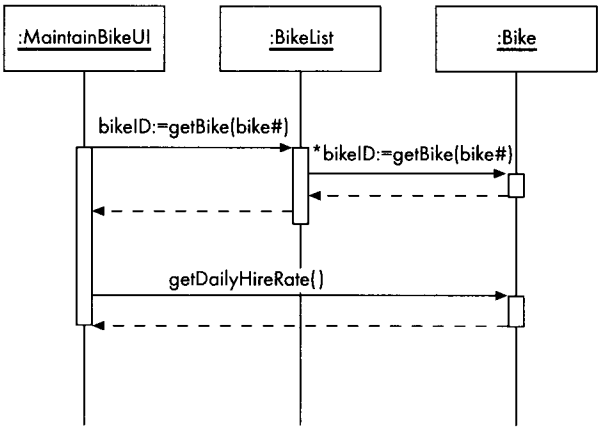

Figure 10.6 *Sequence diagram showing how the collection class works*

In other words, if an object needs to send a message, it must know how to get the message to the recipient. This means it must know the recipient object's identifier. It can do this is various ways.

- Via a direct link as in an association, e.g. in Figure 10.3 the School attribute, library, holds the object identifier for a SchoolLibrary object; this provides a direct link between a School object and a SchoolLibrary object.
- The calling object can be sent the target object's identifier by another object that has a link to the target object. In Figure 10.6 the attribute bikeID is set to the object identifier of the required bike, i.e. :BikeList finds the identifier of the :Bike and returns it to :MaintainBikeUI.
- An object identifier can be passed in as a parameter by a constructor to the object it creates. For example, in Figure 10.11 the interface object :IssueBikeUI creates a new Payment object and passes it the identifier of the relevant Customer object as a parameter. :IssueBikeUI also creates a new Hire object and passes it Bike and Customer object references.

For other ways of creating navigable paths between objects see Deitel and Deitel (2003).

Designing attributes and operations

Attribute signatures. During design we need to specify the details of each attribute's signature. The UML format of an attribute signature is:

visibility name : type-expression = initial-value