CS 458 - Homework 6

Deadline

Due by 11:59 pm on Friday, October 28, 2016

How to submit

Submit your files using `~st10/458submit` on nrs-projects, with a homework number of `6`.

Purpose

To very lightly try out JUnit for unit testing, and to think about several things related to software project risks.

Important notes

- Note that some of your submissions for this assignment may be posted to the course Moodle site.

Problem 1

Fun Fact: in Java, you can declare an array of integers using:

```
int myArray[];
```

And, then, you can instantiate that array using:

```
myArray = new int[desiredNumElements];
```

And then, as in C++, the array `myArray` has elements with indices `0` to `(desiredNumElements-1)`, and you can access or set the element with, say, index `3` (the 4th element in the array) using:

```
myArray[3]
```

(and, as in C++, you need to initialize the values in the array before trying to use them! `8-)`)

Consider the posted `GameDie.java` and `GameDieTest.java` classes (posted along with the Week 9 Lab projections on the public course web site) which we used to demonstrate a little bit about TDD (Test-Driven Development) and JUnit unit testing.

You are expected to follow the coding and testing style of these given classes in completing this problem.

Also remember: you know, from the in-lecture examples, that when you have `GameDie.java` and `GameDieTest.java` in the same directory and open them at the same time in DrJava (freely downloadable from `drjava.org`), you can run the JUnit tests in `GameDieTest.java` using the "Test" button in DrJava.

1 part a

Make copies of the posted `GameDie.java` and `GameDieTest.java` classes, and add your name to the

opening comment block of each (indicating that these are your modifications).

(Note: it is also okay if you'd like to remove some of the more "explanatory" comments from these posted examples you are starting from. But make sure each class and method still has an opening comment, at least giving the purpose of that class or method, and don't remove the javadoc-style comments.)

1 part b

Consider the user story: "A user can see how many times a game die has rolled a given value.."

(For example, the user has a way to find out how many times a 20-sided game die has rolled a 17, or a 12, or any other value they choose to specify.)

(For our purposes, having not discussed exception-handling in Java, you can assume that if the user specifies a non-existent value for a game die -- for example, they ask how many times a 12-sided game die has rolled a 27 -- it can simply return 0, as such a game die has definitely not ever rolled that value...)

Write appropriate tests in `GameDieTest` that should pass if a user can indeed do this -- ideally, do so in a test-driven-development approach, writing a test, then modifying `GameDie` to pass that test (and still pass the previously-existing tests), and so on.

- You need to add at least one new test method to `GameDieTest`.
- You need to add at least one new private data field to `GameDie`; you also need to add at least one new method, and you need to modify at least two of its existing methods.

Submit your resulting `GameDie.java` and `GameDieTest.java`.

Set-up for Problems 2 and 3

Create a file named `458hw6.txt` that starts with your name. Then give the problem number and your answer(s) for each of the following problems.

Problem 2

Consider the list of Boehm's top 10 software risks, followed by risk-management techniques for each, posted along with this homework handout (as well as with the Week 9 Lecture 2 in-class projections).

Consider a 4-person student team working on a semester project. Select one of these which you think could be a major risk for such a project, and describe why you think it could be a major risk for such a project. Then consider the risk-management techniques suggested for your selection, and select one of these which you believe could indeed help manage that risk, and explain why you think it could help.

Then consider a 10-person team, say working in a start-up, working on a computer game. Select a *different* one of these which you think could be a major risk for such a project, and describe why you think it could be a major risk for such a project. Then consider the risk-management techniques suggested for your selection, and select one of these which you believe could indeed help manage that risk, and explain why you think it could help.

Problem 3

And now, for another kind of risk... You will find an overview by Nancy Leveson of an infamous series of accidents involving a computer-controlled radiation therapy machine, the Therac-25, linked from the course

Moodle site, in the "Additional Readings" section. This article is quite long, and please note that I found some of it to be disturbing reading (especially descriptions of the accidents and their aftermath in Section 3), and some of it is also quite technical. For our purposes, read **Section 1, p. 1**, and **Section 4, from pp. 44 - 49**. Section 4 summarizes lessons that the author felt could be learned from this series of accidents.

Consider: which of these lessons would you want to most remember with regard to good software engineering practice? Select at least three that you believe are especially important, and list them, along with why you feel they are important.

Submit your resulting `458hw6.txt`.