matching bike#. :IssueBikeUI then sends a message to that :Bike which displays its details (to check that the correct bike has been found).

3 :IssueBikeUI asks the actor how long the hire is for, collects the actor response and sends a message to the :Bike asking it for the costs, the :Bike displays the deposit, daily hire rate and cost of the hire for the specified number of days.

4 :IssueBikeUI waits for a response from the actor to know whether or not to proceed with the hire. The actor responds with a request to create a new :Customer. :IssueBikeUI creates a new Customer object and populates its attributes with the values passed in by the actor.

5 :IssueBikeUI automatically creates a new Payment object when it creates the Customer object.

6 :IssueBikeUI automatically creates a new Hire object and sets the start date to today's date and the number of days to the number specified in step 3.

7 :IssueBikeUI then asks :Payment to calculate the total cost, including the deposit.

8 :Payment issues a receipt. To do this it asks its :Customer for the name and postcode so that this can be printed on the receipt. :Payment then asks the :Bike for the total cost so that this too can be printed on the receipt.

Notice that the sequence diagram in Figure 10.11 has two Bike object icons. On a sequence diagram it is permissible to represent more than one object of the same class. In this case we use the first Bike icon to represent Bike in its capacity as a collection class and the second to represent the individual Bike object with the matching bike#.

Chapter summary

Analysis is concerned with specifying what a system has to do; design is concerned with specifying how to deliver that functionality. Design activities concerning the overall system design specify the overall system architecture logically, using a layered model, and physically using component and deployment diagrams. This is discussed in Chapter 9. Detailed design activities in an object-oriented system require us to specify the class diagram in more detail. Boundary, control and collection classes must be added. Attribute and operation signatures must be completely specified. Decisions must also be made about how relationships