

## PRAXIS 4

## GENERAL TECHNIQUES

modate. Moreover, when elements are removed from a `Vector`, each element with an index greater than the index being removed is shifted downward to have an index one smaller than the value it had previously.

Unlike arrays, you call a method on a `Vector` to determine its size. The `Vector` class implements a `size` method, which returns the number of elements contained in the `Vector`. The `size` method might not return the result you expect, however. Because the `size` method returns the number of elements contained in a `Vector`, removing an element changes its size. On the other hand, an array is fixed in size. Its length is the same regardless of how many array indexes have been assigned values.

A `Vector` is implemented in terms of an array. That is, when you create a `Vector` the class creates an array of elements of type `java.lang.Object` to manage the items you store in the `Vector`. When a `Vector` grows, its entire array must be reallocated and copied. In addition, when an item is removed from a `Vector`, its underlying array is compacted. These attributes of the `Vector` class, coupled with its array implementation, can create performance problems if `Vectors` are not used properly. See PRAXIS 41 for a detailed analysis of the performance characteristics of arrays and `Vectors`.

Finally, a `Vector` may contain only object references and not primitive types. Arrays, by contrast, may contain either object references or primitive types. This restriction is because the `Vector` class uses an array of type `java.lang.Object` as its supporting data structure. For example, compiling the following code:

```
import java.util.Vector;
import java.awt.Button;
class VecArray
{
    public static void main(String args[])
    {
        int i = 1;
        int[] ia = new int[10];
        ia[0] = i;                                //OK

        Button[] ba = new Button[10];
        ba[0] = new Button("");                    //OK
        Vector v = new Vector(10);
        v.add(new Button(""));                     //OK
        v.add(i);                                  //ERROR
        v.add(new Integer(i));                     //OK
    }
}
```