CS151 - Prelab №2

James Capuder, Oberlin College February 16, 2015

Part 1: MyArrayList

1. You'll be using generics for your implementation, and taking advantage of the fact that List and AbstractList are similarly parameterized. Given the information above, give the declaration (i.e. "public class...") for MyArrayList. Remember to properly indicate the parent and that we'll be using Generics to handle the storage type. (Refer to Weiss 4.7 for information on generics.)

```
public class myArrayListM<AnyType> extends AbstractList
```

2. Skimming through the documentation for AbstractList (specifically the top text description), what methods must you implement to create a concrete child class? What methods do you need to override to be able to do add/removes successfully?

To have a successful child class, two methods must be implemented, and to add or remove values, an additional 3 must be overridden. The two methods neccessary for a basic implementation are the get(int index) and size() methods. For a more robust implementation, set(int, E), add(int, E), and remove(int) must all be overridden.

3. Which of the methods that you identified in question 2 might require that you resize the array in order to be sure all the data can fit?

The add method is the only method mentioned in question 2 that could require the array to be resized.

4. Write a private resize method that increases the length of the data array by one. You will need to make a new array of larger size, copy the data from the old array into the new array, and change the reference to point to your new array. Let's assume that our storage array is declared AnyType data[].

```
private myArrayList resize(dataArray){
    int newLength = dataArray.size();
    data = new AnyType[newLength+1];
    for (int i; i<newLength; i++){
        data.set(i, dataArray[i]);
    }
    return data;
}
```

5. Suppose that data.length is currently s. How many assignment statements does your resize method need to perform, in terms of s? (You may ignore loop counter assignment statements, such as i++.) If you're having a hard time coming up with the formula, try counting the number of assignments when s=2, s=3, etc., and then try to find a pattern.

My resize method performs s assignment statements.

6. Write the implementation for the set(int index, AnyType element) method. Be sure to throw the specified exception when the index value is outside of the valid range (see ArrayList's set method for details.)

```
public AnyType set(int index, AnyType element){
    try{
        AnyType temp = MyArrayList[index];
        MyArrayList[index] = element;
        return temp;
    } catch (IndexOutOfBoundsException e){
        System.out.println("Index out of range"+e.getMessage());
    }
}
```

7. Now write the implementation for the add(int index, AnyType element) method. Again, don't forget to throw the specified exception and be sure to call your resize method from above if necessary. (See ArrayList's add method for details.)

```
public void add(int index, AnyType element){
    try{
        resize(MyArrayList);
        int position = index;
        for (int i = position-1; i >= 0; i--) {
            MyArrayList[i+1] = MyArrayList[i];
        }
    } catch (IndexOutOfBoundsException e){
        System.out.println("Index out of range"+e.getMessage());
    }
}
```

8. Suppose you start with an ArrayList of size 0 and capacity 1, (i.e., data.length==1), and you add n elements one-by-one to the ArrayList. How many calls to resize will you make, in terms of n? What is data.length for these successive calls? Therefore, in total, how many assignment statements are performed for all these resizes? I will accept any answer in an acceptable range, but be as accurate as you can. Completing the following chart may help organize your thoughts (I've filled in a few boxes for you). Your final answer would be the sum of the entries in the bottom row. It may be useful to remember that 1+2+...+n = n(n+1)/2

$$\frac{n(n+1)}{2} - 1$$

9. Now suppose you change your resize method to double the length of the data array, instead of only incrementing by one. Answer the previous question again and comment on the result. If it helps, you may assume that n is a power of 2. It may be useful to remember that 1+2+...+2p= 2p+1-1 = 2*2p-1. Again, I'm most interested in the total number of assignments and the chart is to help you determine this value.

$$2^{p+2} - 1$$

10. Come up with three short tests that will test an aspect of one of the methods you are planning on implementing. Describe (in pseudocode) what actions you will do in the test and what the expected outcome should be. Put some thought into where your programming mistakes are likely to be, and write tests that will suss those out. Frequent sources of errors come from adding/removing things, repeated operations, and dealing with edge cases (is that supposed to be less-than or less-than-or-equal?)

Add method	Print each element in an array, perform my add function, re-print the array for comparison.
Resize method	Check length, sizes, and content of an array before and after resizing it.
Set method	perform method, check return value and elements in the array.