Figure 9.4 Some rules about *Vectors*.

Here's what you need to remember about Vectors:

0. The things stored in a Vector are called *elements*. The sequence number of an element in the Vector is called its *index*.
1. The elements are numbered from 0 upward.
2. There are no gaps in the numbering. When you remove an element, the later ones are (automatically) renumbered.
3. The Vector (automatically) expands as needed to accommodate newly-added elements.
4. All elements must be *objects*. Values of primitive type are *not allowed.*
5. Each element retrieved from a Vector must be cast to the right type.

So how can a single Vector contain *different* kinds of objects? Actually, it doesn't. It contains only one kind—**Object** objects! (Or, more precisely, objects of the class **Object**.)

For example, the signature of the **addElement()** method is:

Fortunately, every object *does* belong to the class **Object**, and we'll see how when we talk about "inheritance" later in the course. This gives us the freedom to store *any* object we like in a Vector!

The catch is when we extract an element from a Vector we must reassure the compiler we have the right type, and that's why we need to use a cast.

For example, if the 5-*th* element stored in our Vector is an object of class **Student**, here's how we extract it: