

## **Part F:**

---

# **Applying the method in various languages and environments**

Previous chapters have developed hand in hand the object-oriented method and the supporting notation. Part *F* will study how to realize the ideas, or emulate them, in some of the most popular languages and environments. There are three broad categories. **Object-oriented languages** such as Smalltalk support many of the fundamental concepts. **Classical languages** such as Fortran are not O-O at all, but people who must still use them (for various reasons, often not technical) may want to apply as many O-O ideas as feasible within the limitations of these older approaches. Somewhere in-between, **encapsulation languages** such as Ada provide a first step to object orientation, essentially limited to modules based on abstract data types, but without classes or inheritance.

Although the logical progression is from Classical to Encapsulation to O-O, it will be convenient to start with Encapsulation languages, focusing on Ada 83, as it provides a good reference to assess techniques applicable in classical languages, the category that will follow; we will end with a review of some of the principal O-O languages other than the notation of this book.