Therese Dachille

CS 1632 – DELIVERABLE 4

PROPERTY-BASED TESTING

GitHub: https://github.com/tcd12/CS1632Deliverable4

I chose the JUnit-based property-based testing option for this deliverable in order to become even more comfortable with JUnit testing, as well as to practice identifying properties for testing. Since we worked with JUnit testing for the past couple of deliverables, I was familiar with the format and syntax, but felt I needed to practice writing tests more in order to master at least what we’ve covered in this course so far. I finally feel as though I could easily produce JUnit tests in the future, whether it be for this course, or another Computer Science course, or for Quality Assurance in the workplace.

In order to test the properties of Java’s Arrays.sort(int[] arr) method, I began by thinking of the properties of arrays. I came up with a list that led me to test four properties: that an unsorted array maintains its length after being sorted, that every element in a sorted array is greater than or equal to each element before it, that sorting an already-sorted array results in the same, sorted array (idempotency), and that each value in an unsorted array still exists after that array is sorted.

While working on this deliverable, I had few issues. However, the most challenging part might have been coming up with examples of properties to test. Since the properties seem to be so common sense, it can be difficult pointing out such obvious features of arrays and Arrays.sort(int[] arr). Beyond that, the other issue I faced was choosing where I should bound the size of my arrays, since using Integer.MAX\_VALUE ran into time and memory problems. But after some trial and error, I decided to cap the size at 10,000, in order to prevent hindering my unit tests’ performance.

Through this deliverable, I learned examples of properties to test during property-based testing. I also feel as though this deliverable helped me to fully understand JUnit testing and ultimately made me comfortable with the process of unit testing.

Screenshot of JUnit Tests

