call that method. This is the approach used in the Java serial port package, javax.comm, discussed next.

The javax.comm Package as a Singleton

The javax.comm package is provided separately from the Java Software Development Kit (SDK) and is downloadable from the Sun Web site. This package is designed to provide control over the serial and parallel ports of a computer. Each supported platform has a different implementation of this package, since it must include native code to handle the actual system hardware.

Serial ports are a good example of a resource that should be represented by a Singleton, since only one program at a time can access a serial port, and even within a program only one module or group of modules should be communicating over the port.

Actually, two Singletons are possible here: one that manages the collection of ports and lets out only one instance of each port and one that manages the port objects themselves, which must each refer to a single port.

In the javax.comm package, the CommPortIdentifier class has a static method for returning an enumeration of all ports in the system:

```
public static Enumeration getPortIdentifiers()
```

This enumeration contains a CommPortIdentifier object for each element in the system. This package handles both parallel and serial ports, so you also need to filter the returned objects to allow only the serial ports.

```
//get enumeration of ports--static method
portEnum = CommPortIdentifier.getPortIdentifiers();

while (portEnum.hasMoreElements()) {
   portId = (CommPortIdentifier) portEnum.nextElement();
   if (portId.getPortType() == CommPortIdentifier.PORT_SERIAL)
       ports.addElement (portId.getName()); //add to list
}
```

The *getPortIdentifiers* method is static, so it provides the single global point of access required by the Singleton definition. Further, since it is a static method, multiple instances are not allowed. You can call this method only from the class and not from its instances. Each CommPortIdentifier instance returned from this enumeration has a distinct port name, which you can access using the *getName* method.

This enumeration consists of all possible ports for the system, whether opened or not, implemented or not. For example, on a PC the enumeration