Figure 9.1 A first look at *Vectors*.

Recall these parts of our **WashingMachine** class:

```
public class WashingMachine {
    private String contents;    // what's in the machine

    public void add (String item) {
        contents = item;
    }
    public String getContents() {
        return contents;
    }
    public void printContents () {
        System.out.println("Inside you will find " + contents);
    }
}
```

When we ran this …

```
public class TestWash {
    public static void main (String[] args) {
        WashingMachine mach = new WashingMachine ();
        mach.add("sock");
        mach.printContents();
        mach.add("hat");
        mach.printContents();
    }
}
```

… we got the output

```
Inside you will find sock.
Inside you will find hat.
```

Our **add()** method didn't "add" at all—it "replaced"!

The problem is that **contents** can hold only a single string. One way around this is to concatenate to **contents** each time by replacing

contents = item in **add()** with $\boxed{}$.

But then it would be cumbersome to retrieve a given item afterwards. Instead, it would be much better if we had an object that could contain many other objects, each of which could be accessed *independently* and *conveniently*.

One such object is a **Vector**. Objects of the **Vector** class can hold objects of any class (including **Vectors** themselves).