Chapter summary

This chapter demonstrates that there is a close relationship between the class diagram and the code. The classes, attributes and operations of the class diagram form the framework of the code. The relationship is so close that in a CASE tool like Together the skeleton of the code is automatically generated from the class diagram. All the programmer has to do is flesh out the methods. Conversely, a CASE tool can automatically reverse engineer the class diagram from the code. This chapter also demonstrates the usefulness of the sequence diagram. The sequence of execution in an object-oriented program is complicated; it is hard to follow the flow of control as it is passed backwards and forwards between objects. Sequence diagrams provide an overview of the inter-object messaging sequence; this is useful for software designers and for programmers, both when they are writing the code and when they are maintaining it.

It is interesting, looking back at the process of developing the software for the Wheels system, to see that the entities Bike, Customer, Hire and Payment existed initially in the real world of the Wheels bike shop, were then classes in the analysis and design models, and are still present in the code. Towards implementation, we picked up a few extra classes and the original classes picked up a few extra features, but basically there was a seamless transition from requirements elicitation through to code.

Bibliography

Britton, C. and Doake, J. (2000) *Object-Oriented Systems Development: A Gentle Introduction*, McGraw-Hill, London.

Charatan, Q. and Kans, A. (2002) *Java in Two Semesters*, McGraw-Hill, London.

Deitel, H.M. and Deitel, P.J. (2003) *Java: How to Program* (5th edition), Prentice Hall, Upper Saddle River, NJ.

Quick check questions

You can find the answers to these in the chapter.

a Which method must be present in all Java applications?

b Which two types of method are usually omitted on a class diagram?

c What is a constructor? How can you identify it?

d How does the Java compiler know where a class starts and ends?