Programming Assignment 2

Due 14 SEP @ 11:59pm

Write a program that implements the List API ("List.java", provided with documentation) as a dynamic array as covered in lecture. A code template DynamicArray.java has been provided for you - place your code in the appropriate methods. If a method indicates an exception for some condition, be sure to check and throw this when the condition occurs.

The methods defined in List.java already have javadoc – use them (copy/paste is fine) in your own implementation for the method documentation. However, documentation rules still apply for methods you add and the code in the methods that you add.

The constructor should initialize the internal array length to the constant MIN_CAPACITY. Your dynamic array must grow as needed (as items are added/inserted) and it must also shrink as needed (items are removed). The grow and shrink must use the approach shown in lecture. For shrink, you must ensure that the capacity never goes below the constant MIN_CAPACITY.

The toString and main method have already been implemented for you, do not change. The main method takes in the file "operations.txt" for unit testing your code. Additionally, a "Stdio.java" file is provided and used in the main, do not modify. You can only add methods – you cannot change the signature of the class, constructor, the constant, or existing methods. Below are notes concerning each method (typically the parameters passed by the caller are checked for "preconditions" that must be satisfied when calling the method).

You only need to work on DynamicArray.java. If you program is working correctly, you will see the same output in the last page of this assignment when you run your code.

public DynamicArray() // constructor

- allocate initial memory using constant

public Type get(int i);

- check i for range

public void set(int i, Type item);

- check i for range, make sure item is not null

public void add(Type item);

- make sure item is not null
- may have to grow to increase capacity

public Type remove(int i);

- check i for range
- may have to shrink to reduce capacity

public void insert(int i, Type item);

- check i for range, make sure item is not null, make sure capacity, shift right

public int size();

- keep a variable and update in other methods, do not try to derive

public int capacity();

- note: not part of the List interface (because not all Lists use arrays)
- return the length of the internal array

Grading Notes

You must:

- Use the template provided for you
- Use the generic type in method signatures and your code
- Have a style (indentation, good variable names, etc.)
- Comment your code well (no need to over do it, just do it well)

You may:

- Add additional private utility methods you need (not necessary but makes easier).

You may not:

- Make your program part of a package.
- Use *code* from anywhere except your own brain.
- Use *code* from the Java Collections library, including ArrayList.
- Cast items to avoid using the generic type (exception: array creation in the constructor)

Submission Instructions:

- Name a folder with your gmu username
- Put your java files in the folder (but not your .class)
- Zip the folder (not just the files) and name the zip “username-pa2.zip”
- Submit to blackboard

Grading Rubric

No Credit:

- Non-submitted assignments
- Late assignments
- Non-compiling assignments
- Non-independent work

1pt	Submission Format
1pt	Style and Comments
2pts	add method with grow
2pts	remove method with shrink
2pts	insert method
2pts	set, get, and size methods

Example Run

```
> java DynamicArray operations.txt
adding student1
adding student2
remove student1
adding student3
adding student4
adding student5
adding student6
capacity=8
adding student7
remove student2
adding student8
adding student9
adding student10
adding student11
capacity=16
remove student10
remove student9
remove student8
remove student7
remove student6
capacity=8
remove student5
size=3
capacity=8
remove student3
capacity=4
remove student4
remove student11
size=0
capacity=4
adding student12
adding student13
adding student14
set 1 to graduate13
size=3

Final list=[student12, graduate13, student14]
>
```