CS2230 Computer Science II: Data Structures
Homework 2
Objects in Java: the cylinder game

Due February 3, 2017, 11:59pm

Goals for this assignment

- Get practice programming with objects
- Understand how to use arrays of objects
- Write an algorithm using an array of objects

Submission Checklist

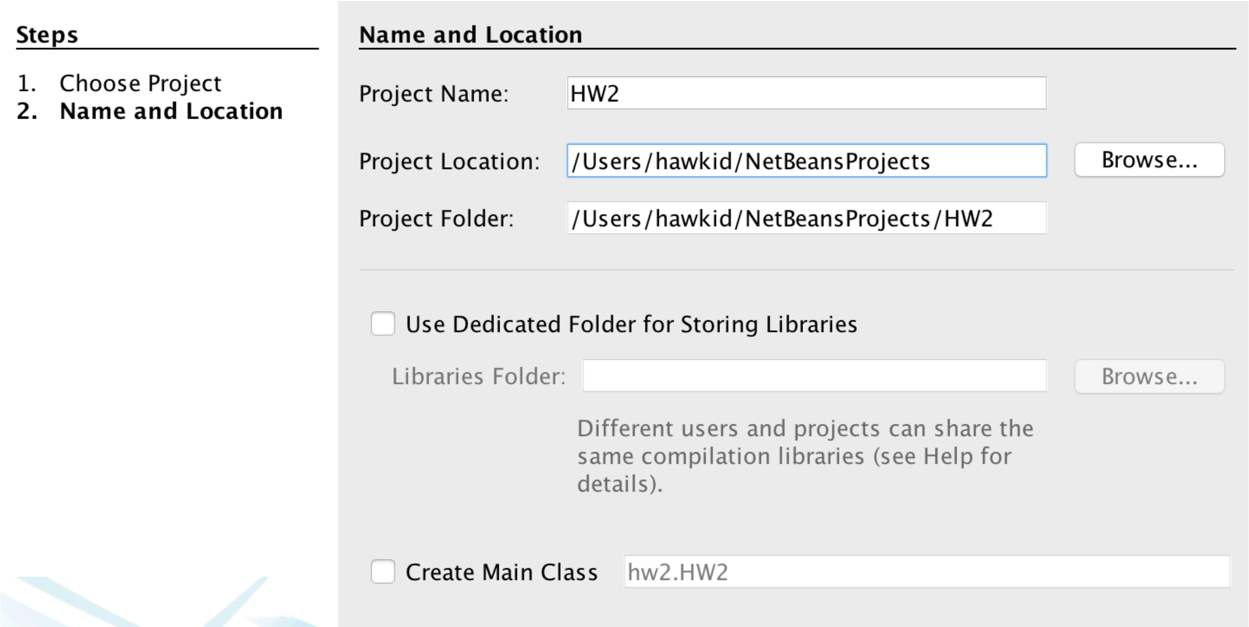
You should submit the Cylinder.java and CylinderGame.java. Upload them on ICON under Assignments > Homework 2. Physical paper copies or images are not acceptable.

- Does CylinderGame.java run properly?
- Did you submit both files to ICON?
- Did you double check your ICON submission to be sure you uploaded the right files?

Part 0: Setup the project in NetBeans

If you want to just create a new project and copy/paste the starter code in like HW1, you can do that. Alternatively, the following steps allow you to copy *files* into your project instead of copying the text.

1. Create a new Java project in NetBeans called HW2. Make sure that you uncheck the box that says "Create main class". Also make sure that you take note of what your Project Folder is (mine is `/Users/hawkid/NetBeansProjects/HW2`). You will need it in the following step. Finish the wizard.



Steps

1. Choose Project
2. **Name and Location**

Name and Location

Project Name: HW2

Project Location: /Users/hawkid/NetBeansProjects Browse...

Project Folder: /Users/hawkid/NetBeansProjects/HW2

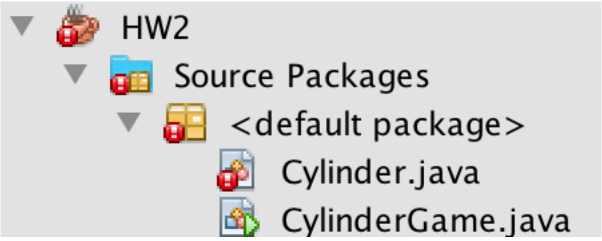
☐ Use Dedicated Folder for Storing Libraries

Libraries Folder: Browse...

Different users and projects can share the same compilation libraries (see Help for details).

☐ Create Main Class hw2.HW2

2. Find the two Java files Cylinder.java and CylinderGame.java on your computer and copy them into the src/ folder inside of the Project Folder you saw in step 1. (for me that is `/Users/hawkid/NetBeansProjects/HW2/src`).
3. You should now see something like this in NetBeans

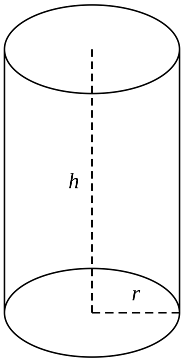


```
HW2
  Source Packages
    <default package>
      Cylinder.java
      CylinderGame.java
```

4. You are ready to start!

Part 1: The Cylinder class

We have provided starter code for a Cylinder class in Cylinder.java. A Cylinder object represents a 3D cup (has an open top) with a radius and height.



