hw0 problem 4(PSOFT)

Brian Wang-Chen

January 2024

4.1

What is wrong with Ball.java?

1. The first issue with Ball.java is that it's constructor is not correctly implemented. It is using the private variables, volume and color incorrectly. It should be this.volume and this.color

Changes:
this.volume = volume;
this.color = color;

2. The second issue is that there is a issue with the accessors of the Ball class, getVolume and getColor. They are not returning the volume and color of the Ball but instead returning 0 and Null. Changed it so it returns the color and volume.

Changes:
return volume;
return color;

3. The third issue is that in the second constructor of the Ball class, where it takes a string as a volume, it does not consider if the string passed in is a invalid string. This will throw errors, so to fix that, a try and catch block is implemented to first try to see if it was a valid string, if not it catches the error, and just simply sets the volume to zero.

```
public Ball(String volume, Color color) {
    try{
        this.volume = Double.parseDouble(volume);
        this.color = color;
    }
    catch (NumberFormatException e){
        this.volume = 0.0;
    }
}
```

4.2

Implement BallContainer class methods. There are two obvious approaches to implementing getVolume(), which one is better? Why?

1. **Approach One**: Every time getVolume() is called, go through all the Balls in the Set and add up the volumes. Hint: one solution might be to use a for-each loop to extract Balls

2. **Approach Two**: Keep track of the total volume of the Balls in BallContainer whenever Balls are added and removed. This eliminates the need to perform any computations when getVolume() is called from the Set.

3. **Answer**: I believe Approach Two is better because it eliminates the need to iterate through the entire set each time getVolume is used. Instead keeping track of the total volume as balls get added or removed would be much more efficient so whenever getVolume is called, no calculation or iteration needs to be done, it will simply return the total volume. This holds true especially when the set gets massive, having to iterate over it multiple times is very inefficient.

4.3

Implementing Box class. Answer 2 questions

1. There are many ways to implement getBallsFromSmallest(). Briefly describe at least two different ways. Your answers should differ in the implementation of Box, not in lower-level implementation (for example, using an insertion sort instead of a selection sort is a lower-level implementation because it does not affect how Box is implemented). Hint: think about different places in the Box class where you could add code to achieve the desired functionality.

   (a) One way to implement getBallsFromSmallest is in the function, iterate through the ball container object, and add the all the elements to a new array list. Then You can simply sort the array list by using the collection sort from the collection library. Once the array list is sorted, you can then return the iterator of the sorted array list.

   (b) Another way to implement getBallsFromSmallest is to add some code in the add method of the Box class to sort the array as it is being made. In the add method, you will iterate through the ballContainer and find the index the new ball will be added to and check conditionals if the ball can be put in. This will eliminate the need of the sorting in getBallsFromSmallest method since the container will always be sorted, you can just return the iterator.

2. Which of the above ways do you think is the best? Why?

   First,which is to iterate and sort in the getBallsFromSmallest because it will be more efficient compared to the second method of sorting as you create the ball container. The first method will be faster because you only iterate the ballContainer once to create the new ArrayList, and then use the Java collection library to sort it without having to iterate. The second method is slower because you would need to iterate through the container each time a new ball is being added to sort it, which will be cause a longer runtime.