Jasque Saydyk

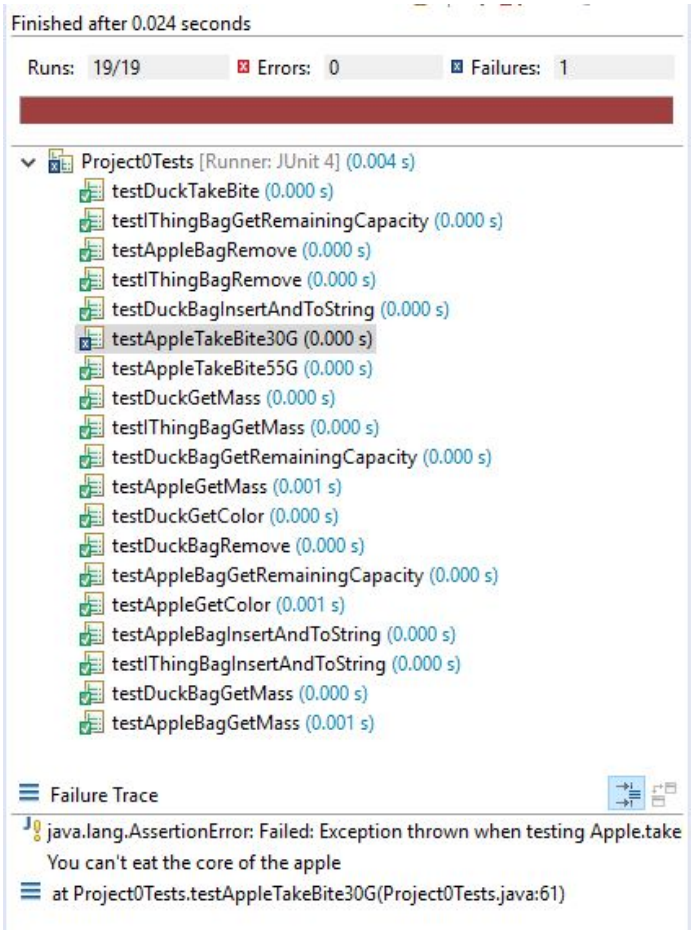
Professor Patrick Kelley

CS 249 Section 3140

15 February 2017

Project 0 - Java Review

For this project, we were tasked with two primary goals, first creating a set of classes that called the IThing interface, and second was creating the Bag class, which implemented the IBag interface using the IThing interface as it's generic.



The Apple class and the Duck class both called the IThing interface, making them have to make a getColor(), getMass(), and takeBite() method for the two of them. For the Apple class, the implementation of the getColor(), getMass(), and takeBite() were straightforward, with the only nuance being that for the takeBite() method, if there were less than ten grams till you reached the apple core, the bite will remove the rest of the grams, leaving only the apple core. The Duck class was simple, with the majority of the methods only containing one line. As for the exceptions, it took me a while to figure out that nothing special was suppose to go on inside of the exceptions that were created, that they should only call the parent constructors as they are only exception identifiers.

As for the Bag class, the IBag interface dictated that it needed to have a create, insert, remove, getMass of the bag, getRemainingCapacity of the bag, and a toString method. I ended up wrapping an ArrayList for the Bag class, and the implementation of the methods were pretty straightforward and simple if one knows how to use for-each loops.

The only difficulty I had was that I had to relearn the purpose of generics and the different generic names, like T for type and E for element. After I had written half of the class, I got stuck on the getMass() method when I realized that the generic was suppose to be for IThing to allow

for polymorphism, and not E. Looking back through the code to refactor it several days later, I also realized that I could remove ten lines of code from the class just by using the getMass() function whenever I needed the mass of the bag, making the code simpler.

For unit testing, I only used the given unit tests and did not implement any of my own. I used JUnit to run the tests, and on the first page, you can see that I failed the testAppleTakeBite30G on the first Unit test I conducted. Below is the current project, having succeed all of the given tests.