A Cylinder may be filled by water up to its height. It could be empty, partially full, or completely full. As you'll see, we can also pour water from one Cylinder into another. After a Cylinder is constructed, its radius and height never change again, but the height of the water may be increased or decreased by pouring.

Your job is to provide the missing method bodies and methods. We suggest that you work in the following order:

1. Fill in the two constructors. One of them takes two arguments and the other takes zero arguments. You should follow the instructions provided in the comments for each one. In the comments, notice that Cylinders must always start out with no water in them.

2. Write the methods `getRadius` and `getHeight.` These methods do nothing more than return the value of the corresponding field.

3. Write the methods `getVolume` and `getWaterVolume`. The methods return the volume of the Cylinder and the volume of the water in the Cylinder, respectively. Recall that the volume of a Cylinder is $\pi r^2 h$. However, to avoid the ugliness of irrational numbers we are going to assume the π is implicit and avoid multiplying it into our answer. Therefore, your methods should just return $r^2 h$.

4. We need a way to pour water from one Cylinder into another. Fill in the `pourWaterFrom` method. This method takes another Cylinder called `other` as an argument and moves water from `other` into `this` Cylinder.

Part 2: The CylinderGame class

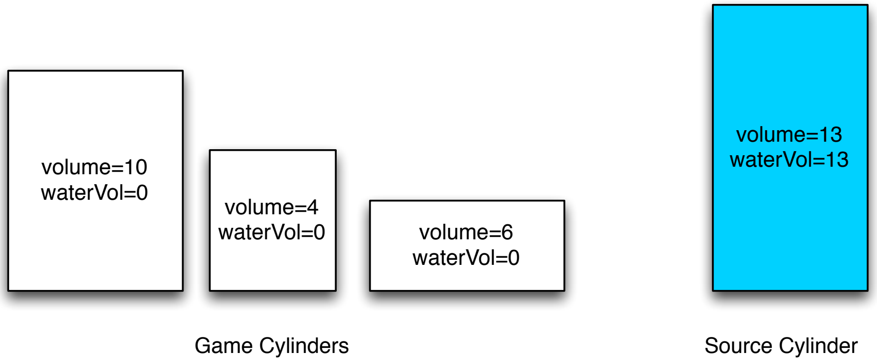
Now that we have working Cylinders that can hold and pour water, let's use several Cylinder objects to play a game. A game has two phases. First, Cylinders of different sizes are added to the CylinderGame. Second, we give the game a full "source Cylinder" which will be poured into the game's Cylinders in the order in which the Cylinders were added to the game.

Notice that there can be many kinds of outcomes depending on the sizes of the source Cylinder and the game's Cylinders.

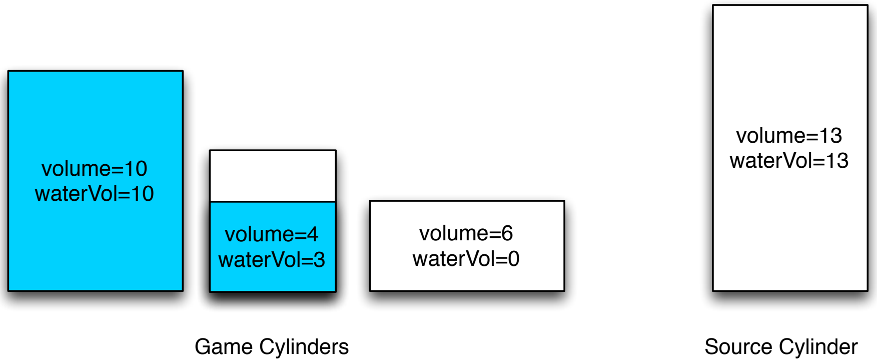
Here are some example games.

