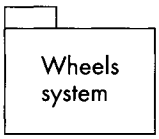

Figure 5.17 *A package representing the Wheels system*

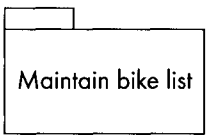

Figure 5.18 *Package of classes in the 'Maintain bike list' collaboration*

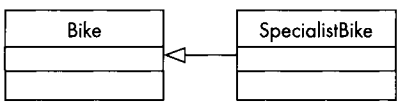

Figure 5.19 *Classes inside the 'Maintain bike list' package forming the 'Maintain bike list' collaboration*

top-level class diagram might consist only of packages. For a small system like Wheels, classes found at analysis can be grouped into one package, the system (see Figure 5.17).

Inside the system package would be all classes shown in the Wheels class diagram in Figure 5.15.

Classes should be grouped into packages on a logical basis. For example, classes are often grouped into collaborations: a collaboration consists of the classes used by a single use case (see Chapter 3). Figure 5.18 shows a package consisting of the classes in the 'Maintain bike list' collaboration.

Figure 5.19 shows the classes that are grouped inside the package.

Other logical groupings of classes are by subsystems, or by class type, e.g. interface classes, domain classes, control classes. These are discussed in Chapters 9 and 10.

Using the class diagram in system development

The class diagram is central to the development of an object-oriented system because it forms the basis for the structure of the software. Objects and classes are the basic building blocks of any object-oriented system. During analysis, the class diagram models