Students:

This content is controlled by your instructor, and is not zyBooks content. Direct questions or concerns about this content to your instructor. If you have any technical issues with the zyLab submission system, use the **Trouble with lab** button at the bottom of the lab.

20.22 Ritchie: MyVector class

Program 5

Due: Monday 11/25/19 , 11:59PM

Grade: 90% Zybooks unit tests, 10% I will hand grade your assignments When I grade your assignments I will be looking for:

- Author documentation
- Good variable names
- Comments: Write appropriate comments and / or java documentation (for an example of this go to blackboard -> Programming -> Programming Guidelines -> Example program)
- Correct implementation of the classes (IE no hard coding the answers).

Objectives

- Design a data structure that behaves like a dynamically allocated vector
- Implement a list interface
- Extend an abstract class
- Convert a generic Object type class to a parameterized data type

Instructions for MyVector

The class header will be:

```
public class MyVector<E> extends MyAbstractList<E>
```

Take care when creating new arrays. We cannot create an array this way:

```
E[] mylist = new E[10]; // cannot do this
```

Instead, we must create an array of Object references and then cast the array to E:

```
E[] mylist = (E[]) new Object[10]; // can do this
```

But now the compiler warns us that this conversion may be problematic. To turn off the warning, we add a compiler directive. Like this example of a no-arg constructor:

```
    @SuppressWarnings("unchecked")
    public MyVector() {
        array = (E[]) new Object[10];
        size = 0;
    }
```

In addition to implementing MyList interface, provide three constructors:

- one with no parameters that initializes the vector to a capacity of 10 elements,
- one getting the initial capacity of the vector from the parameter, and
- another that take the initial capacity of the vector and the capacity increment.

Last, add three more methods:

- *getCapacity()* which returns the capacity (not the size) of the internal array structure
- *getIncrement()* which returns the capacity increment of the vector.

Remember Capacity is different from Size. Capacity refers to the number of items a list or array might hold. Size is actual number of items being held in a list or array.

For example:

```
    array = (E[]) new Object[10];
```

Has a capacity of 10, but a size of 0. The default value for capacityIncrement should be 5.

Also a reminder to grow your array when the size equals the capacity. How much should the array grow by? That is the purpose of the capacity increment variable. For example, if your capacity is 10, and your capacityIncrement is 5, then when your array holds 10 elements you should resize your array to have a new capacity of: old capacity plus capacityIncrement, in the case of this example that would be 15.

Turning in your program

Upload to ZyBooks:
(1) MyVector.java (2) MyList.java (3) MyAbstractList.java

- Your program will be graded automatically against the requirements.
- You may submit as many times as necessary.
- The automatic grading program is very specific. If you feel you have the correct solution but are not receiving full credit, please
    - Carefully review the output -- you might need to scroll all the way to the right to find what is wrong with a particular output.
    - Verify you have the correct names for the program itself and all methods.

- Check your calculations by hand: was there a logic error?
- Review the requirements: did you miss a step? misinterpret a requirement?
- If all these check out, contact the T.A. for assistance.

LAB ACTIVITY | 20.22.1: Ritchie: MyVector class 56 / 56 ✓

Submission Instructions

Downloadable files

`MyList.java` and `MyAbstractList.java` **Download**

Compile command

```
javac MyVector.java MyList.java
MyAbstractList.java -Xlint:all -
encoding utf-8
```

We will use this command to compile your code

Upload your files below by dragging and dropping into the area or choosing a file on your hard drive.

MyVector.java Drag file here or **Choose on hard drive.**

MyList.java Drag file here or **Choose on hard drive.**

MyAbstr...t.java Drag file here or **Choose on hard drive.**

Submit for grading

Latest submission - 5:16 PM on 11/25/19

Submission passed all tests ✓

Total score: 56 / 56

□ Only show failing tests

Download this submission

1: Unit test ^ 4 / 4

Test that add(E data) and get(E data) work correctly.

2: Unit test ^ 4 / 4

Test that the isEmpty method works correctly

3: Unit test 4 / 4

Make sure that your vector grows dynamically.

4: Unit test 4 / 4

Test that add(E data) and remove(int index) work correctly.

5: Unit test 4 / 4

Test that add(int index, E data) and indexOf(E data) work correctly.

6: Unit test 4 / 4

Tests the indexOf(E data) method more extensively.

7: Unit test 4 / 4

Test the contains(E data) method works properly.

8: Unit test 4 / 4

Test the toString() method works properly.

9: Unit test 4 / 4

Test MyVector<>(capacity, increment) sets initial values

10: Unit test 4 / 4

Tests trimToSize() after adding 5 elements: should reduce vector to 5 elements

11: Unit test 4 / 4

Tests getCapacity() returns same as constructor argument passed in and INITIAL_CAPACITY_INCREMENT is the default value.

12: Unit test ^ 4 / 4

Tests getIncrement() returns 20 given constructor call MyVector<>(2,20)

13: Unit test ^ 4 / 4

Test lastIndexOf(3) after adding, in order: 3,1,3,1,3,1,3 returns 6 and Test lastIndexOf(6) returns -1

14: Unit test ^ 4 / 4

Tests clear() successfully emptied the list of its elements

Test feedback

```
After clear size is 0 and capacity is 7
```

5 previous submissions

5:03 PM on 11/25/19	56 / 56	View ⌄
4:55 PM on 11/25/19	44 / 56	View ⌄
4:29 PM on 11/25/19	44 / 56	View ⌄
4:12 PM on 11/25/19	44 / 56	View ⌄
1:51 PM on 11/25/19	44 / 56	View ⌄

Trouble with lab?