Game example #1

BEFORE

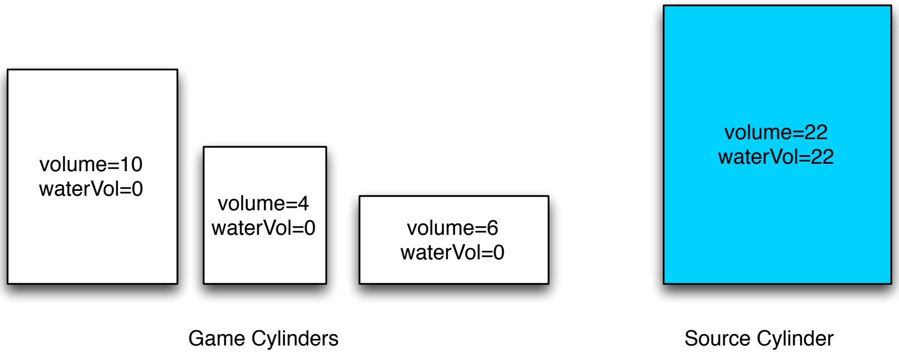


AFTER

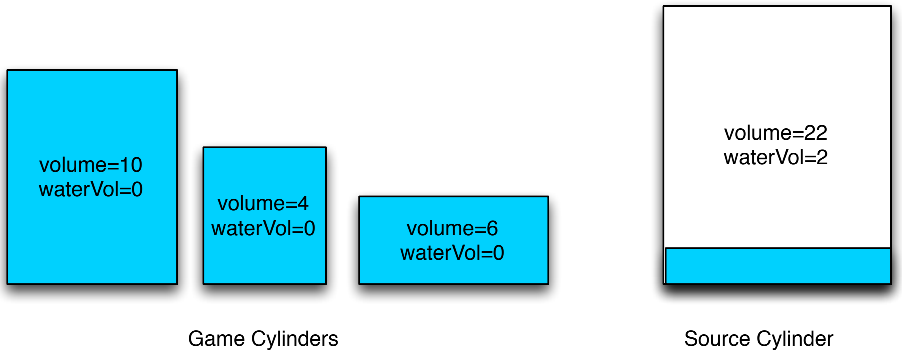


Game example #2

BEFORE



AFTER



To complete the `CylinderGame` class, you should do the following.

5. We need a way to add Cylinders to the game. Fill in the `addCylinder` method. Notice that we've already provided two fields for you to use to store the Cylinders: a Cylinder array called `cylinders` and an integer called `count`. If you need an example of how to add a new object to an array, see textbook chapter 3.1 or the lecture notes from 1/25.

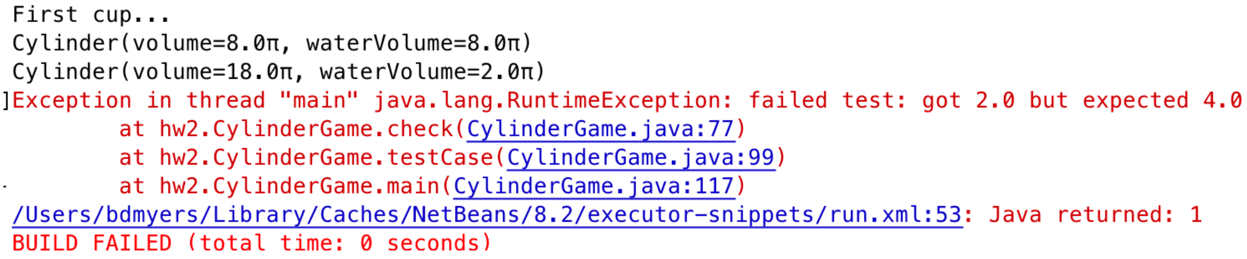
6. We need a way to delete Cylinders from the game. Fill in the `deleteCylinder` method. It should delete the Cylinder at the end of the `cylinders` array.

7. Finally, you will fill in the `fillCupsInOrder` method that pours water from the source Cylinder to the game's cylinders. Use the example games above to make sure you understand the rules. The method must fill the game Cylinders in the order they were added to the game.

8. When you've finished the above parts, you should be able to run `CylinderGame` to see if it works. There are three test cases provided (see the `main` method). Each test case gets a game, a source Cylinder, and the *expected* volumes at the end of the game.

Helpful Tips

- What do I do if my program prints something like the following?



```
First cup...
Cylinder(volume=8.0π, waterVolume=8.0π)
Cylinder(volume=18.0π, waterVolume=2.0π)
]Exception in thread "main" java.lang.RuntimeException: failed test: got 2.0 but expected 4.0
        at hw2.CylinderGame.check(CylinderGame.java:77)
        at hw2.CylinderGame.testCase(CylinderGame.java:99)
        at hw2.CylinderGame.main(CylinderGame.java:117)
/Users/bdmyers/Library/Caches/NetBeans/8.2/executor-snippets/run.xml:53: Java returned: 1
BUILD FAILED (total time: 0 seconds)
```

This message means that the last Cylinder printed (i.e., `Cylinder(volume=18.0π, waterVolume=2.0π)` has the wrong ending state. It was supposed to have a water volume of 4π but it only got 2π. You need to debug your program.

Extra credit

The extra credit will only be looked at if the rest of your submission works completely. It is worth 5% additional points maximum.

Our game wasn't much of a game. Make a copy of `CylinderGame.java` called `CylinderGameEC.java` that is a real game (i.e., can be won, lost, or produce a score), either for 1 player (e.g., puzzle solving) or for 2 players (e.g., versus). The only constraint is that it is in some way based on calling the methods `fillToTop` and `pourWaterFrom`.

Submit your additional file `CylinderGameEC.java` including comments at the top describing the rules of the game.