shared superclass attributes have to be replicated in each of the subclass tables.

The third option when implementing an inheritance relationship is to create separate tables for the superclass and each of the subclasses. This has the advantage of being a straightforward approach, but will produce a very large number of tables in a complex system with many inheritance relationships, and may result in problems with navigability. There may also be problems with maintenance if the superclass is modified.

Although object-oriented databases do exist, it seems unlikely at present that they will replace relational databases for systems with reasonably straightforward data storage requirements. For the moment the most widely used approach to dealing with persistent data in an object-oriented system is to adapt the object-oriented models to fit a relational database, and to incorporate supplementary code, such as JDBC, to act as an interface between the database and the main program.

Technical points

Design patterns

Design patterns are tried and tested solutions to commonly occurring problems. For example, if you need to create a class that will only ever have one instance, you do not need to worry about how to do this – you can use the Singleton pattern. If you want to write code to access all the elements of a collection one by one without revealing the underlying structure of the collection, the Iterator pattern will provide the functionality to do this for you.

One of the aims of the object-oriented approach to developing systems is to promote genuine, effective reuse, and patterns are one of the main ways of achieving this. Using design patterns means that developers do not have to reinvent the wheel for each problem they encounter when they are designing and coding a system. Since patterns have been built up from the cumulative experience of many developers, they exhibit sound object-oriented principles, such as abstraction, cohesion and autonomy.[6]

A design pattern belongs in one of three categories, creational, structural or behavioural. Table 9.1 explains these categories and gives an example of each.

[6] *For a discussion of these, see Chapter 4.*