Introduction to Java

CS9053 Section I2

Wednesday 6:00 PM – 8:30 PM

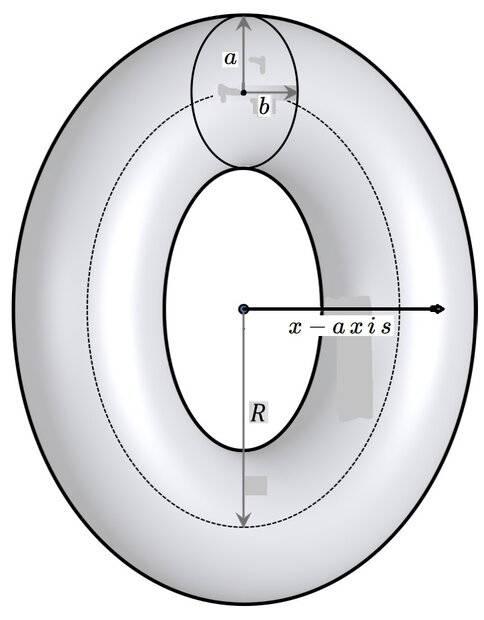
Prof. Dean Christakos

Sept 20th, 2024

Due: Sept 27th, 2024 11:59 PM

Part I – Creating Objects

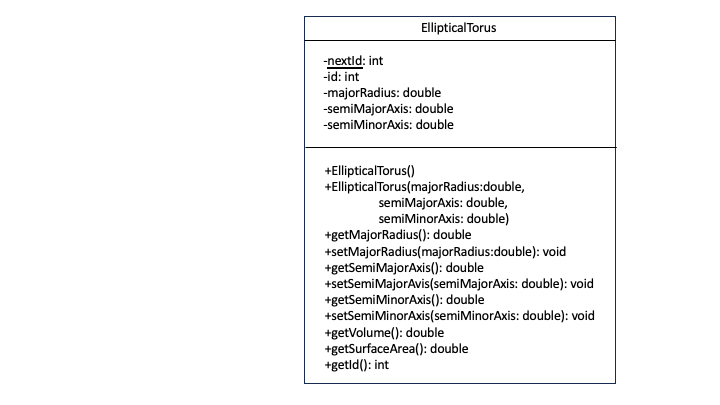
1. Elliptic Torus: In the lecture you have seen the creation of a circle. Here you are going to create an elliptic torus. An elliptic is a torus with an elliptical cross section. It has a major radius ***R***, a semi-major axis ***a***, and a semi-minor axis ***b***



The volume of an elliptic torus is given by

And the surface area is given by

You will create a class EllipticalTorus using the following UML:



**In standard UML parlance, “+” indicates that a field or method is public and “-“ indicates that a field or method is private. An underlined field or method indicates it is static.**

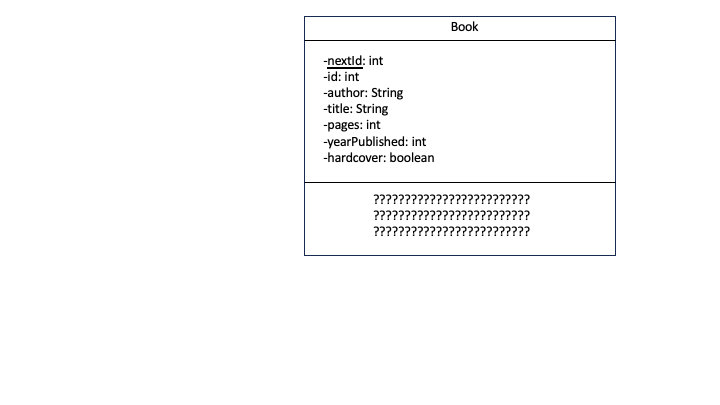
Every time you create a new EllipticalTorus instance, it should have a new sequential id, based on the value of nextId, which should be incremented every time you create a new EllipticalTorus instance.

1. Objects and Arrays of Objects

Your objective is to develop the **Book** and **Library** class and to use their methods. The classes are described below to guide you.

Book

The **Book** class contains data related to books.The **Book** class is immutable—it can’t be changed once it’s been instantiated and follows the following UML:



I didn’t put any methods or constructors in here. Figure that out how to make methods and constructors for an immutable **Book** class while allowing us to read the data.

Library

The **Library** class contains an array of **Book**‘s that are in the library. The Book objects have an id and the library keeps track of which books are lent out by id.

When lending out a book, the lendBook() **method returns the Book object and returns null if it isn’t available** (eg, doesn’t exist or is already lent out).

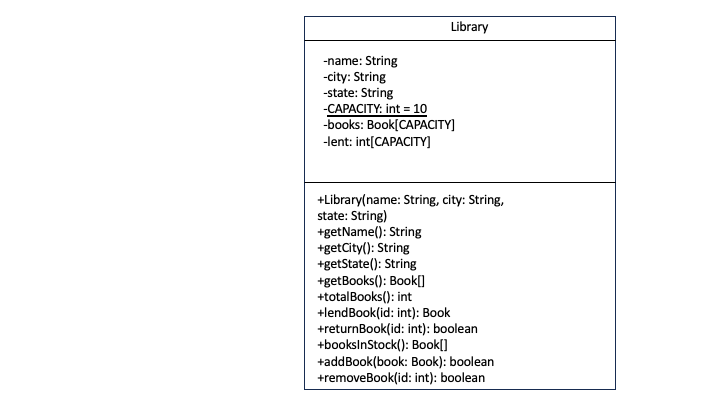
returnBook() returns true if the book is part of the stock and has been lent out and false if it isn’t lent out or doesn’t exist.

The library has a pre-set and **constant** capacity of 10 books and addBook return false if a book is added to the library beyond its capacity.

A book can be removed from the inventory with removeBook unless it is lent out or doesn’t exist in the inventory, in which case it returns false.

Note what can be changed and what can’t be changed based on the methods available in the UML.

Implement the class according to the following UML diagram:



Because we are using arrays rather than other data structures like Maps and Sets, the process of inserting and removing items from arrays is going to be inefficient. That’s ok. We aren’t testing you for a job. Do what you have to do, even if they run in n2

After implementing this, create three books of your choice, add them to the Library. Show the books that are in the inventory. Lend a book, and show which books are remaining. Return a book.