A2/CIS*4150

Specifications and State

courselink.uoguelph.ca

A. Hamilton-Wright

Overview

In this assignment we will examine specifications, unit tests and begin our exploration of mocking.

Note that this is an individual assignment. All of the submitted work should be your own, and based on the starting code and course examples. Any work you have included from elsewhere (*e.g.*, the examples) must be noted and cited.

Skills

specification (6/6)

graph structures (3/6)

testing tools (3/6)

unit testing (6/6)

(*)[The skill scale is from 0 (Fundamental Awareness) to 6 (Main Focus).]

Image description

Pretending to be a pirate. Photo source shutterstock.com Shutterstock license.

CIS 4150/F21: Assignment Two

For this pen and paper assignment, there are a mixture of problems from the the book, as well as specific posed problems, below.

The due date and time for this assignment (and all others) are listed in the syllabus.

Book Question

Please answer question 5.7.

Code related question

Consider the GPA library that we looked at grade conversions in, in the last assignment. Both the Python and Java versions of this library have the same goals and try to provide the same functionality.

The overall goal is to track grades for a set of students, using letter grades. This is done using the "calculator" class, while the conversions among various grade formats is in the "converter" class.

Maintaining grades for a set of students requires state. There will be a grade for each of a set of multiple courses, and this is true for a set of multiple possible students.

For this assignment, you are to think about how the state machine for grade tracking across multiple students, courses and terms must work.

There are Javadoc documents generated for you on the Java side, outlining the set of functions available in the system.

State machine

We will evaluate the system, in terms of its ability to calculate a GPA for a year, assuming that:

- we will only consider one term[1] ~~there are three terms in a year during which students may take courses~~
- students do not take more than two (2)[1] ~~five (5)~~ courses in a term[2] ~~year~~
- for simplicity, we can assume that there are a maximum of three students[3]
- We will only consider the grades 'A', 'B' and 'F', without ± decoration[1] ~~Also for simplicity, please assume that we are only looking at letter grades with no "+" or "-" – that is, only consider 'A', and not all of 'A+/A/A-'~~[4]

Draw a state machine describing the way this system works. Note that this state machine should be applicable to both code bases, as the state machine is not specific to a particular language.

[1] Nov 3: As discussed in class today, I accidentally created a scenario with a huge number of states, so these lower numbers will reduce the number of states required in your state machines

[2] Oct 25: This originally said "year," which was a typo. As the limit in the code is a by-term limit, this makes little sense.

[3] Nov 3: As we discussed in class, this constraint is not directly relevant to your state machine construction

[4] Oct 26: I added this as otherwise your state machines will be huge, to no real benefit.

A2/CIS*4150

Specifications and State

courselink.uoguelph.ca

A. Hamilton-Wright

Overview

In this assignment we will examine specifications, unit tests and begin our exploration of mocking.

Note that this is an individual assignment. All of the submitted work should be your own, and based on the starting code and course examples. Any work you have included from elsewhere (*e.g.*, the examples) must be noted and cited.

Skills

specification (6/6)

graph structures (3/6)

testing tools (3/6)

unit testing (6/6)

(*)[The skill scale is from 0 (Fundamental Awareness) to 6 (Main Focus).]

Image description

Pretending to be a pirate. Photo source shutterstock.com Shutterstock license.

Testing

Design a set of test cases for each language implementation that evaluates whether the implementation matches your state machine.[6] [6]

To do this, you must consider how you can get the system to change state (by adding or removing grades) and what observable outcomes related to state are available.

You should have the same evaluation in both systems (i.e.; if you have a test case where you added an "A" to a term to get to a particular state in one implementation, then you should have a test case adding an "A" in the other as well).

There should be at least one test case per state in your state machine.

You may assume the same limits stated above are in place when designing your testing[5]

Writeup

Produce a document including your answer to the textbook question, and your evaluation of the GPA systems.

Your document should contain:

- Your state diagram, as well as any explanation required to understand the diagram (tell us what the nodes mean, how transitions happen, etc).
- The list of tests, explaining how your test relates to the states.
- ~~For each system, the outcome of each of your tests.~~[7] [7]
- An overall summary explaining what you learned about any differences between the implementations.

Handing In Your Assignment

Collect all of the files you wish to submit into one archive (`.zip` or `.tar`) and upload to the Courselink dropbox.

[6] Nov 4: Note that I am only asking for the tests design here, not implementation. You only need to tell me how your test is going to work (what calls are made, and what results are expected). People who have already written code can have a bonus – please see note 7.

[5] Oct 25: This sentence was added in response to a student question.

[7] Nov 4: When I wrote this assignment up, I was only anticipating you to design the tests, however because of this line, of course people are implementing the tests. As this was (again) my error, I will award a 10% bonus to anyone who has written test implementation code, but all I am really expecting is a statement of what the expected result is.