Figure 4.16   *The Publication class and three subclasses*

A generalization/inheritance relationship can be identified:

- If phrases such as 'is-a' or 'is-a-kind-of' can be used to describe the relationship between classes, e.g. a journal is a-kind-of publication, a horse is-a mammal

- Where one or more classes have very similar attributes and operations and the introduction of a general class would simplify the model.

Theoretically, there is no limit to the number of levels allowed in an inheritance hierarchy; in practice, it has been found that a hierarchy with more than about six levels becomes unmanageable.

Generalization and inheritance are useful techniques because, as we have seen, they allow reuse of existing classes. Classifying classes into an inheritance hierarchy also means that we avoid repeating code. Inherited operations reside in the superclass; subclasses do not need to carry around their own version, unless they are going to specialize the operation (see polymorphism below). This means if we alter the code that implements an operation we need only alter it once, in the superclass.

Classifying classes into an inheritance hierarchy also helps organise and simplify our understanding of the classes in the system. Classes suitable for such classification have both similarities and differences; an inheritance hierarchy emphasizes both. To avoid unnecessary repetition and clarify our understanding of the classes, attributes and operations are defined at the highest applicable level in the hierarchy. Differences are reflected in attributes and operations that are added or redefined as the distinguishing features of the specialized classes.

Abstract classes. Inheritance is a relationship between classes, not between objects; it is a mechanism for organizing and simplifying the classes in the system and the relationships between them. Some classes are only used for purposes of classification and are never