Figure 10.1 Processing a *Vector*.

First, let's create a Vector containing some different objects
(recall this is another useful feature of a Vector; namely that it
can store *different* objects).

```
Vector list = new Vector();     // list is created with no elements
Student stu1 = new Student("Richard", "890792492");
list.addElement(new Integer(7));     // now list has one element
list.addElement("hello");     // now two, an Integer and a String
list.addElement(stu1);     // an Integer, a String, and a Student
```

Since a Vector is indexed, it's natural to use a loop to process it:

```
for (int i = 0; i < 3; i++)
    System.out.println(list.elementAt(i));
```

This prints out

```
7
hello
```

Note that you did not need to know the *type* of each element
when in the body of the **for** loop above. Why?

Using loops is straightforward, but you do need to be careful about
(1) going far enough, and (2) not going too far. If you don't want to
do the counting yourself, you can use an enumerator:

```
Enumeration enum = list.elements();
while (enum.hasMoreElements())
    System.out.println(enum.nextElement());
```

The object **enum** gives you access to the Vector elements one at time
without your having to manage the counting yourself.

Enumeration is an example of an *interface*; you'll learn about those later.