Random Fact 15.1

Standardization

You encounter the benefits of standardization every day. When you buy a light bulb, you can be assured that it fits the socket without having to measure the socket at home and the light bulb in the store. In fact, you may have experienced how painful the lack of standards can be if you have ever purchased a flashlight with nonstandard bulbs. Replacement bulbs for such a flashlight can be difficult and expensive to obtain.

Programmers have a similar desire for standardization. Consider the important goal of platform independence for Java programs. After you compile a Java program into class files, you can execute the class files on any computer that has a Java virtual machine. For this to work, the behavior of the virtual machine has to be strictly defined. If virtual machines don't all behave exactly the same way, then the slogan of "write once, run anywhere" turns into "write once, debug everywhere". In order for multiple implementors to create compatible virtual machines, the virtual machine needed to be *standardized*. That is, someone needed to create a definition of the virtual machine and its expected behavior.

Who creates standards? Some of the most successful standards have been created by volunteer groups such as the Internet Engineering Task Force (IETF) and the World Wide Web Consortium (W3C). You can find the Requests for Comment (RFC) that standardize many of the Internet protocols at the IETF site, `http://www.ietf.org/rfc.html`. For example, RFC 822 standardizes the format of e-mail, and RFC 2616 defines the Hypertext Transmission Protocol (HTTP) that is used to serve web pages to browsers. The W3C standardizes the Hypertext Markup Language (HTML), the format for web pages—see `http://www.w3c.org`. These standards have been instrumental in the creation of the World Wide Web as an open platform that is not controlled by any one company.

Many programming languages, such as C++ and Scheme, have been standardized by independent standards organizations, such as the American National Standards Institute (ANSI) and the International Organization for Standardization—called ISO for short (not an acronym; see `http://www.iso.ch/iso/en/aboutiso/introduction/whatisISO.html`). ANSI and ISO are associations of industry professionals who develop standards for everything from car tires and credit card shapes to programming languages.

When a company invents a new technology, it has an interest in its invention becoming a standard, so that other vendors produce tools that work with the invention and thus increase its likelihood of success. On the other hand, by handing over the invention to a standards committee, the company may lose control over the standard. For that reason, the Java standard is developed by the "Java Community Process", an industry consortium controlled by Sun Microsystems, the inventor of Java.

Of course, many important pieces of technology aren't standardized at all. Consider the Windows operating system. Although Windows is often called a de-facto standard, it really is no standard at all. Nobody has ever attempted to define formally what the Windows operating system should do. The behavior changes at the whim of its vendor. That suits Microsoft just fine, because it makes it impossible for a third party to create its own version of Windows.

As a computer professional, there will be many times in your career when you need to make a decision whether to support a particular standard. Consider a simple example. In this chapter, we use the `LinkedList` class from the standard Java library. However, many computer scientists dislike this class because the interface muddies the distinction between abstract lists and arrays, and the iterators are clumsy to use. Should you use the `LinkedList` class in your own code, or should you implement a better list? If you do the former, you have to deal with a design that is less than optimal. If you do the latter, other programmers may have a harder time understanding your code because they aren't familiar with your list class.