Assignment 7
Problem #3 of 3
Lakes
Lists And Arrays
4/19/19

Getting Started: If you haven't already unzip “Assignment 7 Files.zip” into your working directory. You will see a directory named “Lakes”. That is the directory we'll use for this problem.

See “General Notes” and “General Hints” in Problem #1 document.

All **classes** should be designated as **public** unless otherwise noted.
All **methods** should be designated as **public** unless otherwise noted.
All **instance variables** should be designated as **private** unless otherwise noted

Problem 1 -- Lake

Design and code up a Lake class.

Class Name: Lake

Instance Variables:

Instance Variables (all private):

Name	Type	Description
name	String	Name of lake
nearbyTown	String	Name of town nearest lake
maxDepth	int	Max depth of lake (in feet)
surfaceArea	int	Surface area of lake (in acres)

Constructors:

Name	Parameters	Notes
Lake	Four parameters that match the four instance variables (in same order)	

Instance Methods

Name	Parameters	Return Type
getName	None	String – Return the corresponding ivar
getNearbyTown	None	String – Return the corresponding ivar
getMaxDepth	None	int – Return the corresponding ivar
getSurfaceArea	None	int – Return the corresponding ivar
Any others you would like to add – your choice. But it is suggested you add the method “**toString**” (helps for debugging). Actually “toString” should be added to every class you code.		

Problem 2 -- LakeSet

All of the required work for this guy is exemplified here:

- Course Notes
  - Chapter 4 -- Lists (Introduction)
    - Java Example Files
      - List Labs Using Dog Objects (See Me First)

Class Name: LakeSet

Instance Variables:

Name	Type	Description
lakes	ArrayList<Lake>	A list of lakes

Constructors:

Name	Parameters	Notes
Lake	None	Should construct the lakes ivar (initially an empty list)

Instance Methods **(see descriptions below this table)**:

Name	Parameters	Return Type
add	Lake	None
lakeNamed	String (a name)	Lake
deepest	None	Lake
shallowest	None	Lake
largest	None	Lake
smallest	None	Lake
allLakesNear	String (a name)	ArrayList<Lake>

General Notes

For deepest, shallowest, largest, smallest – do not worry about the case that multiple lakes "tie" – e.g. say two have the same maxDepth. In such a case, returning any of the lakes that "tie" is fine.

Method Descriptions

Method: add
Description: Add a Lake into the "lakes" ivar

Method: lakeNamed
Description: Return the *first lake* you find in the lakes "ivar" that matches the method parameter. If none match, return *null*.

Method: deepest
Description: Return the deepest lake (lake with largest max depth)

Method: shallowest
Description: Return the shallowest lake (lake with smallest max depth)

Method: largest
Description: Return the largest lake (lake with largest surface area)

Method: smallest
Description: Return the smallest lake (lake with smallest surface area)

Method: allLakesNear
Description: Return all the lakes that have a "nearbyTown" that matches the method parameter.

Extra Credit #1 (15 Points)

Using wiki etc, research state, national, or world lakes. Build up a list of at least 100 lakes for interesting queries. Provide instructions so that a user (e.g. grader) could construct this list of lakes (they would then be able to use it in the normal way – with methods defined above).

For help on populating a larger number of objects see:

- Course Notes
  - Chapter 4 -- Lists (Introduction)
    - Java Example Files
      - Countries List Example
        - **CountryListing.java**

Extra Credit #2 (15 Points)

We've been working on space objects. In Assignment 6, we have a ZIP file named "Outer Space.zip" that has a collection of space objects bouncing off walls. But they don't bounce off one another! For this extra credit, have the space object bounce off one another when they collide. Do [not] modify GraphicsApp.java. Rather, change the OuterSpace class (and any classes it uses).