

Programming Assignment 3-3

Let's face it: Sometimes, arrays are not what you need when you are trying to store a sequence of objects – particularly when you don't know in advance how many objects you will need to store.

In this exercise, you will create a class called `MyStringList` that will be a "growable" array for `Strings`. In fact, you will implement this class by having a private array **`strArray`** of `Strings` in the background, but a user will not know about your implementation. A user will know only that you will provide the ability for the user to add, remove, and search for `Strings`.

Here is the specification in detail:

Your class should implement the following public methods:

```
public void add(String s) (
    - adds s to the end of the underlying array
public String get(int i)
    - retrieves the String at the ith position in the underlying array
public boolean find(String s)
    - returns true if String s is found in the array, false otherwise
public boolean remove(String s)
    - removes first occurrence of String s if it is found in the underlying array
    if found, returns true; if not found returns false
public String toString()
    - returns a String representation of the underlying array
    here is a typical output:
    [Bob, Steve, Susan, Mark, Dave]
public int size()
    - returns the next open position in the underlying array – this is precisely the
    number of Strings that have been added minus the number of String that have
    been removed.
```

You should also implement this private method

```
private void resize()
```

In your class, the `resize()` method must be called whenever addition of another `String` to the underlying array goes beyond the current length of the array. The `resize` method should use `System.arraycopy` to create a new `String` array having **twice the size** of the current `String` array, and should copy the old array into the first part of the new array.

You should have an instance variable **`size`** in your class that is incremented or decremented as `Strings` are added or removed.

Your constructor for `MyStringList` should set the value of `size` to 0, but should initialize `strArray` to a much larger value, like 50. This will accommodate modest uses of your list without requiring the underlying array to be resized at all.

For *testing* purposes, however, I want you to do the following (and submit code that includes this additional test code):

- Initialize `strArray` so that it has size 2
- Add the following statement to your `resize()` method:
`System.out.println("Resizing...");`
- Create a main method in your class that executes the following test code:

```
MyStringList l = new MyStringList();
l.add("Bob");
System.out.println("The list of size " + l.size() + " is " + l);
l.add("Steve");
System.out.println("The list of size " + l.size() + " is " + l);
l.add("Susan");
System.out.println("The list of size " + l.size() + " is " + l);
l.add("Mark");
System.out.println("The list of size " + l.size() + " is " + l);
l.add("Dave");
System.out.println("The list of size " + l.size() + " is " + l);
l.remove("Mark");
System.out.println("The list of size " + l.size() + " is " + l);
l.remove("Bob");
System.out.println("The list of size " + l.size() + " is " + l);
```

Here is the expected output:

```
The list of size 1 is [Bob]
The list of size 2 is [Bob, Steve]
Resizing...
The list of size 3 is [Bob, Steve, Susan]
The list of size 4 is [Bob, Steve, Susan, Mark]
Resizing...
The list of size 5 is [Bob, Steve, Susan, Mark, Dave]
The list of size 4 is [Bob, Steve, Susan, Dave]
The list of size 3 is [Steve, Susan, Dave]
```