Keith Ellison

Project Two

CS 320

When given guidelines to follow, my work is influenced by that before anything else. Knowing what was being asked of me in the requirements came first, then on to mock names and addresses that could be tested against checks for their length or if the fields are empty. Since I had to write code, the tests used were dynamic though my first few runs were full of errors and I spent more time than I would like to remember revising code over and over, running tests, revising further and then a few rounds of exporting and importing the files to make sure everything was ready for submission.

I know my tests were effective because I learned the process the hardway; re-doing tests over and over until I understood what was going wrong and where, submitting the results over and over until I got a respectable score. Clearing the IDE of errors was how I made sure there were no errors, beyond that, I will let what I said above speak for the rest. Coverage percentage, If I was looking at the right thing, was pretty low those first few times, but I would get one or two positive check marks that wouldn’t always repeat, so that part had to be rewritten.

Task task = new Task("867530900","checkChance","checking Chance for errors or null");

Assertions.assertEquals("867530900", task.getTaskId());

}

As stated, I wrote my own code, so my tests were dynamic, the above code is an example of what was written. In my case I used individual if/then statements to check if the required conditions were being met. This approach is simple enough to avoid bugs that won’t be picked up by the IDE, and dynamic enough to catch issues against any or all of the conditions.

I used a form of boundary testing, which is where you set conditions for a check you’re running between two extremes and if the result is neither of those extremes, then the condition will pass/fail, depending on how you wrote the condition to return. So in my case, the extreme would be between null(0), and 10, if the returned value is less than 0 or more than ten, that condition fails and the user must fill out that field again. My tests are like that for each condition, checking the number of characters used for each required field.

I feel like I either did or didn’t use “Decision table” tests, these tests check for certain conditions and user inputs; the included package expects certain inputs from the user and then reacts accordingly. The developer can still set their conditions, and the program would produce an error if an end-user failed to meet those conditions, just like with the boundary tests, but I am not sure I can take credit for applying both methods.

In day to day, or practical use, “Boundary tests” have been my trusted coding method since I first taught myself how; if/then statements, which for me, have been the backbone of a program outside of school, well they can so easily apply boundary tests as their conditions that I was using them before I knew what they were. Decision tables ditto, writing code and checking for harder, control conditions instead of variables one has I am positive come up before outside of school but I can remember the application I applied that technique for.

I am a little uncomfortable working within someone else’s conditions, or I start off thinking that way every time before actually starting to relax once I read the requirements and begin to relax and settle into the project. The problem I think is that I’m not comfortable with how I’ve learned these techniques in the school setting, I never feel that I really get to know my professors or gain any impression of them during the term before it is time to move on. On top of that, I don’t feel that I’m being taught by them so much as a lesson packet, though this time has been very different Dr.Bolton with me trying to make up so many assignments. There is no personal attachment to my code when it’s not mine personally, or not this time anyway, maybe if I overcame my expectations of myself I would feel differently. Without discipline I know from experience that you may not even stick with an assignment, and that’s one skill I for sure came into school with having coded on my own, and raged at on my own.