PROJECT #2: 80 points, CSE 274 – Fall 2019
Set, Resizable Array-based Implementation

Outcomes:

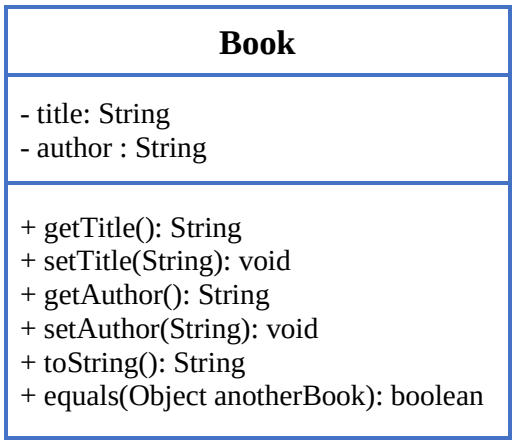
- Implement an ADT (specifically a resizable array-based implementation of the Set ADT)
- Test a class using the JUnit library

Naming requirements (not following any of these may result in a score of 0):

- The Eclipse project name must be **Project2**.
- You will have write exactly two source code files: **ResizableArraySet.java** and **SetTester.java**
- You will also need to include the Set interface as specified in **SetInterface.java**. Do not modify this file. Your resizable array set must implement this interface.
- You will use the **default package** (this means there should be no package statements in any of your files).

Your assignment is to:

1. In a class named **ResizableArraySet**, implement the abstract data type Set by implementing all the methods in the interface found in **SetInterface.java**.
2. It will be a set of **Book** objects (write the **Book** class using the following UML class diagram).



The toString method in Book class should return a formatted string with title and author of the book. For example, if it is a book titled "The Book Of Why" by Judea Pearl then toString should return "The Book Of Why by Judea Pearl". The equals method compares two Book objects and returns a boolean. If two books have the same title and author then it should return true and false, otherwise. A book can have at most 1 author. You should write a constructor to initialize the instance variables.

3. Your implementation will use a resizable array of Book objects (use the **Book** class designed above) that doubles in size any time an item is added when the array is already full.
   Also, when you remove an item check the number of items and size of array too. If the number of items in the array is less than or equal to one-third of size (capacity) of the array, then shrink the array to half of the current capacity. Do not use Java's ArrayList. You should simply work with an array, and handle the array resizing on your own.

PROJECT #2: 80 points, CSE 274 – Fall 2019
Set, Resizable Array-based Implementation

4. The only instance variables should be the array of Book objects and a counter for the number of entries in the set.
5. There should be two constructors:
    a. A constructor with no parameters. By default, this should use an array instance variable with a starting size of 10.
    b. A constructor with an int parameter specifying the starting size of the array instance variable.

6. SetTester should thoroughly test the methods and constructors in the ResizableArraySet class. It should utilize the JUnit library to test the ResizableArraySet class. You should comment your testing code on what is being tested, what results are expected, and the actual results. **This should not involve any interaction from the user.** Just run test cases. Comments in your test methods should look something like this:

    // **Creating an empty set and adding two books** ("Eyewitness", "Carolann Camillo") ("Great Lakes", "Andrea Boeshaar")
    // **Expecting to see** ("Eyewitness", "Carolann Camillo") ("Great Lakes", "Andrea Boeshaar")
    // **Returns** ("Eyewitness", "Carolann Camillo") ("Great Lakes", "Andrea Boeshaar")

    // **Removing** ("Great Lakes", "Andrea Boeshaar") **and expecting to see** ("Eyewitness", "Carolann Camillo")
    // **Returns** ("Eyewitness", "Carolann Camillo")
    Your test code should not require me to look at your source code. I should know, by running your tester, what is being tested, what results are expected, and what the actual results are. I will be looking for:
    - Thoroughness (test all constructors and methods)
    - Organization (keep related tests together)
    - Readability (use single blank lines in appropriate places to break your code into "chunks" so it's easy to know when one part of your testing is done, and the next part begins). The Arrays.toString() is a useful way to display the contents of an array. Feel free to use it to output your results.

Grading Rubric:

Outcome	**Max score**
Correct Book Class	7
Constructors implemented correctly	2
Array grows and shrinks as specified	10
Intersection, union, difference work as specified	21

PROJECT #2: 80 points, CSE 274 – Fall 2019
Set, Resizable Array-based Implementation

Remaining methods work as specified. Add and remove methods use the same algorithms we learned for Bag	16
Tester is thorough, organized, and readable (2 points for each test methods, there are 11 methods) Must use JUnit library, otherwise 0	24
Code formatted according to generally accepted standards	0 (deductions only)
Code follows approaches taught in CSE 174 and 271	0 (deductions only)

Submission:

You will submit Java source code files: **Book.java, ResizableArraySet.java** and **SetTester.java** on Canvas.