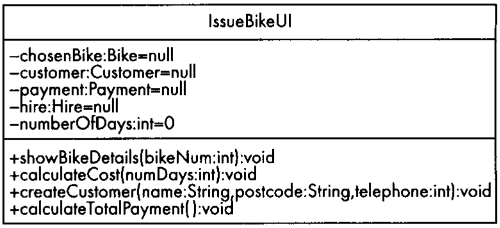
IssueBikeUI
–chosenBike:Bike=null
–customer:Customer=null
–payment:Payment=null
–hire:Hire=null
–numberOfDays:int=0
+showBikeDetails(bikeNum:int):void
+calculateCost(numDays:int):void
+createCustomer(name:String,postcode:String,telephone:int):void
+calculateTotalPayment( ):void

Figure 11.5 *Class diagram for the IssueBikeUI class*

- findBikeByNumber(bikeNum) iterates through its array of bike objects until it finds one with a matching bike number (one that matches the value of the parameter bikeNum). The reference (object identifier) of the matching bike is returned and assigned to the attribute chosenBike. This reference is then used to send the message, showDetails(), to the matching bike.

The remaining method declarations are:

- +calculateCost(numDays:int):void on line 50
- +createCustomer(name:String,postcode:String,tel:int):void on line 57
- +calculateTotalPayment():void on line 64.

Notice that all of these methods are public, i.e. they can be called by any object. The method, createCustomer(), actually creates a Customer object, a Payment object and a Hire object.

Bike class

The class diagram for the Bike class is shown in Figure 11.7, the code for the Bike class is shown in Figure 11.8.

- The Bike class is declared in line 71.
- An array of 5 Bike objects, bikeList, is declared in line 74.
- The attributes (also known as member variables) are declared in lines 76–78.
- The Bike constructor is declared on line 91; in lines 93-95 the member variables are set to the values passed in as parameters to the constructor.

There are three get methods:

- getDeposit() declared on line 98
- getRate() declared on line 102
- getBikeNumber() declared on line 106.

The method findBikeByNumber() is declared on line 110. This method, as we mentioned above, iterates over the array of