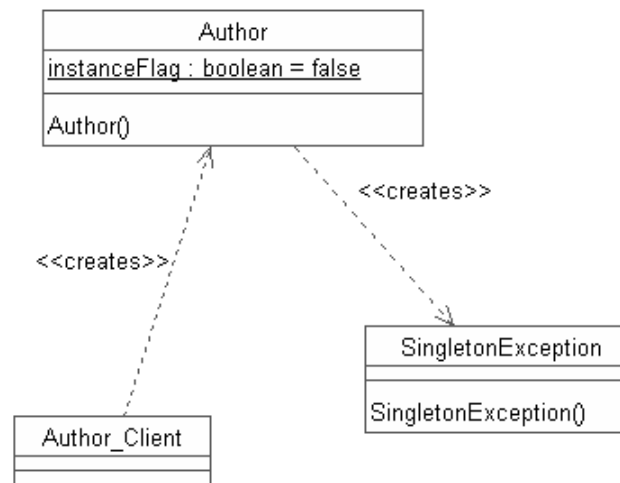
Username: University of Illinois Chicago **Book:** UML for Mere Mortals®. No part of any chapter or book may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher of the book or chapter. Redistribution or other use that violates the fair use privilege under U.S. copyright laws (see 17 USC107) or that otherwise violates these Terms of Service is strictly prohibited. Violators will be prosecuted to the full extent of U.S. Federal and Massachusetts laws.

Architectural Patterns

The UML can also be used to depict architectural patterns. Patterns are repeatable architecture designs that have been tested and proven over time. Some people might say that *every* design has a pattern, similarly to saying every design has an architecture, but we are talking about patterns that are well-defined, tested, proven, and often used to improve designs and architectures. Patterns can be used for both logical and physical architectural designs.

Many well-known patterns are described in books that go into deep details on their usage. Some of the most popular patterns in the industry are known as the Gang of Four (GoF) patterns. These patterns were developed and made popular by four gentleman, Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides, in their book *Design Patterns: Elements of Reusable Object-Oriented Software*. The patterns described in that book are broken into different categories—Creational, Structural, and Behavioral. [Figure 4-17](#) is a UML representation of one of the most popular GoF creational patterns called the *singleton*.

Figure 4-17. Singleton pattern.

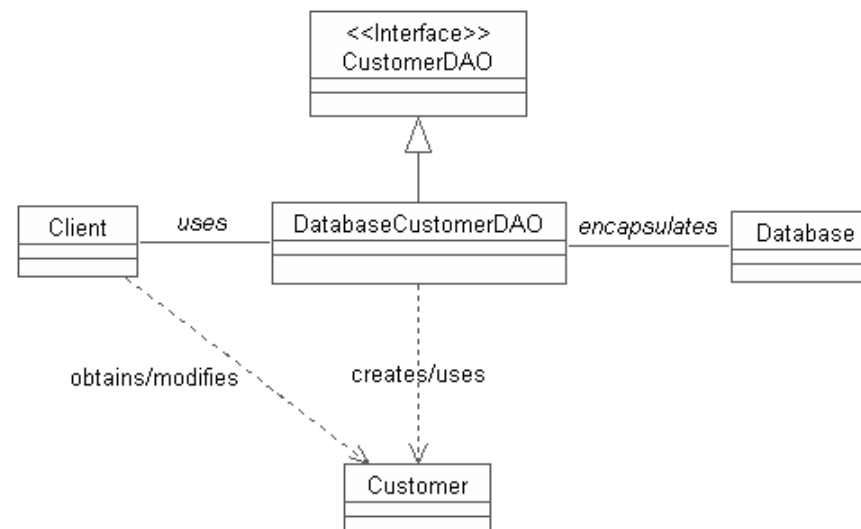


The singleton pattern is applied against an individual class, and it creates several classes in a pattern. As you saw in [Figure 4-17](#), the pattern is applied to a class named Author, and it creates two additional classes including some attributes and operations as part of the pattern. The intent of the singleton pattern is to ensure that a class only has one instance and to provide a global access point to it. [GAMM1] This pattern is quite logical in nature, but it can also be applied using additional methods with specific languages to become more physical.

There are also several patterns that are specifically physical in nature. An example of some physical patterns were created by a group of engineers at Sun Microsystems called the “Core J2EE Patterns” and are described in great detail in a book with the same title. Here, we will look at a pattern known as the Data Access Object (DAO) pattern.

An example DAO for a persistent object that represents Customer information is shown in [Figure 4-18](#). The CustomerDAO creates a Customer value object. [ALUR1]

Figure 4-18. DAO J2EE pattern.



Although the DAO pattern might not be self-explanatory to a mere mortal, it is used in physical architectural design to provide database access in J2EE applications. Just like the logical architectural patterns, physical patterns are applied to existing classes and model elements to create additional elements using standard and proven processes.