Figure 11.0 A first look at *Arrays*.

Vectors are useful, but they suffer from a couple of drawbacks:

- they're a nuisance when you want to deal with primitive values, because you need to use [ ] to store them and [ ] to get them back, and
- they're not efficient to process, especially when they get long.

In Java, *arrays* provide another way of handling lists of items. They are more efficient to process than Vectors, but they achieve this efficiency by giving up some of the freedom of Vectors.

We'll introduce arrays by developing a program to find the *median* of a sequence of integers. Suppose we have the following sequence:

```
2  11  -3  428  97  -12  -96  13  ...
```

Let's say there are N of them. Finding the *average* is easy:

- read the numbers one at a time
  – add each to a running total
  – count them (that's how we find out there's N of them)
- divide the total by N

But what about the median? This is defined as the member of the list that has the "middle" value; i.e., the member that is bigger than exactly $N/2$ of the others.

Here's one way we could find the median: