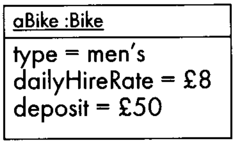

Figure 4.2 *A bike object*

attributes, the same behaviour and the same relationships. Classes are discussed in detail later in this chapter.

Object diagrams can be drawn with attributes and their values displayed as in Figure 4.2, or without attributes (as in Figure 4.6). In Figure 4.2 the attributes that are common to all Bike objects are type, dailyHireRate and deposit, and the values for this particular object are 'men's', '£8' and '£50'.

However, there is more to an object than its ability to record data. For the sake of completeness we are now going to give a more detailed definition of an object and discuss what is meant by the terms used in the definition. Some of this discussion will involve concepts that we have not yet met, so don't worry if you do not understand it all at the moment; you might find it useful to return to this section once you have read about classes.

An object is a concept, abstraction, or thing with clear boundaries and meaning in the current application area. We use objects both to model the real world characteristics of the application area and to provide us with a basis for the computer implementation.

An object can represent something concrete in the real world[1] such as this chair, that red bike, Ben's car or a concept such as a financial transaction (e.g. paying for a stamp), a customer order, the transaction of borrowing a book from a library etc. An object is always an abstraction (see Chapter 1) because although we want it to represent a real world thing, we are only interested in certain aspects of the real world thing. For example, a library member object will store information of interest in a library system: the member's name and address but not her passport number or her taste in shoes – we ignore currently irrelevant details. We are only including properties of the real world thing that have meaning for the problem at hand.

Every object in a system has three characteristics: behaviour, state and identity.

[1] *During design and implementation (see Chapters 9 and 10) we will meet objects that do not represent things in the real world, but artefacts we need for structural and implementation purposes.*